



VISION DE PROCESO PARA ESTRATEGIAS INTEGRADAS DE MEDICION Y EVALUACION DE LA CALIDAD

Ing. Pablo Javier BECKER

Grupo de Investigación y Desarrollo en Ingeniería de Software y Web
(GIDIS_Web)

Calle 9 esq. 110, (6360) General Pico, La Pampa, Argentina
beckerp@ing.unlpam.edu.ar

Director: Dr. Luis OLSINA


Co-Director: Dr. Gustavo ROSSI

Tesis presentada para obtener el grado de

DOCTOR EN CIENCIAS INFORMÁTICAS

Defendida el día 03 de Octubre de 2014

**Facultad de Informática
Universidad Nacional de La Plata
Argentina**



**VISION DE PROCESO PARA
ESTRATEGIAS INTEGRADAS DE
MEDICION Y EVALUACION DE
LA CALIDAD**

Resumen

En la actualidad, las áreas de Medición, Evaluación y Análisis son pilares de la Ingeniería de Software ya que brindan información para, principalmente, tomar decisiones sobre cuestiones relacionadas a la calidad de productos, procesos, etc. En estas áreas, como en otras, es deseable garantizar repetibilidad para lograr consistencia en los resultados. Para ello, las organizaciones deben contar con estrategias de Medición y Evaluación (M&E) que integren tres capacidades: i) *una especificación de las vistas del proceso de M&E* en la cual se documenten, entre otros aspectos, las principales actividades que deben ser planificadas y ejecutadas, ii) *un marco conceptual de M&E* que permita disponer de un vocabulario común, y iii) *métodos y herramientas* específicos para llevar a cabo las actividades. En este sentido, hemos desarrollado una estrategia llamada GOCAME (*Goal-Oriented Context-Aware Measurement and Evaluation*), que integra estas tres capacidades.

La presente tesis doctoral se centra en la primera capacidad mencionada, es decir: la *especificación de las vistas del proceso de M&E* para la estrategia GOCAME. Contar con la especificación del proceso desde diferentes perspectivas permite tener una guía clara y repetible de las actividades que deben llevarse a cabo, como así también detectar puntos de control, identificar los productos de trabajo insumidos y producidos, y escoger adecuadamente los agentes que ocuparán los roles involucrados en la realización de las actividades, entre otros aspectos. Para especificar las diferentes vistas del proceso se utilizarán lenguajes estándares apropiados, permitiendo que exista un entendimiento compartido dentro y fuera de la organización que lo aplique, de modo que cualquier usuario interesado pueda comprender el proceso, hacer uso del mismo y/o plantear mejoras a éste.

Si bien existe literatura, e incluso estándares, que describen el proceso de M&E, lo hacen como si fueran dos procesos disjuntos, de hecho muchos ni hablan de Evaluación. Sin embargo, estos procesos están muy relacionados. Además, en la literatura revisada se observa que los procesos de M&E no se especifican desde sus diferentes vistas. Incluso se ha detectado que no existe un consenso respecto de la terminología de proceso utilizada, motivo por el cual aquí se propone una ontología de proceso (genérica). Esta ontología también se utiliza para enriquecer el marco conceptual de M&E (segunda capacidad mencionada).

La hipótesis principal de este trabajo es que sin una clara definición de las vistas del proceso, los programas y proyectos de M&E, suelen ser menos repetibles, reproducibles y controlables. Además, poseer un (modelo de) proceso basado en un marco conceptual de M&E no sólo permite contar con programas y proyectos con dichas características, sino que también favorece la robustez y confiabilidad de la información obtenida (datos y metadatos), la cual es utilizada en procesos de toma de decisión. También, un modelo de procesos facilita la definición de métodos y herramientas que permitan llevar a cabo las actividades, ya sea de forma total o parcialmente automatizada.

Por otra parte, cabe mencionar que GOCAME es una estrategia de M&E de propósito múltiple, la cual puede ser personalizada para desarrollar estrategias para propósitos específicos. Particularmente, a partir de GOCAME se ha derivado la estrategia SIQinU (*Strategy to Improve Quality in Use*), la cual se enfoca en mejorar sistemáticamente la Calidad en Uso de las aplicaciones evaluadas. Esta estrategia también cuenta con las tres capacidades de GOCAME, y en este trabajo se abordará la *especificación de las vistas del proceso* que subyace en SIQinU.

Agradecimientos

Esta tesis de doctorado es el producto de mucho tiempo de trabajo, estudio y esfuerzo. Terminar este trabajo hubiera sido muy difícil sin el apoyo de muchas personas, a quienes quiero agradecer.

En primer lugar agradezco a mi director, Luis Olsina, por su apoyo y su dirección. La verdad es que poco de este trabajo se habría realizado si no fuera por su dedicación y disposición. Sus oportunos consejos y las apropiadas directivas me han sido de gran utilidad. También fueron acertadas sus recomendaciones y críticas, siempre desde una perspectiva positiva y constructiva, permitiendo que se mejorara en gran manera esta investigación. Igualmente quiero agradecer a Gustavo Rossi, por co-dirigir esta tesis y brindarme el soporte logístico necesario.

También debo mencionar que este trabajo de investigación no hubiera sido posible sin la ayuda que he encontrado en el grupo de investigación GIDIS_Web. Muchas gracias a todos por su compañerismo, aliento y confianza. Una especial mención merece Fernanda Papa, con quien he trabajado hombro a hombro en parte de los avances de esta tesis. Y, alguien a quien no puedo dejar de agradecer, es a Hernán Molina, por su meticulosa revisión, así como por sus acertados comentarios.

En lo institucional, quiero dar gracias a la Facultad de Ingeniería de la UNLPam por haberme brindado la posibilidad de trabajar en dicha casa de estudios y por contribuir con parte del financiamiento para realizar los cursos del doctorado.

En cuanto a mis afectos, quiero agradecer a mi esposa, Vanessa, quien muchas veces debió compartirme con este trabajo y soportar mis ausencias, debido a los viajes a La Plata para tomar los diferentes cursos del doctorado. Además, un agradecimiento y un reconocimiento especial a mis padres, quienes hicieron grandes sacrificios para que pudiera realizar mis estudios universitarios, y así abrirme las puertas para hoy haber podido llegar hasta aquí.

Por último, y para que nadie se quede afuera de mis reconocimientos, agradecer a todas las personas que, de una u otra manera, también me han ayudado a lo largo de estos años.

Gracias a todos por su ayuda, tiempo y dedicación.

General Pico, La Pampa, 01 de Junio de 2014

Pablo Javier Becker

Tabla de Contenidos

Índice de Figuras	XII
Índice de Tablas	XVIII
Capítulo 1: Introducción	1
1.1. Introducción	1
1.2. Motivación.....	3
1.3. Contribución.....	5
1.4. Publicaciones Relevantes	6
1.4.1. Publicaciones en Revistas (Journals) Internacionales	6
1.4.2. Publicaciones en Congresos Internacionales.....	6
1.4.3. Publicaciones en Congresos Nacionales	7
1.5. Organización de la Tesis.....	7
Capítulo 2: Fundamentos sobre Estrategias Integradas de M&E y sobre Procesos	9
2.1 Fundamentos sobre Estrategias de M&E.....	9
2.1.1 Relevancia de M&E en las organizaciones	9
2.1.2 Estrategias Integradas de M&E.....	13
2.2 Fundamentos sobre Modelado de Procesos.....	15
2.2.1 Beneficios de modelar los procesos	16
2.2.2 Enfoques y Vistas de Modelado.....	17
2.2.3 Lenguajes de Modelado de Proceso	19
2.2.4 Lenguaje de Modelado de Proceso Utilizado	21
Capítulo 3: Estado del Arte de Procesos en Estrategias Integradas de M&E	25
3.1 Revisión de Enfoques y Estrategias de M&E	25
3.1.1 ISO - International Organization for Standardization	25
3.1.2 CMMI - Capability Maturity Model Integration.....	27
3.1.3 PSM - Practical Software Measurement.....	28
3.1.4 Alarcos Quality Center	29
3.1.5 GQM+Strategies - Goal Question Metric Plus Strategies	32
3.2 Especificación de las Vistas del Proceso de M&E	33
3.2.1 ISO - International Organization for Standardization	33
3.2.2 CMMI - Capability Maturity Model Integration.....	38

3.2.3 PSM - Practical Software and Systems Measurement	42
3.2.4 Alarcos Quality Center	46
3.2.5 GQM+Strategies - Goal Question Metric Plus Strategies	48
3.3 Marcos Conceptuales de Proceso	50
3.3.1 Marco Conceptual de Proceso Propuesto	56
3.4 Solución Propuesta	63
Capítulo 4: Estrategia GOCAME.....	67
4.1. Introducción	67
4.2. Marco conceptual de M&E.....	68
4.2.1 Componente de Proyecto de M&E	71
4.2.2 Componente de Requerimientos No Funcionales.....	71
4.2.3 Componente de Contexto	72
4.2.4 Componente de Medición.....	72
4.2.5 Componente de Evaluación	73
4.2.6 Componente de Análisis y Recomendación.....	73
4.3. Relación entre el Marco Conceptual C-INCAMI y el Marco Conceptual de Proceso.....	73
4.4. Especificación de las Vistas del Proceso de M&E	75
4.4.1 Proceso de GOCAME: Vista Funcional y de Comportamiento	76
4.4.1.1 Definir los Requerimientos No Funcionales (A1)	78
4.4.1.2 Diseñar la Medición (A2).....	85
4.4.1.3 Implementar la Medición (A3)	89
4.4.1.4 Diseñar la Evaluación (A4)	92
4.4.1.5 Implementar la Evaluación (A5).....	99
4.4.1.6 Analizar y Recomendar (A6).....	102
4.4.2 Roles de GOCAME: Vista Organizacional	104
4.4.3 Productos de Trabajo de GOCAME: Vista Informacional	109
4.5. Métodos y Herramientas.....	112
4.6. Conclusiones.....	115
Capítulo 5: Estrategia SIQinU	117
5.1. Introducción	117
5.2. Proceso de SIQinU	119
5.2.1 Fase I: Specify Requirements and Evaluation Criteria for QinU (Ph I).....	121
5.2.1.1 Establish Information Need.....	122
5.2.1.2 Specify Project Context.....	123

5.2.1.3	Design Tasks	123
5.2.1.4	Establish QinU Requirements Tree	124
5.2.1.5	Design QinU Measurement	125
5.2.1.6	Design QinU Evaluation.....	126
5.2.1.7	Design Preliminary Analysis	126
5.2.2	Fase II: Perform QinU Evaluation and Analysis (Ph II).....	126
5.2.2.1	Collect Data	127
5.2.2.2	Quantify Attributes	128
5.2.2.3	Calculate Indicators	128
5.2.2.4	Conduct Preliminary Analysis	129
5.2.3	Fase III: Derive/Specify Requirements and Evaluation Criteria for EQ (Ph III).....	129
5.2.3.1	Establish EQ Requirements Tree.....	130
5.2.3.2	Design EQ Measurement.....	130
5.2.3.3	Design EQ Evaluation.....	131
5.2.3.4	Design EQ Analysis.....	131
5.2.4	Fase IV: Perform EQ Evaluation and Analysis (Ph IV)	131
5.2.5	Fase V: Recommend, Perform Improvement Actions, and Re- evaluate EQ (Ph V)	132
5.2.5.1	Recommend Improvement Actions.....	132
5.2.5.2	Design Improvement Actions	133
5.2.5.3	Perform Improvement Actions	133
5.2.5.4	Evaluate Improvement Gain.....	133
5.2.6	Fase VI: Re-evaluate Quality in Use and Analyze Improvement Actions (Ph VI).....	134
5.2.6.1	Conduct Improvement Actions Analysis	134
5.3.	Alineación de GOCAME y SIQinU	135
5.4.	Plantilla de Especificación de Procesos.....	137
5.5.	Conclusiones.....	141
	Capítulo 6: Caso de Estudio usando SIQinU	143
6.1	Introducción	143
6.2	Fase I: Specify Requirements and Evaluation Criteria for QinU	144
6.3	Fase II: Perform QinU Evaluation and Analysis	148
6.4	Fase III: Derive/Specify Requirements and Evaluation Criteria for EQ	151
6.5	Fase IV: Perform EQ Evaluation and Analysis	153

6.6 Fase V: Recommend, Perform Improvement Actions and Re-evaluate EQ	154
6.7 Fase VI: Re-evaluate Quality in Use and Analyze Improvement Actions	157
6.8 Consideraciones Finales.....	159
Capítulo 7: Conclusiones.....	163
7.1. Problemas Observados.....	163
7.2. Contribuciones Realizadas.....	164
7.3. Otras Consideraciones.....	165
7.4. Trabajos Futuros.....	168
Referencias.....	171
Apéndice A: Lenguajes de Modelado de Proceso.....	183
A.1. IDEF0 (Integration DEFinition 0).....	183
A.2. Diagrama de Actividad de UML.....	184
A.3. BPMN (Business Process Modeling Notation)	185
A.4. SPEM 2.0 (Software & Systems Process Engineering Meta-Model 2.0)	187

Índice de Figuras

Figura 2-1: Taxonomía propuesta en [Abran, y otros, 2004] para un posible <i>Área de Conocimiento de Medición de Software</i>	10
Figura 2-2: Áreas de proceso de soporte de CMMI y su relación con las demás áreas de proceso (tomado de [CMMI Product Team, 2010]).	11
Figura 2-3: Alegoría de los tres pilares sobre los cuales una estrategia de M&E debería sostenerse.	15
Figura 3-1: Estándares del proyecto SQuaRE.	27
Figura 3-2: Marco conceptual de FMESP.	30
Figura 3-3: Modelo de Proceso de Medición de Software (adaptado de [ISO/IEC, 2002a]).	34
Figura 3-4. Modelo de Proceso de Evaluación de Software (adaptado de [ISO/IEC, 2001b]).	36
Figura 3-5: Diagrama de contexto de Medición y Análisis en CMMI (adaptado de [Ahern, y otros, 2008]).	40
Figura 3-6: Proceso de Medición de PSM.	42
Figura 3-7: Vista funcional de la actividad <i>Ajustar las medidas</i>	43
Figura 3-8: Vista funcional de la actividad <i>Aplicar las medidas</i>	43
Figura 3-9: Proceso de Medición de PSM usando la terminología de ISO 15939.	44
Figura 3-10: Ejemplo de composición del Plan de Medición (tomado de [PSM, 2000]).	45
Figura 3-11: Proceso y fases de la metodología de evaluación EVVE (tomado de [Rodríguez, y otros, 2010b]).	46
Figura 3-12: Vista organizacional y de comportamiento de la actividad <i>Obtención y Análisis de Artefactos a Evaluar</i> (tomado de [Rodríguez, y otros, 2010b]).	47
Figura 3-13: Roles y relaciones en el proceso de evaluación.	47
Figura 3-14: Vista funcional del proceso general de GQM (tomado de [Gresse, y otros, 1995]).	49
Figura 3-15: Vista de comportamiento del proceso general de GQM utilizando un diagrama de tiempo.....	50
Figura 3-16: Vista informacional del producto de trabajo <i>documentación del plan GQM</i> (tomado de [Gresse, y otros, 1995]).	50

Figura 3-17: Constructores de proceso según ISO/IEC 12207/15288 (tomado de [ISO/IEC, 2004]).	51
Figura 3-18: Componentes básicos de modelado de proceso según [Acuña, y otros, 2001b].	52
Figura 3-19: Modelo conceptual para el dominio de procesos Web (tomado de [Esteban, y otros, 2003]).	53
Figura 3-20: Principales conceptos del metamodelo de SPEM 1.1.	54
Figura 3-21: Principales conceptos de la <i>Ontología de Proceso de Software</i> (tomado de [Bringunte, y otros, 2011]).	55
Figura 3-22: Principales componentes de proceso de OPF y sus relaciones (adaptado de [OPF]).	56
Figura 3-23: Modelo conceptual propuesto para el dominio de proceso.	57
Figura 4-1: Principales términos, atributos y relaciones del marco C-INCAMI.	70
Figura 4-2: Principales términos, atributos y relaciones de los componentes de Medición y de Evaluación enriquecidos con estereotipos del componente de Proceso.	74
Figura 4-3: Vista funcional y de comportamiento del proceso de M&E de GOCAME.	77
Figura 4-4: Flujo de actividades para <i>Definir los Requerimientos No Funcionales</i> .	79
Figura 4-5: Vista funcional y de comportamiento de la actividad <i>Establecer la Necesidad de Información</i> .	80
Figura 4-6: Tareas para <i>Especificar el Contexto</i> .	81
Figura 4-7: Tareas que intervienen para <i>Establecer el Árbol de Requerimientos</i> .	83
Figura 4-8: Modelo de Calidad en Uso definido en ISO 9126-1.	83
Figura 4-9: Árbol de Requerimientos del caso de estudio analizado [Covella, y otros, 2006].	83
Figura 4-10: Ejemplo de una <i>Especificación de Requerimientos No Funcionales</i> .	84
Figura 4-11: <i>Diseñar la Medición</i> .	86
Figura 4-12: Tareas de la actividad <i>Asignar Métricas</i> .	87
Figura 4-13: Especificación de la métrica utilizada para el atributo Satisfacción.	89
Figura 4-14: Actividad <i>Implementar la Medición</i> .	90
Figura 4-15: Vista funcional y de comportamiento para <i>Medir los Atributos</i> .	91
Figura 4-16: Actividades para <i>Diseñar la Evaluación</i> .	93
Figura 4-17: Actividades involucradas para <i>Identificar Indicadores Elementales</i> .	93

Figura 4-18: Especificación del Indicador Elemental <i>Grado de Cumplimiento</i> <i>Respecto a Proporción de Tareas Completadas sobre Tareas Propuestas.</i>	95
Figura 4-19: <i>Identificar Indicadores Derivados.</i>	96
Figura 4-20: Ejemplo de una <i>Especificación de Indicadores.</i>	98
Figura 4-21: Actividad <i>Implementar la Evaluación.</i>	99
Figura 4-22: Flujo de actividades para <i>Calcular Indicadores Elementales.</i>	100
Figura 4-23: Flujo de actividades para <i>Calcular Indicadores Derivados.</i>	101
Figura 4-24: Tareas intervinientes durante la actividad <i>Analizar y</i> <i>Recomendar.</i>	103
Figura 4-25: Extracto de las conclusiones obtenidas para el caso de estudio analizado.	104
Figura 4-26: Roles involucrados en GOCAME.	104
Figura 4-27: Vista organizacional del proceso de M&E de GOCAME.	106
Figura 4-28: Roles de las actividades A1, A2 y A3, asociados a las tareas que ejecutan y los productos de trabajo por los cuales son responsables.	108
Figura 4-29: Roles de las actividades A4, A5 y A6, asociados a las tareas que ejecutan y los productos de trabajo por los cuales son responsables.	109
Figura 4-30: Composición del artefacto <i>Especificación de los Requerimientos</i> <i>No Funcionales.</i>	110
Figura 4-31: Composición del artefacto <i>Especificación de Métricas.</i>	111
Figura 4-32: Composición del artefacto <i>Especificación de Indicadores.</i>	112
Figura 4-33: Capturas de pantalla de la herramienta C-INCAMI_Tool.	114
Figura 4-34: Arquitectura de la aplicación C-INCAMI_Tool.	115
Figura 5-1: Ciclos del proceso de SIQinU: (1) mejora del sistema, (2) re- evaluación de Calidad Externa, y (3) re-evaluación de Calidad en Uso.	118
Figura 5-2: Vista de comportamiento del proceso de SIQinU.	120
Figura 5-3: Fase (Ph) I: <i>Specify Requirements and Evaluation Criteria for QinU.</i>	122
Figura 5-4: Tareas involucradas en la actividad <i>Establish Information Need.</i>	123
Figura 5-5: Tareas involucradas en la actividad <i>Design Tasks.</i>	124
Figura 5-6: Vista informacional del artefacto <i>Tasks Specification.</i>	124
Figura 5-7: Actividad <i>Design QinU Measurement.</i>	125
Figura 5-8: Fase (Ph) II: <i>Perform QinU Evaluation and Analysis.</i>	127
Figura 5-9: Actividad <i>Collect Data.</i>	128
Figura 5-10: Fase (Ph) III: <i>Derive/specify requirements and evaluation criteria</i> <i>for EQ.</i>	129
Figura 5-11: Tareas de la actividad <i>Establish EQ Requirements Tree.</i>	130
Figura 5-12: Fase (Ph) IV: <i>Perform EQ Evaluation and Analysis.</i>	131

Figura 5-13: Fase (Ph) V: <i>Recommend, Perform Improvement Actions and Re-evaluate EQ</i>	132
Figura 5-14: Entradas y salidas de la actividad <i>Conduct Improvement Actions Analysis</i>	135
Figura 5-15: Alegoría de las tres capacidades de GOCAME, las cuales son reutilizadas en gran medida por SIQinU.....	136
Figura 5-16: Plantilla ETVX.....	138
Figura 5-17: Ejemplo de la plantilla de descripción de proceso utilizada en [Heidrich, y otros, 2006].....	138
Figura 5-18: Plantilla de especificación de proceso.	139
Figura 5-19: Especificación de la actividad Analizar y Recomendar (A6) utilizando la plantilla propuesta.....	140
Figura 6-1: Artefacto <i>Information Need Specification</i>	144
Figura 6-2: Ejemplo del documento <i>Tasks Specification</i> producido en la actividad <i>Design Tasks</i>	145
Figura 6-3: <i>QinU NFR Tree</i> para el caso JIRA.....	146
Figura 6-4: Artefacto <i>Metric Specification</i>	147
Figura 6-5: Parte del código Perl de la herramienta parser usada en JIRA.....	148
Figura 6-6: Pantalla SC1-2 en la cual se resaltan diversas cuestiones que podrían impactar en el valor del atributo <i>Sub-task correctness</i>	151
Figura 6-7: Pantalla SC3-1 en la cual se resaltan diversas cuestiones que podrían impactar en el valor del atributo <i>Sub-task correctness</i>	151
Figura 6-8: <i>EQ NFR Tree</i> derivado a partir del <i>QinU NFR Tree</i> y los problemas detectados en JIRA.	152
Figura 6-9: Especificación de la métrica para el atributo <i>Navigability feedback Completeness</i> (codificado 1.1.1.1.1 en la Figura 6-8).	153
Figura 6-10: Extracto del <i>Recommendations report</i> para el caso de estudio JIRA.....	155
Figura 6-11: Extracto del documento <i>Improvement Plan</i> producido en la actividad <i>Design Improvement Actions</i>	155
Figura 6-12: Relaciones hipotéticas de influencia entre los atributos de <i>Calidad Externa</i> y los de <i>Calidad en Uso</i>	159
Figura 6-13. Roles involucrados en cada fase de SIQinU.....	160
Figura 7-1: Vistas funcional y de comportamiento de la actividad <i>Medir los atributos</i>	167
Figura 7-2: Conceptos del marco C-INCAMI involucrados en la medición.....	168
Figura A-1: Ejemplo de un modelo de proceso utilizando IDEF0.	184
Figura A-2: Algunos símbolos de los Diagramas de Actividad de UML.	185
Figura A-3: Algunos símbolos de BPMN.	186

Figura A-4: Marco de trabajo de SPEM 2.0 (adaptado de [OMG-SPEM, 2008]).....	187
Figura A-5: Extracto del Modelo conceptual de SPEM (adaptado de [OMG-SPEM, 2008]).....	190
Figura A-6: Iconos de los principales elementos de modelado de SPEM.....	191

Índice de Tablas

Tabla 2-1: Vistas de modelado de los procesos y lenguajes base aplicables.	20
Tabla 3-1: Productos de trabajo producidos en las tareas de las principales actividades del proceso de medición de ISO 15939.....	37
Tabla 3-2: Productos de trabajo producidos en las prácticas específicas para el área de proceso Medición y Análisis.	41
Tabla 3-3: Definiciones de los conceptos del marco conceptual de proceso.....	58
Tabla 3-4: Definiciones de las relaciones del marco conceptual de proceso.....	61
Tabla 3-5: Definiciones de los atributos del marco conceptual de proceso.....	62
Tabla 4-1: Ontología para Medición y Evaluación: glosario abreviado de términos (ver [Olsina, y otros, 2008b] para más detalles).....	68
Tabla 4-2: Resultados obtenidos por los usuarios para la métrica <i>Número de Tareas Completadas Correctamente</i>	91
Tabla 4-3: Resultados obtenidos por los usuarios para la métrica <i>Proporción de Tareas Completadas sobre Tareas Propuestas</i>	92
Tabla 4-4: Medidas obtenidas para cada uno de los atributos al <i>Implementar la Medición</i>	92
Tabla 4-5: Asignación de operadores lógicos y de pesos.	97
Tabla 4-6: Valores obtenidos al <i>Calcular Indicadores Elementales</i>	100
Tabla 4-7: Resultados de los indicadores para el caso <i>Qplus Campus Virtual</i>	102
Tabla 4-8: Definición de los roles de GOCAME.....	105
Tabla 6-1: Definiciones de correcto e incorrecto usadas para el caso de estudio.	146
Tabla 6-2: Nivel de satisfacción alcanzado por JIRA respecto de <i>Actual Usability (Calidad en Uso)</i>	149
Tabla 6-3: Nivel de satisfacción para el atributo <i>Sub-task correctness</i> a nivel de sub-tareas.	149
Tabla 6-4: Nivel de satisfacción de los atributos de JIRA analizados para <i>Actual Usability</i>	150
Tabla 6-5: Extracto del documento <i>QinU Preliminary analysis report</i>	150
Tabla 6-6: Nivel de satisfacción alcanzado por JIRA respecto de <i>External Quality</i>	154
Tabla 6-7: Comparación de Calidad Externa entre JIRA y JIRA B.....	156

Tabla 6-8: Nivel de satisfacción alcanzado por JIRA B respecto de <i>Actual Usability</i> (Calidad en Uso).	157
Tabla 6-9: Comparación entre JIRA y JIRA B respecto del nivel de satisfacción alcanzado por los atributos de <i>Actual Usability</i>	158
Tabla 6-10. Productos de trabajo producidos en cada fase de SIQinU.	161

Capítulo 1: Introducción

1.1. Introducción

Se dice que una organización es tan eficiente como lo son sus procesos. Tanto grandes como pequeñas organizaciones, ya sea que desarrollen productos o brinden servicios de software/web, si desean que estos sean de calidad deberían contar con procesos que sean de calidad. Esto se debe a que existe una relación de influencia entre la calidad de los procesos y la calidad de los productos desarrollados [Humphrey, 1989]. Como se indica en [ISO/IEC, 2001a], la calidad del proceso contribuye a mejorar la calidad del producto. De hecho, hay evidencia empírica que soporta esta relación de causa-efecto [Gibson, y otros, 2006].

En base a lo expuesto, la tendencia de las organizaciones de hoy debiera ser apuntar a la gestión de sus procesos. En [Colla, 2009] se menciona que “la clave para la exitosa ejecución de desarrollos de software, esto es, con la calidad esperada, en tiempo y dentro del presupuesto, es la efectiva gestión de las acciones a partir de un proceso definido”. Al mismo tiempo comenta que este concepto no es otro que la premisa conceptual en la que se basa CMMI (*Capability Maturity Model Integration*), donde se considera que “la calidad de un producto de software está mayormente impulsada por la calidad del proceso utilizado para desarrollar y mantenerlo” [Paulik, y otros, 1995]. CMMI es un modelo de mejora de proceso (en inglés *Software Process Improvement*, SPI) que permite clasificar a las organizaciones según sus niveles de madurez. Estos niveles sirven para conocer la madurez de las capacidades de los procesos que se llevan a cabo para producir y mantener software.

Teniendo en claro que contar con procesos definidos es esencial, es importante acordar qué es un proceso desde el punto de vista ingenieril. Existen muchas definiciones sobre el término proceso. Algunas de las más aceptadas son:

- ✓ “Conjunto de actividades interrelacionadas, que transforman entradas en salidas, para alcanzar un objetivo dado” [CMMI Product Team, 2010].
- ✓ “Secuencia de trabajo, operaciones, o eventos que usualmente consumen tiempo, experiencia, u otros recursos y producen algún resultado” [OMG-SPEM, 2008].
- ✓ “Conjunto de actividades interrelacionadas o que interactúan, las cuales transforman entradas en salidas” [ISO/IEC, 2008].
- ✓ “Conjunto de actividades mutuamente relacionadas o que interactúan, las cuales transforman elementos de entrada en resultados” [CTN-ISSI, 2006].
- ✓ “Conjunto de actividades realizadas para un propósito dado” [Cooper, y otros, 2002].
- ✓ “Conjunto parcialmente ordenado de actividades llevadas a cabo para gestionar, desarrollar y mantener sistemas software” [Acuña, y otros, 2001a].

Claramente la mayoría de las definiciones se centran en decir, de manera explícita, que un proceso es un conjunto de actividades interrelacionadas que producen un resultado. Sin embargo, la realidad es que un proceso es más amplio. Como indica [Pall, 1987], un proceso comprende además una “organización lógica de personas, materiales, energía, equipamiento, y procedimientos en actividades de trabajo diseñadas para producir un resultado final específico”. Por ende, un proceso también involucra agentes, herramientas y métodos (entre otros recursos) para llevar a cabo las actividades. Entonces, en nuestra visión se podría definir un proceso de la siguiente manera:

“Un conjunto de actividades interrelacionadas, que consumen y producen productos de trabajo para lograr un objetivo particular, y que son llevadas a cabo en un orden lógico por agentes (humanos o automáticos) en cumplimiento de un rol siguiendo un método y utilizando herramientas”.

Teniendo en mente la diversidad de elementos que abarca un proceso (actividades, roles, productos de trabajo, etc.), se puede entender que especificar (o describir) de manera completa un proceso no es una tarea trivial. A menudo las descripciones textuales o informales de los procesos suelen ser complejas, extensas, difíciles de analizar y comprender y, en ocasiones, hasta confusas y tediosas. Esto hace que muchas veces las descripciones de los procesos no se actualicen cuando surgen cambios, se introduzcan inconsistencias, o directamente se decida no documentar los procesos, impidiendo así la repetibilidad y la reproducibilidad de las actividades y resultados. Con el objetivo de mitigar estos problemas, surge el modelado de procesos.

Modelar los procesos implica identificar cuáles son las principales características de las actividades realizadas para llevar a cabo un objetivo. Por lo tanto, un modelo de proceso es una representación abstracta de un proceso [Feiler, y otros, 1993] y, por ende, este puede ser modelado en diferentes niveles de abstracción. Además, dependiendo del propósito del modelo, el proceso puede modelarse desde diferentes puntos de vistas o perspectivas [Curtis, y otros, 1992; Verlage, 1994], enfocándose así en algunos aspectos más que en otros, como por ejemplo: la secuencia de actividades, las entradas y salidas, los roles, etc.

Por otra parte, los procesos se modelan usando lenguajes de modelado de proceso (PML, *Process Modeling Languages*) como BPMN (*Business Process Model and Notation*) [OMG-BPMN, 2011], Diagramas de Actividad de UML (*Unified Modeling Language*) [OMG-UML, 2010] y SPEM (*Software & Systems Process Engineering Meta-Model*) [OMG-SPEM, 2008], por citar algunos de los más usados en la industria y la academia. Utilizar lenguajes de modelado de proceso contribuye al entendimiento, a la facilidad para editar y mantener una descripción, simular, comunicar, auditar, etc. Sin embargo, no todos los lenguajes de modelado son apropiados para modelar todas las vistas [Acuña, y otros, 2005].

En vista de lo anterior, las organizaciones de software/web debieran prestar suma atención a la definición y especificación de sus procesos principales (Adquisición, Desarrollo, Mantenimiento, etc.), como así también de aquellos procesos de soporte (Gestión de la Documentación, Gestión de la Configuración, Aseguramiento de la Calidad, etc.) y organizativos (Gestión, Recursos Humanos, etc.) [ISO/IEC, 2004]. Por ejemplo, respecto a la importancia de los procesos de

soporte, en una encuesta realizada para comprender el estado de las prácticas de medición [Goldenson, 2007], el 63% de los encuestados, que trabajan en organizaciones que poseen un nivel de madurez 4 o 5 (según CMMI), afirman que los procesos de medición contribuyen a mejorar la calidad de sus productos, y solo el 7% dice que no ayuda, la empeora o raramente contribuye.

CMMI, uno de los modelos referentes en el área de procesos, establece en su nivel de madurez 2 al área de proceso de Medición y Análisis [CMMI Product Team, 2010]. Este área de proceso favorece, en dicho nivel, a que todos los proyectos de la organización hayan asegurado que los requerimientos son gestionados y que los procesos se planifican, ejecutan, miden y controlan. Por su parte, en el estándar ISO 25010 [ISO/IEC, 2011] se definen modelos de calidad (interna, externa y en uso) para su instanciación en procesos de medición y evaluación de requerimientos no funcionales.

Además, los procesos de Medición, junto con los de Evaluación y Análisis son procesos muy importantes debido a que brindan información útil para poder comprender, mejorar o predecir situaciones de interés. En la encuesta citada anteriormente se muestra que el 54% de los encuestados considera que contar con procesos de Medición ha permitido que sus organizaciones (de nivel 4 y 5) tomaran mejores decisiones tácticas, y el 49% indica que mejoró sus decisiones estratégicas. También en [Kasunic, 2006] se muestra que el 44,5% de los 1852 encuestados utilizan frecuentemente la Medición para comprender la calidad de sus productos y el 39% lo hace ocasionalmente.

En vista de la importancia de los procesos de Medición, Evaluación y Análisis, las organizaciones deben entonces establecer programas de Medición y Evaluación como parte de la tarea de gestionar la calidad dentro de sus líneas de producción, mantenimiento y evolución de productos y servicios. A su vez, estos procesos deben ser modelados adecuadamente, y así promover la repetibilidad y la reproducibilidad.

1.2. Motivación

Para contar con programas de Medición y Evaluación¹ que permitan conocer y mejorar la calidad de los productos y servicios, las organizaciones deben establecer claramente un conjunto de actividades y métodos para especificar, recolectar, almacenar y usar consistentemente los resultados de la medición y de la evaluación. Además, es necesario que las medidas (en Medición) y los valores de indicadores (en Evaluación) sean repetibles y comparables entre diferentes proyectos. De esta manera, los ulteriores procesos de análisis de resultados y toma de decisiones serán más robustos y completos. Para asegurar repetibilidad y consistencia en los resultados es necesario contar con un enfoque o estrategia de M&E basado en al menos tres pilares o capacidades [Olsina, y otros, 2008b], a saber:

1. un *marco conceptual*,
2. una *especificación (de las vistas) del proceso*, y
3. *métodos y herramientas*.

¹ De ahora en más se usará el acrónimo M&E para Medición y Evaluación.

El primer pilar indica que, para diseñar e implementar un programa de M&E robusto, es necesario hacer uso de un *marco conceptual* de M&E que sea flexible y terminológicamente consistente. Esto es necesario debido a que las organizaciones, muchas veces, por no contar con un marco conceptual con estas características, no ponen la debida atención a la forma en que los requerimientos no funcionales, métricas e indicadores deben ser diseñados, guardados y analizados a lo largo del tiempo, y por lo tanto tienen que comenzar desde cero sus programas de medición. Un marco bien establecido debería estar construido sobre una base conceptual (como por ejemplo: glosarios, taxonomías u ontologías). Una base conceptual es robusta si especifica de manera formal y explícita, los componentes, conceptos, relaciones y restricciones acordadas para un dominio particular. Esta capacidad asegura la uniformidad terminológica entre las otras capacidades y la consistencia de los resultados. Algunos de los marcos conceptuales propuestos para el dominio de M&E son: [Kitchenham, y otros, 2001], [Abran, y otros, 2002], [Olsina, y otros, 2004], y [García, y otros, 2004].

La segunda capacidad, es decir, la *especificación del proceso* de M&E, prescribe (o informa) mediante modelos el conjunto de actividades principales, las entradas y salidas, roles e interdependencias, entre otros aspectos. Generalmente, un modelo de proceso dice qué se debe hacer, pero no cómo debe hacerse, es decir, no especifica qué métodos, técnicas o herramientas deben ser utilizados para llevar a cabo las actividades. Una especificación robusta del proceso de M&E debe incluir modelos del proceso desde diferentes vistas o perspectivas. De esta manera, se asegura la repetibilidad y reproducibilidad en la implementación de las actividades y resultados y, además, se facilita el entendimiento y comunicación entre los interesados. La Organización Internacional de Normalización (*International Organization for Standardization, ISO*) publicó originalmente dos estándares relacionados con los procesos de M&E, a saber: ISO 15939 [ISO/IEC, 2002a], el cual se enfoca en el proceso de medición, e ISO 14598, donde se analiza el proceso para evaluadores en su quinta parte [ISO/IEC, 1998].

En la práctica, existen muchas situaciones donde el tamaño, la complejidad, duración y otros factores requieren que la ejecución de un proceso sea soportada por métodos y herramientas [ISO/IEC, 2002b]. La tercer capacidad, los *métodos y herramientas* de M&E, permite llevar a cabo y automatizar las descripciones de las actividades prescriptas/informadas en un modelo de proceso. Existen diferentes categorías de métodos según el tipo de actividad a realizar (inspección, testing, simulación, etc.). Los métodos deben ser asignados de una forma flexible para realizar las actividades especificadas y son usualmente automatizados por herramientas.

Como se puede apreciar, estas tres capacidades o pilares permiten desarrollar programas de M&E en los cuales se conozca claramente qué hacer (*proceso*), cómo hacerlo (*métodos y herramientas*) y que exista un entendimiento común de los conceptos claves (*marco conceptual*). Teniendo presente estas tres capacidades, decimos que una estrategia de M&E es *integrada* si está conformada al menos por: *i*) un modelo del dominio de M&E que define todos los conceptos necesarios y sus relaciones, *ii*) la definición de un proceso de M&E, y *iii*) un soporte metodológico y tecnológico que dé soporte a las actividades.

En la literatura se pueden encontrar muchas estrategias o enfoques de M&E, por ejemplo: *Goal Question Metric (GQM)* [Basili, y otros, 1994], *Practical Software*

and Systems Measurement (PSM) [McGarry, y otros, 2001], *Goal-Oriented Context-Aware Measurement and Evaluation (GOCAME)* [Olsina, y otros, 2008b], *Framework for the Modeling and Evaluation of Software Processes (FMESP)* [García, y otros, 2006b], etc. Sin embargo, no todas son estrategias integradas de M&E [Olsina, y otros, 2011], ya que no cuentan con las tres capacidades a la vez y, si lo hacen, muchas veces estas no están relacionadas. De nada sirve definir una base conceptual, y luego al momento de definir el proceso utilizar términos que no se encuentran en dicha base o se utilizan de manera distinta. También cabe mencionar que, teniendo presente que el proceso es fundamental al momento de saber qué es lo que se debe hacer para lograr los objetivos, por ejemplo, asegurar la calidad, generalmente este pilar no suele tenerse presente o no es modelado/especificado con cierto nivel de detalle. Esta debilidad se observa, por ejemplo, en la estrategia GOCAME documentada en [Olsina, y otros, 2008b], donde el proceso de M&E está especificado de una manera informal y solo a un alto nivel. Estas cuestiones hacen que una estrategia de M&E pueda llegar a ser difícil de implementar, e incluso que los agentes que deben llevar a cabo las actividades de M&E no tengan el perfil necesario para hacerlo, acarreando todo esto al potencial fracaso del proyecto de M&E.

1.3. Contribución

Teniendo presente la problemática planteada al final de la sección anterior, el presente trabajo contribuye en:

- ✓ Presentar, haciendo foco en el proceso, una Estrategia Integrada de M&E multi-propósito (denominada GOCAME –por sus siglas en inglés de *Goal-Oriented Context-Aware Measurement and Evaluation*), que sirve como base para definir otras estrategias de M&E de propósito específico (como SIQinU –por sus siglas en inglés de *Strategy to Improve Quality in Use*).
- ✓ Modelar el proceso de M&E de GOCAME desde diferentes vistas (funcional, organizacional, de comportamiento e informacional), y así robustecer la especificación del proceso presentado en [Olsina, y otros, 2008b]. Modelar las diferentes vistas proveerá una visión completa, detallada e integrada del proceso. Esto permite tener una guía clara y repetible de las actividades que deben llevarse a cabo para definir los requerimientos no funcionales, establecer el contexto, y diseñar e implementar la medición, la evaluación, y el análisis y la recomendación, con el fin de soportar un proceso de toma de decisiones más robusto. También, esto facilita la comunicación y la realización de las tareas de las partes intervinientes, como así también la selección del personal asociado a las actividades en función del rol que deben cumplir.
- ✓ Emplear una terminología consistente en los modelos de proceso de M&E utilizando una base conceptual robusta de M&E y de procesos.
- ✓ Facilitar la definición de métodos y herramientas que permitan llevar a cabo las actividades del proceso, ya sea de forma total o parcialmente automatizada. Particularmente, se comentará una herramienta de soporte al proceso de M&E, la cual puede usarse tanto en GOCAME como en SIQinU.
- ✓ Identificar los principales productos de trabajo del proceso, como así también, especificar claramente la estructura de los mismos. Esto permite saber, no solo qué se debe hacer, sino también qué productos de trabajo son necesarios y

cuáles deben producirse en cada actividad o tarea, y cómo se componen los mismos.

- ✓ Proveer una plantilla que permita especificar de manera completa una actividad. Se tendrá en cuenta el principio W5H (*Who, What, Where, When, Why, How*). Esto permite reunir toda la información relacionada a una actividad en un solo documento. La plantilla permite conocer: el objetivo de la actividad (*why*), la descripción de la actividad y productos de trabajo necesarios (*what*), la secuencia de las actividades (*when*), los roles responsables de la ejecución de la actividad (*who*), el lugar donde se realizará la actividad (*where*), y los métodos, técnicas, guías y herramientas asociados a la actividad (*how*).

Se considera que este trabajo permitirá a una organización llevar a cabo proyectos y programas de M&E más eficaces, repetibles y robustos. Además, contar con vistas del proceso de M&E ayuda a reducir el esfuerzo de puesta en marcha asociado a aspectos técnicos como son el diseño y especificación de procesos. En consecuencia, las organizaciones podrán concentrarse directamente en la implementación de las actividades de M&E de las entidades de interés de sus proyectos de software y web en pos de alcanzar sus objetivos.

1.4. Publicaciones Relevantes

Los resultados de las investigaciones referidas a esta tesis fueron publicados en congresos y revistas, tanto de nivel nacional como internacional. A continuación se listan las principales publicaciones relacionadas al presente trabajo. Las mismas aparecen ordenadas por fecha de publicación de manera decreciente, y divididas entre publicaciones internacionales y nacionales.

1.4.1. Publicaciones en Revistas (Journals) Internacionales

- ✓ *Specifying Process Views for a Measurement, Evaluation and Improvement Strategy* (**Becker P.**, Lew P., Olsina L.). En *Advances in Software Engineering Journal*, Academic Editor: Osamu Mizuno, Hindawi Publishing Corporation, Vol. 2012, 27 pp., DOI:10.1155/2012/949746, 2012.
- ✓ *An Integrated Strategy to Systematically Understand and Manage Quality in Use for Web Applications* (Lew P., Olsina L., **Becker P.**, Zhang L.). En *Requirements Engineering Journal*, Special Issue on Quality Requirement Engineering for Systems & Architecting, Springer London, Vol.17, N. 4, pp. 299-330, DOI 10.1007/s00766-011-0128-x, ISSN 0947-3602, 2012.
- ✓ *Measurement and Evaluation as Quality Driver* (**Becker P.**, Molina H., Olsina L.). En *Journal ISI (Ingénierie des Systèmes d'Information)*, Special Issue "Quality of Information Systems", Hermes, Lavoisier, vol. 15/6, pp. 33-62, Paris, France, DOI:10.3166/ISI.15.6.33-62, ISSN 1633-1311, 2010.

1.4.2. Publicaciones en Congresos Internacionales

- ✓ *Process Conceptual Base for Enriching a Measurement and Evaluation Ontology* (**Becker P.**, Papa M.F., Olsina L.). En CD de la XVII Conferencia Iberoamericana en Software Engineering (CibSE'14). ISBN: 978-956-236-247-4, pp. 53-66, Pucón, Chile, 2014.
- ✓ *Enhancing the Conceptual Framework Capability for a Measurement and Evaluation Strategy* (**Becker P.**, Papa M.F., Olsina L.). En ICWE 2013 Workshops,

LNCS 8295, Q.Z. Sheng and J. Kjeldskov (Eds), Springer International Publishing Switzerland, pp. 104-116, Aalborg, Denmark, 2013.

- ✓ *Assessing Integrated Measurement and Evaluation Strategies: A Case Study* (Olsina L., Papa M.F., **Becker P.**). En 7º Central & Eastern European Software Engineering Conference CEE-SEC(R), IEEE Xplore, ISBN: 978-1-4673-0843-4, pp. 1-10, Moscow, Russia, 2011.
- ✓ *Strategy to Improve Quality for Software Applications: A Process View* (**Becker P.**, Lew P., Olsina L.). En ACM proceedings of ICSE, Int'l Conference of Software and System Process (ICSSP), ISBN: 978-1-4503-0730-7, DOI 10.1145/1987875.1987897, pp. 129-138, Honolulu, Hawaii, USA, 2011.
- ✓ *Towards Support Processes for Web Projects* (**Becker P.**, Olsina L.). En LNCS 6385, Springer book: Current Trends in Web Engineering - 10th International Conference on Web Engineering, ICWE 2010 Workshops, Daniel and Faca (Eds), ISBN: 978-3-642-16984-7, DOI: 10.1007/978-3-642-16985-4_10, pp. 102-113, Vienna, Austria, 2010.
- ✓ *Integrando Proceso y Marco de Medición y Evaluación* (**Becker P.**, Molina H., Olsina L.). En Memorias de la XII Conferencia Iberoamericana de Software Engineering (IDEAS09-CIbSE), ISBN 978-958-44-5028-9. pp. 261-266, Medellín, Colombia, 2009.

1.4.3. Publicaciones en Congresos Nacionales

- ✓ *Vista Funcional del Proceso de Medición y Evaluación* (**Becker P.**, Molina H., Olsina L.). En ASSE'10, Simposio Argentino de Ingeniería de Software (39 JAIIO), pp. 294-308, ISSN: 1850-2792, 2010.
- ✓ *Estrategias de Medición y Evaluación: Diseño de un Estudio Comparativo* (Papa F., **Becker P.**, Olsina L.). En ASSE'10, Simposio Argentino de Ingeniería de Software (39 JAIIO), pp. 323-337, ISSN: 1850-2792, 2010.
- ✓ *Modelado de Proceso para un Marco de Medición y Evaluación* (**Becker P.**, Olsina L., Molina, H.). En Actas (CD-ROM) del XIV Congreso Argentino de Ciencias de la Computación (CACIC 2008), Workshop WISBD, Chilecito, La Rioja, ISBN 978-987-24611-0-2, 2008.

1.5. Organización de la Tesis

En lo que sigue, esta tesis se estructura de la siguiente manera:

Capítulo 2: tiene por objetivo dar al lector una base sobre distintos aspectos que serán útiles para comprender mejor los capítulos siguientes. En primer lugar se resalta la importancia de los procesos de M&E en las organizaciones de Software y la necesidad de contar con estrategias de M&E. Luego se exponen los aspectos fundamentales referentes a Estrategias Integradas de M&E, haciendo énfasis en los tres pilares que consideramos que una estrategia de M&E debe tener de manera integrada para ser eficaz, repetible y robusta, a saber: un proceso, un marco conceptual, y métodos y herramientas. Por otro lado, se introduce al lector en aspectos relacionados al modelado de proceso. Se muestran los beneficios del modelado de proceso, las diferentes vistas o perspectivas que se pueden tener presente al modelar un proceso, y se abordarán algunos conceptos básicos de SPEM, el lenguaje de modelado de proceso que se utiliza en esta tesis.

Capítulo 3: se presenta el estado del arte relacionado a Estrategias Integradas de M&E y sus procesos. En primer lugar se comentan diferentes trabajos que aportan enfoques o estrategias para M&E y se discute si pueden ser consideradas integradas o no. También, teniendo presente que el foco de esta tesis es la especificación de las vistas del proceso de M&E, se discute sobre la forma en la que diferentes autores modelan el proceso de M&E. Además, en consideración de la variedad de términos y significados que existen en el área de proceso, se mencionan distintos marcos conceptuales para el dominio de proceso y se presenta un nuevo marco conceptual basado en algunos de los trabajos revisados. Este nuevo marco conceptual permite, entre otras cosas, aclarar el significado que se dará en esta tesis a los distintos términos referidos al área de proceso. Por último se comenta cuál es la solución que se propone en este trabajo en función de las debilidades y oportunidades de mejoras encontradas en el estado del arte.

Capítulo 4: se detalla, desde el punto de vista de sus tres capacidades, la Estrategia Integrada de M&E de propósito general denominada GOCAME, y se pone principal énfasis en la especificación de las vistas del proceso de M&E. Además se discute la relación que existe entre los diferentes términos del marco conceptual de M&E utilizado en GOCAME y los del marco conceptual de proceso.

Capítulo 5: se especifica la Estrategia Integrada de M&E de propósito específico llamada SIQinU, abordando sus tres capacidades y resaltando, principalmente, su proceso visto desde diferentes perspectivas. Debido a que SIQinU se deriva de GOCAME, en este capítulo se muestra la alineación existente entre estas dos estrategias. Además se presentará una plantilla, basada en el principio W5H, que permite especificar de manera completa las actividades del proceso.

Capítulo 6: se considera un caso de estudio realizado en una empresa real, siguiendo el proceso de la estrategia SIQinU y ejemplificando los diferentes productos de trabajo de entrada y salida.

Capítulo 7: se presentan las conclusiones y posibles trabajos futuros de investigación.

Apéndice A: se comentan algunos lenguajes de modelado de proceso.

Capítulo 2: Fundamentos sobre Estrategias Integradas de M&E y sobre Procesos

Este capítulo pretende, por un lado, mostrar al lector la relevancia del área de M&E, y la necesidad de disponer de Estrategias de M&E para establecer programas y proyectos de M&E que sean eficaces, repetibles y robustos. Además, se consideran las tres capacidades que debe tener, de manera integrada, una Estrategia de M&E para lograr programas y proyectos de M&E con tales características.

Por otro lado, teniendo en cuenta que en este trabajo nos concentraremos en la especificación de procesos de M&E, también se introducirá al lector en aspectos relacionados al modelado de proceso. Por ejemplo, se indicarán los beneficios de contar con modelos de proceso y las diferentes vistas o perspectivas que se pueden tener presente al modelar un proceso. Además se abordarán conceptos básicos de SPEM por ser el lenguaje de modelado de proceso utilizado en esta tesis.

2.1 Fundamentos sobre Estrategias de M&E

Hoy en día es de gran importancia que las organizaciones cuenten con un área de M&E, ya que permite, entre otras cosas, asegurar la calidad de los productos y procesos, dar soporte a otras áreas y tomar decisiones basadas en conocimiento. Pero además de contar con áreas de M&E, también es fundamental disponer de una Estrategia Integrada de M&E que permita desarrollar programas y proyectos eficaces, repetibles y robustos.

A continuación se destacará la importancia que diversos autores/organizaciones le dan al área de M&E, y luego se pondrá el foco en el significado que se da en este trabajo al concepto “Estrategia Integrada de M&E”.

2.1.1 Relevancia de M&E en las organizaciones

Un aporte notable para la Ingeniería de Software es la Guía SWEBOK (*Software Engineering Body of Knowledge*). Esta guía tiene por objetivo desarrollar un consenso internacional sobre el conocimiento generalmente aceptado en el dominio de Ingeniería de Software. En la edición 2004 de la Guía SWEBOK [IEEE, 2004] se definen 10 áreas de conocimiento, donde las primeras 5 áreas se enfocan en los procesos principales de Ingeniería de Software: i) *Requerimientos de Software*, ii) *Diseño de Software*, iii) *Construcción de Software*, iv) *Pruebas de Software* y v) *Mantenimiento del Software*; mientras que las restantes se centran en los procesos organizacionales y de soporte: vi) *Configuración de Software*, vii) *Gestión de la Ingeniería de Software*, viii) *Proceso de la Ingeniería de Software*, ix) *Herramientas y Métodos de la Ingeniería de Software*, y x) *Calidad de Software*. A su

vez, cada una de estas áreas de conocimiento se divide en tópicos. Por ejemplo, el área de conocimiento *Proceso de la Ingeniería de Software*, el cual se enfoca en la definición, implementación, gestión, cambio y mejora, entre otras, de los procesos de Ingeniería de Software, se compone de cuatro tópicos, dónde uno de ellos se centra en la medición de la calidad de los productos software y de los procesos.

La inclusión de un tópico centrado en Medición no es simple casualidad, sino una necesidad ya que la Medición es, desde hace tiempo, un área fundamental para mejorar la gestión de las prácticas de la Ingeniería de Software. Incluso, en [Abran, y otros, 2004] se propone agregar a la Guía SWEBOK un área adicional de conocimiento llamada *Medición de Software* en vista de que es uno de los tres temas comunes, junto a *Herramientas* y *Calidad*. En la Figura 2-1 se muestra la taxonomía inicial que proponen los autores para el nuevo área de conocimiento. La guía citada menciona que la medición efectiva se ha convertido en una de las piedras angulares de la madurez de las organizaciones y que la gestión sin medición, ya sea cualitativa o cuantitativa, sugiere una falta de rigor al momento de tomar decisiones.

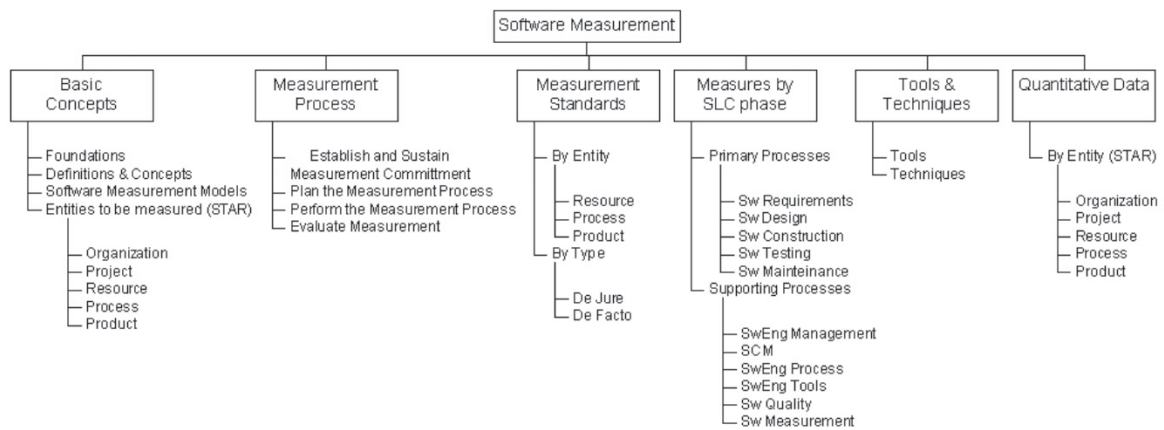


Figura 2-1: Taxonomía propuesta en [Abran, y otros, 2004] para un posible Área de Conocimiento de Medición de Software.

Afirmar que contar con un área de Medición es esencial para una organización que quiere alcanzar la madurez no es algo fuera de lo común si se tienen en cuenta los niveles de madurez definidos en CMMI. Este estándar *de facto* define 5 niveles de madurez, a saber: 1) Inicial –*Initial*-, 2) Gestionado –*Managed*-, 3) Definido –*Defined*-, 4) Gestionado Cuantitativamente –*Quantitatively Managed*- y 5) En Optimización –*Optimizing*. El nivel 4 (Gestionado Cuantitativamente) implica que la organización cuenta con la recolección de medidas detalladas de la calidad de los productos y procesos de software [Software Engineering Laboratory, 1995]. Sin embargo, ya en el nivel 2 (Gestionado) se debe disponer de un área de *Medición y Análisis*. Este área de proceso da soporte a todas las demás áreas de proceso, y los resultados pueden ser utilizados en la toma de decisiones y para la elección de acciones correctivas apropiadas [CMMI Product Team, 2010]. En la Figura 2-2 se puede apreciar que el área de proceso *Medición y Análisis*, al igual que las áreas de proceso *Gestión de Configuración y Aseguramiento de la Calidad del Proceso y del Producto*, se relacionan con todas las demás áreas de proceso.

Un ejemplo de la relación entre Medición y otras áreas podemos también observarlo en PMBOK (*Project Management Body of Knowledge*) [PMI, 2009], donde se la relaciona íntimamente con la *Gestión de Proyectos*. Las mediciones del

alcance del proyecto, costo y cronograma ayudan al equipo de dirección del proyecto a conocer el desempeño y el avance del mismo y a tomar decisiones sobre la necesidad de aplicar o no acciones preventivas o correctivas. Por ejemplo, la actividad *Monitorear y Controlar el Trabajo del Proyecto* implica recopilar, medir y distribuir la información relativa al desempeño, con el fin de permitir efectuar mejoras a sus procesos. El seguimiento continuo mediante actividades de Medición proporciona al equipo de dirección del proyecto conocimientos sobre la “salud” del proyecto y permite identificar las áreas susceptibles en caso de requerir una atención especial.

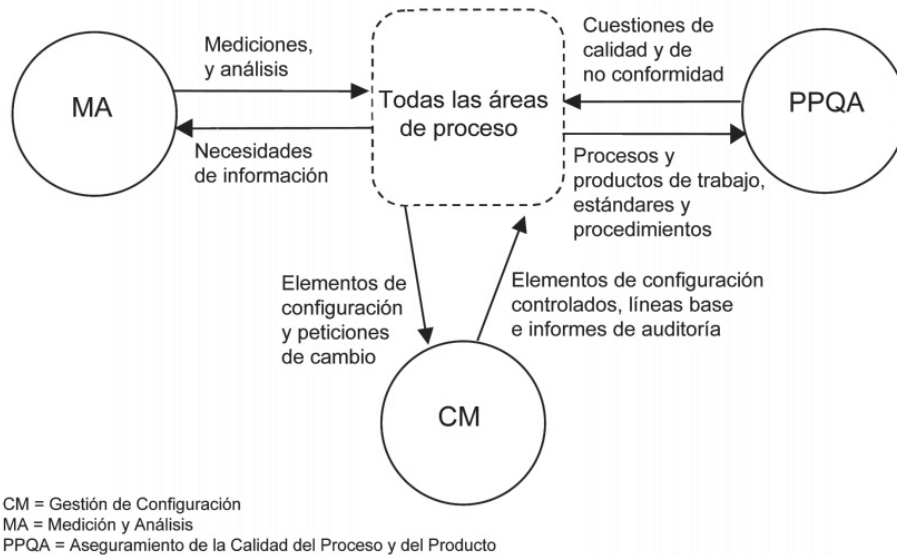


Figura 2-2: Áreas de proceso de soporte de CMMI y su relación con las demás áreas de proceso (tomado de [CMMI Product Team, 2010]).

Es importante destacar que, contar con actividades de Medición requiere también contar con actividades de Evaluación. La medición sin evaluación no es más que sólo un conjunto de valores o datos intermedios, los cuales no siempre son útiles para la toma de decisiones. Para notar este hecho, supongamos que debemos conocer si en nuestro proyecto debemos aumentar la productividad para llegar a tiempo con la entrega de los módulos solicitados por el cliente. Una medición que indique que hemos desarrollado completamente 30 módulos no nos permite decidir si debemos tomar acciones para aumentar la productividad. Incluso una medición que nos indique que 30 módulos equivalen al 78% de los módulos a desarrollar tampoco es completamente útil para tomar una decisión, aunque obviamente esta medida brinda más información que la anterior. Claramente la medición no es totalmente suficiente para tomar decisiones, ya que no produce información relativa a un propósito en un contexto dado. Sin embargo, la evaluación, haciendo uso de indicadores basados en una necesidad de información, permite interpretar los valores medidos y, así, brindar información útil para la toma de decisiones [Olsina, y otros, 2013]. El uso de indicadores contribuye a conocer el nivel de satisfacción alcanzado con el fin de mantener cierto grado de aceptabilidad en las entidades medidas, ya sean procesos, organizaciones, proyectos, recursos o productos. Siguiendo con el caso hipotético, una evaluación podría indicarnos cuan aceptable es haber desarrollado 30 módulos en base al calendario previsto. Poseer esta información, ahora si nos

permitirá saber si se debe aumentar o no la productividad para estar al día con lo programado.

Teniendo presente la importancia de contar con procesos de Medición y de Evaluación, ISO ha publicado los estándares ISO/IEC 15939 y 14598, los cuales se enfocan, respectivamente, en dichos procesos. Actualmente existe el proyecto SQuaRE (*Software Product Quality Requeriments and Evaluation*) que trata de crear una convergencia entre diferentes estándares ISO relacionados a la calidad de los productos software -tales como ISO 14598 y 9126. Este proyecto intenta eliminar las distancias, conflictos y ambigüedades entre los estándares previos, dando como resultado la serie de estándares 25xxx.

Respecto al proceso de Evaluación de la calidad del software, en [IRAM, 2006] se indica que su propósito es apoyar directamente el desarrollo así como la adquisición de software que satisfaga las necesidades del usuario y del cliente. El objetivo final es asegurarse de que el producto proporciona la calidad requerida, que satisface las necesidades explícitas e implícitas de los usuarios (incluyendo a los operadores, los receptores de los resultados del software y al personal de mantenimiento del software). Además, indica que la evaluación de la calidad de los productos intermedios permite:

- ✓ decidir sobre la finalización de un proceso y sobre cuándo enviar productos al proceso siguiente;
- ✓ predecir o estimar la calidad del producto final;
- ✓ decidir sobre la aceptación de un producto intermedio de un subcontratista; y
- ✓ recolectar información sobre los productos intermedios para controlar y administrar el proceso.

También, el mismo estándar agrega que la evaluación de la calidad del producto final sirve para:

- ✓ decidir sobre la aceptación del producto;
- ✓ decidir cuándo lanzar el producto;
- ✓ comparar un producto con los productos competidores;
- ✓ seleccionar un producto entre productos alternativos;
- ✓ determinar tanto los efectos positivos como los negativos del producto, cuando se lo utiliza; y
- ✓ decidir cuándo mejorar o sustituir un producto.

Finalmente, en [McGarry, y otros, 2001] se comentan los siguientes beneficios de contar con procesos de M&E:

- ✓ *Comunicar efectivamente*: La M&E proporciona información objetiva. Esto reduce la ambigüedad que a menudo existe en los proyectos de software. La M&E ayuda a los gerentes a identificar, priorizar, controlar y comunicar los objetivos en todos los niveles de la organización.
- ✓ *Seguir los objetivos específicos del proyecto*: La M&E puede describir con precisión el estado de los procesos y productos de los proyectos de software. Esto es clave para conocer objetivamente el progreso de las actividades del

proyecto y la calidad de los productos de software en todo el ciclo de vida del proyecto.

- ✓ *Identificar y corregir problemas en etapas tempranas:* La M&E facilita una estrategia de gestión proactiva. Los posibles problemas son identificados objetivamente como riesgos a ser analizados y manejados, y los problemas existentes pueden ser mejor analizados y priorizados. La M&E favorece el descubrimiento y corrección temprana de los problemas técnicos y de gestión que pueden ser más difíciles y costosos de resolver más tarde. Los gerentes utilizan la M&E como un recurso para anticiparse a los problemas y, así, evitar usar un enfoque reactivo.
- ✓ *Justificar decisiones:* Los gerentes de negocios, técnicos y de proyecto deben ser capaces de fundamentar sus estimaciones y planificaciones en base a datos históricos. También, deben ser capaces de justificar los cambios en los planes en base a las medidas de rendimiento actuales. La M&E proporciona un fundamento eficaz para la selección de las mejores alternativas.

Como se puede observar en los trabajos citados, los procesos de M&E son de vital importancia en la Ingeniería de Software. La M&E se incluye a lo largo de todo el ciclo de vida de desarrollo del software y da soporte a otros procesos de forma directa o indirecta. Sin embargo muchas veces se habla de Medición sin hacer mención de la Evaluación. Tal es el caso de CMMI que no posee actividades de evaluación en su área de proceso *Medición y Análisis* (ver Sección 3.1.2 para más detalles). Otro aspecto a tener en cuenta es que Medición y Evaluación son dos procesos que están estrechamente relacionados. No se puede realizar una Evaluación sin una previa Medición, y una Medición sin Evaluación no es más que sólo un conjunto de valores o datos con mínima información interpretada, la cual no siempre es útil a la hora de realizar análisis y tomar decisiones. Sin embargo, a pesar de esto, ISO trata cada uno de estos procesos en diferentes estándares, como si fueran disjuntos y pudieran existir el uno sin el otro (ver Sección 3.1.1 para más detalles). En vista de esto, se considera necesario contar con un proceso que integre y coordine las actividades de Medición junto con las de Evaluación para dar lugar a programas de M&E robustos y claramente definidos que permitan conocer de forma eficiente y coherente el estado relevante de cada entidad en proyectos de desarrollo de software y así producir información apropiada para la toma de decisiones.

Pero no solo es importante tener definido un proceso de M&E para conseguir proyectos y programas de M&E robustos, repetibles y consistentes. También es fundamental contar con *Estrategias Integradas de M&E*.

2.1.2 Estrategias Integradas de M&E

Como se comentó, los procesos de M&E desempeñan un papel clave en las organizaciones de Ingeniería de Software. Estos permiten controlar, mejorar y mantener aspectos relacionados a la calidad de los productos, como así también de los procesos. Las organizaciones de Software y Web que introducen programas de M&E, quizás como parte de un área de proceso de M&E, y/o como una estrategia de aseguramiento de calidad, necesitan establecer un conjunto de actividades y procedimientos para especificar, recolectar, almacenar y usar correctamente medidas (en la Medición) y valores de indicadores (en la Evaluación). También es

necesario asegurar, para propósitos de análisis, que las medidas y los valores de los indicadores sean repetibles, reproducibles y comparables entre distintos proyectos² de M&E, por lo tanto es importante que se almacenen metadatos apropiados de las métricas e indicadores. A raíz de lo mencionado, se considera que, para diseñar e implementar programas y/o proyectos de M&E que sean robustos y consistentes, es deseable disponer de una estrategia –o enfoque- de M&E que se base en los siguientes tres pilares o capacidades [Olsina, y otros, 2008b; Becker, y otros, 2010c]:

1. *un marco conceptual para el dominio de M&E* que sea flexible y consistente, preferentemente basado en una ontología para eliminar ambigüedades;
2. *una especificación de las vistas del proceso de M&E*, en la cual se describan cuáles son las principales actividades que deben ser planificadas y ejecutadas, sus entradas y salidas, roles participantes, entre otros aspectos; y,
3. *métodos y herramientas* específicas que permitan llevar a cabo las actividades del proceso de M&E.

Al momento de realizar una actividad de M&E, como puede ser ejecutar la medición de los atributos de un producto, muchas veces, los involucrados no saben exactamente qué insumos necesitan para llevar a cabo tal actividad, o qué es exactamente lo que deben producir. Peor aún, en ocasiones, directamente no se conoce puntualmente qué es lo que se debe hacer. Contar con una *especificación de las vistas del proceso* puede prevenir esta situación. Modelar el proceso desde diferentes vistas o perspectivas permite identificar qué se debe hacer, en qué orden, quiénes son los responsables, cuáles son los productos de trabajo necesarios y cuáles serán producidos. Aun más, un proceso de M&E bien especificado no solo permite el entendimiento del proceso, sino que también facilita la comunicación entre los interesados y, además, asegura la repetibilidad y la reproducibilidad en la implementación de las actividades -características deseables a lo largo de todo el proceso pero a menudo no observadas [Becker, y otros, 1997].

Por otro lado, la utilización de términos técnicos por varios individuos sin la misma base conceptual, generalmente, lleva a confusiones debido a que un término puede llegar a tener varias interpretaciones según el contexto. Sin una terminología concisa y consistente, es imposible una comunicación efectiva entre los analistas de la medición, los recolectores de las medidas y los usuarios de la información. Por lo tanto, contar con una terminología de medición consistente es mandatorio [McGarry, y otros, 2001]. El uso de un *marco conceptual* robusto (como una ontología) en el que explícitamente se definen todos los conceptos necesarios de un dominio y las relaciones entre ellos, promueve la facilidad de aprender y la replicabilidad de la estrategia a aplicar, asegura la uniformidad entre las otras capacidades, y brinda consistencia de los resultados.

Finalmente, según [Lavazza, 2000], una medición efectiva y eficiente requiere un soporte metodológico (conjunto de métodos) y tecnológico. Contar con un conjunto específico de *métodos* permite tener una guía clara de cómo deben

² Es deseable que, no solo se puedan comparar proyectos que pertenecen a una misma organización, sino que también se puedan efectuar comparaciones y análisis entre proyectos de distintas organizaciones.

realizarse las actividades especificadas. Aun más, utilizar *herramientas* automáticas o semiautomáticas para realizar tareas puede proveer una mayor confiabilidad en el resultado obtenido y un ahorro de tiempo. Estos tres pilares son representados alegóricamente en la Figura 2-3.

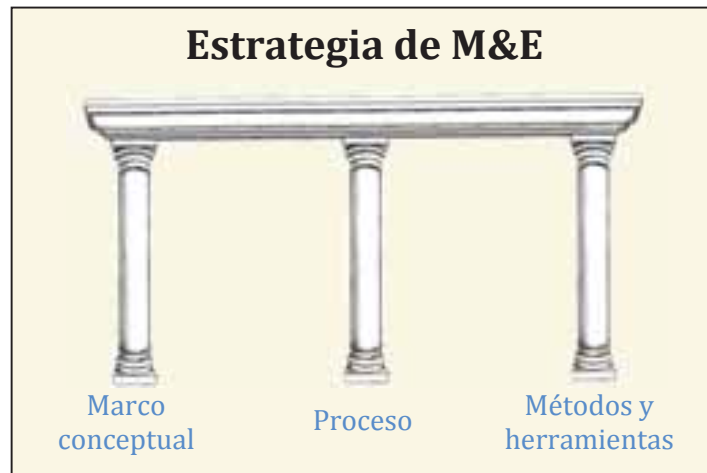


Figura 2-3: Alegoría de los tres pilares sobre los cuales una estrategia de M&E debería sostenerse.

Si una Estrategia de M&E cuenta con estos tres pilares -o capacidades- y además, las especificaciones del proceso de M&E usan la terminología del marco conceptual de M&E y, al mismo tiempo, las actividades tienen asociadas algún método (especificado también usando el marco conceptual de M&E) y/o herramienta que permita ejecutarlas, decimos que es una *Estrategia Integrada de M&E* [Olsina, y otros, 2011; Papa, 2012]. Tener presente estas tres capacidades permite desarrollar programas y/o proyectos de M&E en los cuales se sepa claramente qué hacer (*proceso*), cómo hacerlo (*métodos y herramientas*) y que exista un entendimiento común de los conceptos claves (*marco conceptual*).

Si bien existen muchos enfoques o estrategias de M&E, se puede decir, teniendo en cuenta lo antes mencionado, que no todas pueden considerarse *Estrategias Integradas de M&E*. En el Capítulo 3 se presenta una revisión de varias estrategias/enfoques de M&E reconocidas en la academia y/o la industria con el objetivo de analizar si reúnen simultáneamente los tres pilares comentados y pueden ser llamadas Estrategias Integradas de M&E.

2.2 Fundamentos sobre Modelado de Procesos

En todo proyecto de desarrollo de productos software, si se desea lograr un producto de calidad, es clave contar con un proceso de desarrollo. Además, si el proceso es de calidad, este permitirá, en gran medida, que se produzca un producto de calidad [Paulik, y otros, 1995]. Contar con procesos es fundamental para poder lograr los objetivos propuestos. Al establecer un programa de M&E, esta idea no deja de ser válida, ya que el hacer uso de un proceso bien definido para dicho programa permitirá tener una guía al momento de realizar actividades de especificación de requerimientos no funcionales, diseño y ejecución de mediciones y evaluaciones, para hacer un análisis consistente y, si fuera necesario, poder efectuar recomendaciones adecuadas.

El problema de los procesos es que estos, generalmente, comprenden muchas actividades, son difíciles de entender y/o están débilmente especificados. A raíz de

esto, a menudo se hacen descripciones textuales de los procesos y sus actividades, así como de sus elementos de entrada y salida. Sin embargo, estas especificaciones suelen ser realizadas con un bajo nivel de detalle, lo que las hace imprecisas, ambiguas, e incomprensibles en algunos casos. O por el contrario, son tan extensas que quien las lee termina desorientado entre tantas líneas de texto. Por ejemplo, en [Bröckers, y otros, 1995a], a partir del estudio de varios proyectos del mundo real, donde se habían utilizado puramente descripciones textuales, concluyen que estas no son fácilmente entendidas por los miembros del proyecto. Otro punto en contra que tienen las descripciones textuales es que, cuando surgen cambios, estos no son debidamente documentados. Con el fin de solucionar o reducir algunos de estos problemas surge el modelado de procesos.

El modelado de proceso trata de identificar cuáles son las principales características de las actividades realizadas para llevar a cabo un objetivo, como puede ser desarrollar un producto software, evaluar la calidad de un producto, entre otros. Por lo tanto, un modelo de proceso es una abstracción de un proceso [Feiler, y otros, 1993], es decir que, la totalidad del proceso es especificada en términos de las actividades, secuencias, roles y productos de trabajo que lo integran.

Generalmente, un modelo de proceso puede ser dividido en varios sub-modelos, los cuales muestran diferentes puntos de vistas o perspectivas del proceso [Curtis, y otros, 1992; Verlage, 1994], dependiendo de qué sea lo que se quiere resaltar. A continuación, se resaltan los beneficios de contar con modelos de proceso. Luego, en la Sección 2.2.2 se describen diferentes vistas de modelado y en la Sección 2.2.3 se comenta la importancia de escoger lenguajes apropiados para modelarlas. Finalmente, en la Sección 2.2.4, se presenta el lenguaje de modelado que se utilizará en esta tesis.

2.2.1 Beneficios de modelar los procesos

Al realizar un modelo, si se utiliza una representación gráfica, es decir, un diagrama de proceso, se pueden observar fácilmente las distintas interrelaciones que existen entre las actividades, dependencias, puntos de control y paralelismos, entre otros, dada su mayor expresividad respecto de los documentos textuales. Los diagramas también permiten observar con mayor facilidad la existencia de problemas y luego tomar acciones para corregirlos, como así también permiten identificar aspectos posibles de mejora.

En [Curtis, y otros, 1992] se pueden encontrar algunos de los beneficios del modelado de procesos:

- ✓ Facilita el entendimiento y la comunicación, lo cual implica que el modelo de proceso sea entendible por el personal y que contenga suficiente información en su representación. Los modelos permiten ser usados como herramienta para formar al personal. Da formalidad al proceso, por lo tanto las personas pueden trabajar juntas de una manera más eficiente.
- ✓ Da soporte a la mejora de los procesos, ya que se identifican todos los componentes fundamentales del proceso. Permite reutilizar, en otros proyectos, los procesos bien definidos y efectivos. Ayuda a identificar posibles impactos ante potenciales cambios en el proceso sin la necesidad de ponerlos en práctica.

- ✓ Da soporte a la gestión del proceso (planificación, monitoreo y control), para lo cual se requiere un proceso definido contra el cual comparar. Proporciona una base para la medición del proceso, tal como la definición de los puntos de medición en el contexto de un proceso específico.
- ✓ Permite la automatización de los procesos, mejorando el rendimiento, y proporcionando a los usuarios: guías, instrucciones y material de referencia. Permite tener almacenadas las representaciones de los procesos reutilizables en un repositorio.

Otras ventajas de los modelos de proceso son:

- ✓ Debido a su naturaleza gráfica, son más expresivos que un documento textual. Esto permite, por ejemplo, observar fácilmente las distintas interrelaciones que existen entre las actividades, dependencias, puntos de control y paralelismos.
- ✓ El hecho de utilizar un lenguaje de modelado permite validar el modelo respecto a un conjunto de reglas conocidas, haciendo que el mantenimiento y la edición sean más fácil y menos propenso a errores.
- ✓ El uso de un lenguaje de modelado estándar permite el intercambio de modelos de procesos entre diferentes herramientas o, incluso, entre empresas.

Además, contar con modelos de proceso es un requisito para poder alcanzar los niveles 2 y 3 de CMMI. Igual ocurre con las certificaciones de ISO. Por lo tanto, modelar los procesos trae una ventaja especial si se desea alcanzar la certificación.

Uno de los objetivos primarios del modelado de proceso es lograr modelos precisos, es decir, modelos que pueden ser fácilmente mapeados a conceptos del mundo real [Becker, y otros, 1997]. Sin embargo, si bien es importante contar con modelos de procesos detallados, se debe tener cuidado de no abarrotar el diagrama con tanta información de tal manera que se vuelva difícil de comprender. El nivel de detalle en el modelo debería ser tal que la repetibilidad y la reproducibilidad de las actividades sean aseguradas y así lograr resultados consistentes. De igual manera, se deben fijar puntos de corte respecto al nivel de descomposición de las actividades. En el modelado de proceso, demasiada cantidad es sinónimo de baja calidad [Ministry of Forests, Lands and Natural Resource Operations, 1996].

2.2.2 Enfoques y Vistas de Modelado

Tal como se afirma en [McChesney, 1995], en la literatura se pueden encontrar varios enfoques de modelado de proceso que pueden ser utilizados con diferentes propósitos, como por ejemplo comprender, evaluar o automatizar. Dada esta diversidad de enfoques, en [McChesney, 1995] se propone un esquema de clasificación en base a los objetivos de los diferentes enfoques analizados:

- ✓ **Modelos Descriptivos:** tienen por objetivo hacer explícito un proceso que se está llevando a cabo. Los modelos descriptivos pueden ser usados como base para entender y mejorar un proceso. Permiten responder la pregunta "¿cómo se está realizando el proceso?" [Lonchamp, 1993]. A su vez, se pueden distinguir dos clases de modelos descriptivos:

- ❖ **Informales:** se utilizan principalmente como un medio para mejorar el entendimiento sobre un proceso, requisito previo para el logro de objetivos más ambiciosos, como la automatización.
- ❖ **Formales:** el objetivo es identificar las características del proceso con el fin de cuantificar, comparar, identificar problemas y hacer recomendaciones para la mejora, y para predecir situaciones futuras (cuellos de botella, *deadlocks*, tiempos, etc.).
- ✓ **Modelos Prescriptivos:** tienen por objetivo definir la manera recomendada para la ejecución de un proceso. Suelen ser utilizados como guías o marcos para organizar y estructurar las actividades de una organización. Permiten responder la pregunta "¿cómo se debería realizar el proceso?" [Lonchamp, 1993]. Un caso especial de este tipo de modelo son los modelos proscriptivos, los cuales están orientados a prescripciones negativas, es decir, prohibiciones. Los modelos prescriptivos pueden ser clasificados en:
 - ❖ **Manuales:** pueden ser estándares, como el proceso marco de ISO 12207 [ISO/IEC, 2008].
 - ❖ **Automatizados:** son especificaciones de estándares de procesos, relacionadas con la guía, soporte y ejecución asistida por computadoras. Los modelos prescriptivos automatizados, a su vez, se puede clasificar en: orientado a actividades u orientado a personas, en función de los aspectos en los que se centran.

También existen enfoques con diferentes niveles de abstracción, por ejemplo, los modelos genéricos contra los modelos específicos. Otro enfoque para clasificarlos es en base a qué se desea resaltar del proceso. Por ejemplo, en algunos casos se quiere dar énfasis a los roles que intervienen en la realización de las actividades, mientras que otras veces se desea dar mayor atención a las relaciones (orden, paralelismo) que existen entre las actividades. Por lo tanto se puede decir que los modelos permiten ver el proceso desde diferentes perspectivas y, así, focalizar la atención sobre determinados aspectos de un proceso. Como se comentó previamente, un modelo es una abstracción de la realidad, es decir, representa de una forma parcial y simplificada el mundo real; por ende no siempre todas las partes o aspectos de un proceso son tomadas en cuenta al momento de realizar un modelo. Sin embargo, si se desea tener un panorama completo, las *especificaciones de procesos* deberían incluir múltiples vistas o perspectivas y, así, cubrir las particularidades y complejidad de los procesos [Acuña, y otros, 2001b].

En [Curtis, y otros, 1992] se indica que un proceso puede ser modelado desde los siguientes puntos de vistas o perspectivas:

- ✓ **Funcional**, en el cual se describe qué actividades se deben llevar a cabo y qué flujo de entidades de información (documentos, productos, etc.) es importante para realizar las actividades.
- ✓ **De comportamiento**, que especifica cuándo y cómo deben ejecutarse las actividades, lo cual incluye identificar secuencias, paralelismos, sincronización, iteraciones, etc., y bajo qué condiciones son realizadas las actividades. Asimismo, puede especificar el ciclo de vida de un ente (por ejemplo de un artefacto, un proceso o un agente) con formalismos como un diagrama de transición de estados, *statecharts*, redes de Petri, etc.

- ✓ **Organizacional**, que tiene como fin mostrar dónde y quiénes son los agentes (en cumplimiento de roles) que intervienen en la realización de las actividades, como así también, qué estrategias de comunicación y dinámica de grupos se aplican.
- ✓ **Informacional**, que se centra en la estructura de los productos de trabajo producidos o requeridos por las actividades, en sus interrelaciones, y en las estrategias de administración de cambios de los productos de trabajo y seguimiento de los mismos.

Además de estas cuatro vistas, en [Olsina, 1997] se define una vista metodológica, en la cual se muestra particularmente qué constructores (por ejemplo, métodos específicos) centrados en modelos realizan las descripciones de las actividades.

Otros trabajos que definen varias vistas similares son [Derniane, y otros, 1999], donde se indica que el proceso puede ser analizado a través de los siguientes modelos: de actividades, organizacional, de productos, de recursos y de roles; y [Lonchamp, 1993], quien hace mención a los modelos de datos, de actividades, de recursos, de roles y organizacional. Otro trabajo relacionado es MVP-L (*Multi View Process-Language*) [Lott, y otros, 1990; Bröckers, y otros, 1995a], donde los autores modelan el proceso desde los diferentes puntos de vistas de cada rol. Cada modelo tiene en cuenta sólo las actividades que realiza un rol específico. Luego todas estas vistas (la de cada rol) son integradas en un modelo único y completo.

El presente trabajo hará uso de las vistas definidas en [Curtis, y otros, 1992] para modelar el proceso de M&E. Principalmente, se utilizará la perspectiva *funcional* para describir cada una de las actividades que deben llevarse, como así también la estructura jerárquica que existen entre ellas, y la de *comportamiento*, para mostrar cuál es el orden o secuencia en que se deben ejecutar las diferentes actividades. Con un menor grado de detalle, aunque no por ser menos importantes, se hará uso de la vista *informacional* con el fin de ejemplificar los productos de trabajo y la estructuración de los mismos, y la vista *organizacional* a los efectos de indicar los roles y así poder determinar las habilidades y capacidades que deberían tener los agentes para realizar las tareas de M&E.

2.2.3 Lenguajes de Modelado de Proceso

Existe una amplia variedad de lenguajes de modelado de procesos (*Process Modeling Languages* - PMLs). Estos surgieron a partir de mediados de la década de 1980 a causa del artículo "*Software Processes are Software Too*" [Osterweil, 1987], en el cual se exponían similitudes entre los procesos de software y el software mismo. Algunos de los primeros lenguajes fueron basados en reglas (por ejemplo: MARVEL), otros basados en redes (por ejemplo: SPADE) y otros basados en lenguajes de programación (como SPELL y APPL/A) [Bendraou, y otros, 2010]. Estos primeros lenguajes eran ejecutables y ponían énfasis en la formalidad. Sin embargo no fueron muy aceptados en la industria [Nitto, y otros, 2002] debido a que eran complejos, poseían formalismos de muy bajo nivel y eran inflexibles [Henderson-Sellers, y otros, 2005].

Con el tiempo surgieron otros lenguajes de modelado tales como: Redes de Petri, Diagramas de Actividad de UML, SPEM (*Software & Systems Process*

Engineering Meta-Model), XPDL (*XML Process Definition Language*), BPMN (*Business Process Modeling Notation*), IDEF0 (*Integration DEFINition method*), y Diagramas ETVX (*Entry criteria, Tasks, Validation and eXit criteria*), por citar algunos.

El uso de lenguajes de modelado permite:

- ✓ formalizar los procesos,
- ✓ poseer una especificación precisa de los procesos,
- ✓ utilizar herramientas de edición y mantenimiento de los modelos,
- ✓ automatizar los procesos, ya sea parcial o completamente,
- ✓ contar con descripciones estandarizadas o comunes en la organización, y
- ✓ un entendimiento común de los modelos.

Por otra parte, en [Finkelstein, y otros, 1994] se define un modelo de proceso como “la descripción de un proceso expresado en un lenguaje de modelado adecuado”. Esto significa que no todos los lenguajes son apropiados para representar o describir todos los aspectos de un proceso. Cada lenguaje nos permite observar, con mayor o menor detalle, el proceso desde algunas de las vistas ya analizadas. La razón es que cada lenguaje ha sido creado para especificar, capturar y comunicar cierto tipo de información, haciéndolos, por lo tanto, aptos para representar determinadas vistas y no para otras. En [Acuña, y otros, 2005] se realiza una asociación entre distintos lenguajes (base) y las perspectivas o puntos de vistas del proceso que estos cubren, según se muestra en la Tabla 2-1.

Tabla 2-1: Vistas de modelado de los procesos y lenguajes base aplicables.

Tipo de Lenguaje	Punto de vista
Lenguaje de Programación Procedural	Funcional De Comportamiento Informacional
Análisis y Diseño de Sistemas, incluyendo Diagramas de Flujo de Datos y técnicas de diseño estructurado	Funcional Organizacional Informacional
Lenguajes y aproximaciones de Inteligencia Artificial, incluyendo reglas y pre/post condiciones	Funcional De Comportamiento
Eventos y Disparadores / Control de Flujo	De Comportamiento
Diagramas de Transición de Estados y Redes de Petri, Diagramas de Estado	Funcional De Comportamiento Organizacional
Lenguajes Funcionales / Lenguajes Formales	Funcional
Modelado de Datos, incluyendo Diagramas Entidad/Relación, datos estructurados y declaraciones de relación	Informacional
Modelado de Objetos	Organizacional Informacional
Modelado Cuantitativo	De Comportamiento
Redes de Precedencia, incluyendo modelado de dependencias de actores	De Comportamiento Organizacional

No es materia de esta tesis el describir los diferentes lenguajes de modelado de proceso³, no obstante en el Apéndice A el lector puede encontrar una

³ Para una revisión de literatura relacionada a lenguajes de modelado de proceso ver [Zamli, 2001].

descripción de algunos lenguajes, a saber: IDEF0, Diagramas de Actividad de UML, BPMN y SPEM. Se han descrito dichos lenguajes por tener un gran uso dentro de la industria, la academia o por ser los que están apoyados por organismos que tienen un gran peso dentro del ámbito de la Ingeniería del Software o del modelado y definición de procesos.

2.2.4 Lenguaje de Modelado de Proceso Utilizado

La elección del lenguaje de modelado a utilizar depende de varios aspectos. Según [Pérez-Jiménez, y otros, 2009], se debe optar por un lenguaje que:

- ✓ tenga una notación formal que permita, entre otras cosas, validar los modelos obtenidos,
- ✓ sea expresivo, es decir, que el lenguaje permita expresar los diferentes aspectos o vistas que se quieren representar del proceso,
- ✓ posea una semántica bien definida, lo cual ayuda a su comprensibilidad,
- ✓ cuente con aplicaciones que nos permitan crear modelos con esa notación o lenguaje,
- ✓ tenga motores que nos permitan crear instancias de los procesos descritos para monitorear, simular o ejecutar dichos procesos.

La comparación de los diferentes lenguajes de modelado escapa al alcance de este trabajo. Algunas comparaciones pueden hallarse en [Portela, y otros, 2012a; Bendraou, y otros, 2010; Wei, y otros, 2006; Niknafs, y otros, 2009]. Por ejemplo, en [Fig, y otros, 2009] los autores realizan la comparación en base a la facilidad de uso de los lenguajes, ya que esta característica es importante para aprender un lenguaje, crear y entender los modelos. Los lenguajes comparados son: Diagramas de Actividad de UML, YAWL, BPMN y EPCs; la comparación se realizó en base a cinco criterios [Moody, y otros, 2008], a saber: claridad representacional (adecuación entre los símbolos gráficos y los conceptos semánticos), distinción perceptual (cuán fácil es para el usuario reconocer las diferencias –visual y conceptualmente- entre los símbolos), intermediación perceptual (facilidad para asociar símbolos con su correspondiente significado), expresividad visual (relacionado a las características visuales de los símbolos, tales como: forma, tamaño, color, orientación, etc.) y parsimonia gráfica (relación entre complejidad de los gráficos y sus expresividad).

Otra manera de comparar los lenguajes de modelado de proceso es según su nivel de expresividad. Este nivel se obtiene en base a la cantidad de patrones de *workflow* [van der Aalst, y otros, 2003] que soporta el lenguaje. Las comparaciones mediante patrones son muy utilizadas debido a que:


- ✓ están ampliamente difundidos,
- ✓ han sido aceptados por la comunidad de investigadores,
- ✓ son comprensibles por los profesionales de la informática, y
- ✓ presentan el nivel de abstracción adecuado para comparar las características de los lenguajes y notaciones de modelado de procesos.



Según estudios realizados [White, 2004; Russel, y otros, 2006; Eloranta, y otros, 2006] en base a la cantidad de patrones de *workflow* soportados, y teniendo en cuenta los lenguajes del Apéndice A, se puede decir que tanto BPMN como los



Diagramas de Actividad de UML 2.0 son muy similares; de hecho, hasta podrían fusionarse en el futuro [White, 2004]. Por otro lado, SPEM 2.0 es más robusto, ya que permite utilizar tanto Diagramas de Proceso de Negocio de BPMN como Diagramas de Actividad de UML, junto a otros diagramas (como por ejemplo los Diagramas de Clase), permitiendo así, especificar varias perspectivas o vistas. En cuanto a IDEF0 se puede decir que es un lenguaje de menor expresividad que los ya mencionados, debido a su limitada simbología (cajas y flechas).





En este trabajo de tesis se hará uso de SPEM junto con Diagramas de Actividad y de Clase de UML 2.0. Estos lenguajes son altamente aceptados, expresivos, tienen una semántica y una sintaxis bien definidas, son fáciles de comprender y lo suficientemente amplios para modelar el proceso de M&E desde las diferentes vistas, a saber: funcional, de comportamiento, informacional y organizacional. Además, SPEM es considerado como el estándar *de facto* en la industria para la representación de modelos de procesos de Ingeniería del Software e Ingeniería de Sistemas. Otro punto a favor para las notaciones elegidas es que existen herramientas gratuitas que las soportan, como por ejemplo, StarUML [StarUML] y EPFC (*Eclipse Process Framework Composer*) [Eclipse]. Este último editor permite definir, gestionar y reutilizar un repositorio de fragmentos de procesos. Por otro lado, cabe mencionar que SPEM tiene la desventaja de que sus modelos no son ejecutables. Sin embargo, para este trabajo, esta es una característica menos relevante ya que no es el objetivo de esta tesis desarrollar modelos interpretables por una máquina, sino sólo representar las vistas del proceso. Cabe acotar, no obstante, que existen trabajos [Feng, y otros, 2006; Ellner, y otros, 2011; Bendraou, y otros, 2007b; Portela, y otros, 2012b; Bendraou, y otros, 2007a; Riesco, y otros, 2009] que dan solución a esta cuestión.

A continuación se listan y definen los conceptos fundamentales de SPEM 2.0 que se utilizan en los modelos presentados en los Capítulos 4 y 5:

 **Activity**: representa una unidad general de trabajo –actividad– dentro de un proceso. Una actividad involucra a un ejecutor específico representado por un rol (Rol Use). A su vez, una actividad insume entradas y produce salidas representadas por Work Product Use.

 **Task Definition** y  **Task Use**⁴: representan una unidad asignable de trabajo –tarea–, que se asocia a un rol (Rol Definition/Use). La granularidad de una tarea es generalmente de unas pocas horas a unos pocos días. Usualmente, afecta a uno o a un pequeño número de productos de trabajo (Work Product Definition/Use).

 **Role Definition** y  **Role Use**: representan a un ejecutor o participante de una tarea (Task Definition/Use) o actividad (Activity), en cumplimiento de un rol. Cabe aclarar que un rol no identifica a una persona en particular y que los miembros de una organización pueden cumplir diferentes roles.

 **Work Product Definition** y  **Work Product Use**: representan un tipo de entrada/salida de una actividad (Activity) o tarea (Task Definition/Use). Tipos particulares son:  Outcome (resultado) y  Artifact (artefacto).

⁴ SPEM utiliza el sufijo “Definition” (como por ejemplo en: Task Definition y Work Product Definition) para representar elementos reutilizables que son definidos de forma genérica, y el sufijo “Use” (como por ejemplo en: Task Use y Work Product Use) para representar elementos específicos.



Phase: representa un periodo significativo en un proyecto, el cual termina con un hito, un conjunto de entregables o un punto de control importante para la gestión.



Tool Definition: representa una herramienta CASE, una herramienta de propósito general, o cualquier otra unidad de automatización.

Cabe notar que, además de los conceptos fundamentales, también se han presentado los íconos que representa cada concepto de SPEM. Estos íconos son los elementos visuales utilizados en la construcción de diagramas, tal como veremos en los capítulos citados.

Capítulo 3: Estado del Arte de Procesos en Estrategias Integradas de M&E

En este capítulo, en primer lugar, se analizarán distintos enfoques de M&E con el objetivo de identificar aquellos que cuentan con las tres capacidades descritas en la Sección 2.1.2. Luego, teniendo presente que el foco de esta tesis es el modelado del proceso de M&E, se analizará si los enfoques revisados especifican tal proceso desde las diferentes vistas comentadas en la Sección 2.2.2, a saber: funcional, organizacional, de comportamiento e informacional.

En segundo lugar, se exponen algunos trabajos que definen una base conceptual para el dominio de proceso. Tal revisión permite ver la falta de consenso que existe, aún al día de hoy, en la terminología empleada en este dominio y, por lo tanto, la necesidad de definir una base conceptual de proceso que permita fijar el significado que se dará en este trabajo a los distintos términos referidos al área de proceso.

Finalmente, se mencionan los aportes que se proponen en este trabajo en función de las debilidades y oportunidades de mejoras encontradas en el estado del arte.

3.1 Revisión de Enfoques y Estrategias de M&E

Con el objetivo de diseñar e implementar programas y proyectos de M&E que sean robustos, consistentes, repetibles y reproducibles, como se comentó en el capítulo anterior, es necesario contar con enfoques o estrategias integradas de M&E, es decir que posean y relacionen las siguientes tres capacidades: *i) un marco conceptual* en el cual se definan los principales términos para el dominio de M&E, *ii) una especificación de las vistas del proceso* en la cual se describan las principales actividades que deben realizarse, junto con sus entradas, salidas, roles y otros aspectos, y *iii) métodos y herramientas* específicas que permitan llevar a cabo las actividades del proceso.

Si bien en la literatura existen muchas estrategias o enfoques de M&E, se puede decir que no todas ellas poseen las tres capacidades citadas. Y, aquellas estrategias que cuentan con estas capacidades, no siempre se puede decir que son integradas, ya que a veces las capacidades no están relacionadas entre sí. Teniendo esto presente, a continuación se comentarán (sin entrar en detalles) diferentes trabajos que aportan enfoques de M&E y se examinará si cuentan con las tres capacidades y además si pueden ser consideradas integradas o no.

3.1.1 ISO - International Organization for Standardization

ISO ha definido los estándares 15939 [ISO/IEC, 2002a; ISO/IEC, 2007b] y 14598

[ISO/IEC, 2001b] para la medición y evaluación de software. El estándar 15939, denominado “*Systems and software engineering – Measurement process*”, define un proceso de medición aplicable en la Ingeniería de Software y las disciplinas de gestión. Este proceso de medición se describe mediante las actividades y tareas necesarias para definir, aplicar y mejorar la medición dentro de un proyecto en particular o dentro de la estructura de medición organizacional. Las actividades y tareas definidas permiten especificar: qué información de medición es requerida, cómo las medidas y los resultados del análisis son aplicados, y cómo determinar si el análisis de los resultados es válido.

Además, el estándar ISO 15939 también provee definiciones de los términos de medición usados comúnmente dentro de la industria del software. Principalmente, se enfoca en los conceptos del proceso de medición, aunque también cubre algunos otros referidos a medidas y objetivos. Básicamente, toma los conceptos de la norma ISO 14598 e ISO 9126 y redefine algunos términos con el fin de ajustarse al vocabulario de metrología publicado en [ISO/IEC, 2007a]. Respecto a métodos y herramientas, en el estándar ISO 15939 se especifican un conjunto de métricas que pueden ser aplicadas al proceso de medición.

Por otro lado, el estándar ISO 14598, denominado “*Information technology - Software product evaluation*”, consta de seis partes que proporcionan un marco de trabajo para evaluar la calidad de todos los tipos de productos de software e indica los requisitos para los métodos de medición y evaluación del proceso. La primer parte de este estándar, denominada “*Information technology - Software product evaluation - Part 1: General overview*”, provee una visión general de las otras cinco partes y explica la relación entre este estándar y el modelo de calidad definido en ISO 9126. Adicionalmente, puntualiza los términos técnicos usados, y clarifica los conceptos generales.

La quinta parte de la norma, denominada “*Information technology - Software product evaluation - Part 5: Process for evaluators*”, provee requisitos y recomendaciones para la aplicación práctica de la evaluación de productos software cuando existen varias partes que necesitan entender, aceptar y confiar en los resultados de la evaluación. Esta parte prescribe un proceso de evaluación que se divide en cinco actividades y puede ser usado para evaluar productos existentes, componentes que necesitan ser provistos o productos en desarrollo. En caso de que la evaluación sea de un producto en desarrollo, el proceso de evaluación necesita ser sincronizado con el proceso de desarrollo de software y los componentes del producto son evaluados como entregables.

Es importante remarcar que el estándar ISO 14598 está muy ligado al estándar ISO 9126 y, más allá de que ambos comparten la misma terminología, existen inconsistencias a raíz de la utilización de ciclos de vida separados para la especificación de requisitos y para la evaluación del producto. Con el fin de eliminar las distancias, conflictos y ambigüedades entre estos estándares, y así lograr una convergencia entre ambos, surge el proyecto SQuaRE (*Software Product Quality Requirements and Evaluation*). El resultado de este proyecto es una serie de estándares (ver Figura 3-1):

- ✓ ISO/IEC 2501n (*Quality Model Division*): Presenta modelos de calidad para software, calidad en uso y para datos.
- ✓ ISO/IEC 2502n (*Quality Measurement Division*): Exhibe un modelo de referencia

de medición de la calidad de productos software, junto con métricas y guías prácticas para su aplicación.

- ✓ ISO/IEC 2503n (*Quality Requirements Division*): Brinda ayuda para especificar los requerimientos de calidad. Estos requerimientos pueden ser usados en el proceso de obtención de los requerimientos de calidad del producto software a ser desarrollado o como entrada a un proceso de evaluación.
- ✓ ISO/IEC 2504n (*Quality Evaluation Division*): Provee requerimientos, recomendaciones y guías para la evaluación de productos software.

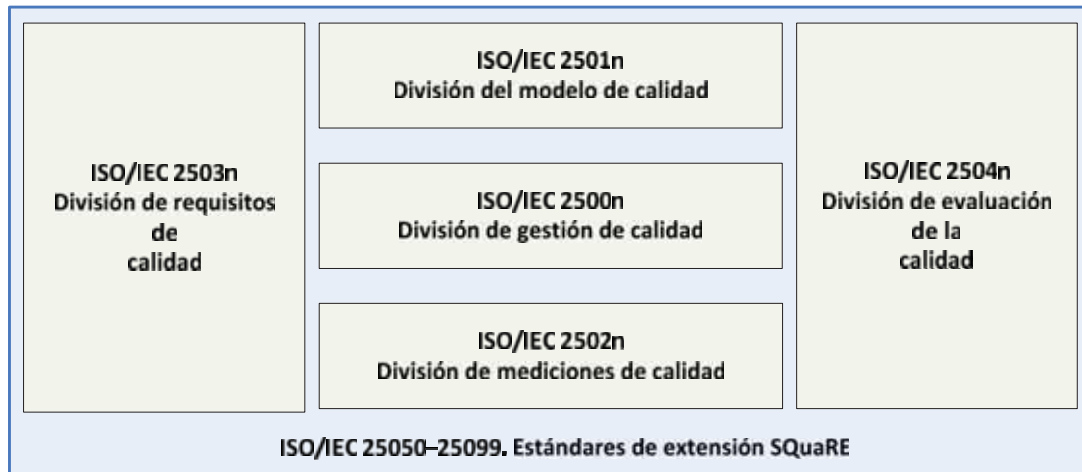


Figura 3-1: Estándares del proyecto SQuaRE.

Si bien SQuaRE ha visto la luz al momento de escritura de esta tesis, aun no han sido publicados todos estos estándares. Por lo tanto siguen siendo válidos los anteriores (14598 y 15939), los cuales no permiten establecer una estrategia de M&E integrada. Una de las razones es la falta de consenso en los términos utilizados, por ejemplo, en ISO 14598-5 [ISO/IEC, 2001b] y en ISO 9126-1 [ISO/IEC, 2001a] se usan los términos “*métrica*”, “*medida directa*” y “*medida indirecta*”, mientras que en ISO 15939 [ISO/IEC, 2002a] se usan los términos “*medida base*” y “*medida derivada*”, no siendo meros sinónimos (para una discusión más detallada ver [Olsina, y otros, 2004]). Además, se presentan dos procesos, uno para medición y otro para evaluación, los cuales no están integrados, ya que hay actividades que se solapan entre ambos (ver Sección 3.2.1). Cabe destacar que la nueva familia de estándares ISO 25000 tampoco permiten definir una estrategia integrada de M&E. Por ejemplo, no existe un estándar en el cual se defina un vocabulario, taxonomía, ontología, etc., que permita aclarar y brindar uniformidad respecto de los términos utilizados.

3.1.2 CMMI - Capability Maturity Model Integration

El estándar *de facto* CMMI [CMMI Product Team, 2010] define 22 áreas de proceso. Un área de proceso es un grupo de prácticas (como actividades) relacionadas en un área que, cuando se implementan de forma conjunta, satisfacen un grupo de metas consideradas importantes para la mejora en ese área. Particularmente nos interesa el área de proceso *Medición y Análisis*. Este área proporciona prácticas que guían a los proyectos y a las organizaciones durante la alineación de las necesidades y objetivos de medición con una forma de medir que proporcionará resultados verificables. A su vez, estos resultados pueden usarse para la toma de decisiones

informadas y de acciones correctivas apropiadas. El enfoque de medición que utiliza CMMI es orientado a metas, y se basa en las ideas de GQM (*Goal Question Metric*) [Basili, y otros, 1988], y en el proceso de medición definido en el estándar ISO 15939. Sin embargo, CMMI no define explícitamente ningún modelo de proceso. Más bien se limita a fijar metas que deben alcanzarse para cada área de proceso, y prácticas para alcanzar tales metas, pero sin establecer ningún proceso específico. Para Medición y Análisis las metas específicas son: 1) *Alinear las actividades de Medición y Análisis* y 2) *Proveer Resultados de la Medición*, mientras que las prácticas específicas para alcanzarlas son: 1.1) *Establecer los objetivos de la medición*, 1.2) *Especificar medidas*, 1.3) *Especificar procedimientos de recolección y almacenamiento de datos*, y 1.4) *Especificar procedimientos de análisis* -para la primera meta-, y 2.1) *Obtener datos de la medición*, 2.2) *Analizar datos de la medición*, 2.3) *Almacenar datos y resultados*, y 2.4) *Comunicar los resultados* -para la segunda meta.

Cabe destacar que para el área de proceso Medición y Análisis no se indica ningún método a seguir ni herramientas a utilizar para la realización de las diferentes prácticas. Aunque en muchos casos se proporcionan ejemplos orientadores de posibles métodos y herramientas. Por ejemplo, para la práctica específica *Almacenar datos y resultados* se aconseja que se debe prevenir que la información almacenada no sea utilizada inapropiadamente, y a continuación indica que “*ejemplos de formas de prevenir el uso inapropiado de los datos es incluyendo controles de acceso a los datos y educando al personal sobre el uso apropiado de los datos*”. Sin embargo estos ejemplos no siempre sirven de guía para mostrar cómo debe llevarse a cabo la práctica.

Finalmente, podemos indicar que CMMI cuenta con un glosario como base conceptual en el cual se definen los principales términos utilizados en el estándar. Como puede apreciarse, CMMI carece de una especificación de procesos, de métodos y herramientas para llevar a cabo las prácticas/actividades y de una base conceptual robusta. Sin embargo, a pesar de esto, este estándar es ampliamente utilizado y reconocido en la academia y la industria.

3.1.3 PSM - Practical Software Measurement

Practical Software Measurement (PSM) [McGarry, y otros, 2001] se inicia en 1994 en el Departamento de Defensa de los Estados Unidos. Sus fundamentos surgen de la experiencia de sus autores en diferentes proyectos. PSM sirvió como documento base para el desarrollo del estándar ISO 15939. Y con la llegada de dicho estándar, PSM fue adaptado para proveer detalles adicionales sobre las actividades y tareas presentadas en el estándar. Además, provee una guía detallada -es decir, métodos- para llevar a cabo las tareas. Por lo tanto, PSM contribuye a asegurar una efectiva implementación del estándar ISO 15939 y también del área de proceso Medición y Análisis de CMMI.

PSM utiliza un *Modelo de Información de Medición* y un *Modelo de Proceso de Medición*. El *Modelo de Información de Medición* provee un mecanismo formal para enlazar las necesidades de información definidas a los procesos de Ingeniería de Software y productos que pueden ser medidos. El modelo sirve como un recurso primario para el proceso de medición y como una guía para las actividades de planificación e implementación de la recopilación y análisis de los datos. Además, el *Modelo de Información de Medición* establece una estructura para relacionar los

diferentes conceptos de medición. Por lo tanto, establece una base para la definición consistente de la terminología de medición. Los términos definidos en la base conceptual de PSM se derivan del *Modelo de Información de Medición* documentado en el estándar ISO 15939.

Por otro lado, el *Modelo de Proceso de Medición* está basado en la típica secuencia de gestión “*Plan-Do-Check-Act*”, pero adaptado para soportar actividades y tareas específicas de medición. El *Modelo de Proceso de Medición* incluye cuatro actividades primarias, a saber: *Planificar la Medición* (Plan Measurement), *Ejecutar la Medición* (Perform Measurement), *Evaluar la Medición* (Evaluate Measurement), y *Establecer y Mantener el Compromiso* (Establish and Sustain Commitment).

En la documentación de PSM se hace referencia a herramientas analíticas (tales como histogramas, análisis de regresión, y diagramas de causa-efecto, entre otras), así como a métodos matemáticos y estadísticos (como por ejemplo 6 sigma y distribución normal) que pueden ser aplicados a diferentes tareas. Además, se presentan plantillas que ayudan a estructurar los objetivos de la medición, y se muestran procedimientos (mediante diagramas de flujo) para ciertas actividades. Incluso, existe la herramienta *PSM Insight* (PSMI), la cual es un software de medición gratuito que implementa completamente el proceso PSM.

Teniendo en cuenta lo comentado, PSM cubre los tres pilares fundamentales de una estrategia. Incluso, cabe resaltar que el *Modelo de Proceso de Medición* trabaja en conjunción con el *Modelo de Información de Medición*, proveyendo un marco para definir programas de medición que sean apropiados para cada proyecto de la organización. Sin embargo, es importante destacar que PSM, al igual que CMMI, no incluye actividades del proceso de Evaluación. Y, como se indicó al finalizar la Sección 2.1.1, una medición sin evaluación no es más que sólo un conjunto de valores o datos con mínima información interpretada, la cual, no siempre es útil a la hora de realizar análisis y tomar decisiones.

3.1.4 Alarcos Quality Center

Alarcos Quality Center (AQC) es una empresa surgida de la Universidad de Castilla-La Mancha cuyo objetivo es proporcionar un servicio de aseguramiento de la calidad del software tanto a empresas y departamentos de desarrollo, como a empresas que compran software. Uno de sus aportes respecto a enfoques de M&E es FMESP (*Framework for the Modeling and Evaluation of Software Processes*) [García, y otros, 2006b]. Los principales componentes de FMESP son su marco conceptual y un entorno de Ingeniería de Software (*Software Engineering Environment, SEE*).

El marco conceptual de FMESP puede observarse en la Figura 3-2. Este consta de tres componentes:

- I. *Arquitectura Conceptual*: permite proveer la gestión integrada del modelado y la medición mediante la representación de los elementos relacionados en las diferentes capas de abstracción. Esta integración se realiza mediante la inclusión de lenguajes de modelado necesarios para definir el proceso, los modelos de medición de las entidades del proceso y los metamodelos del dominio para representar cualquier entidad relacionada al proceso de software.

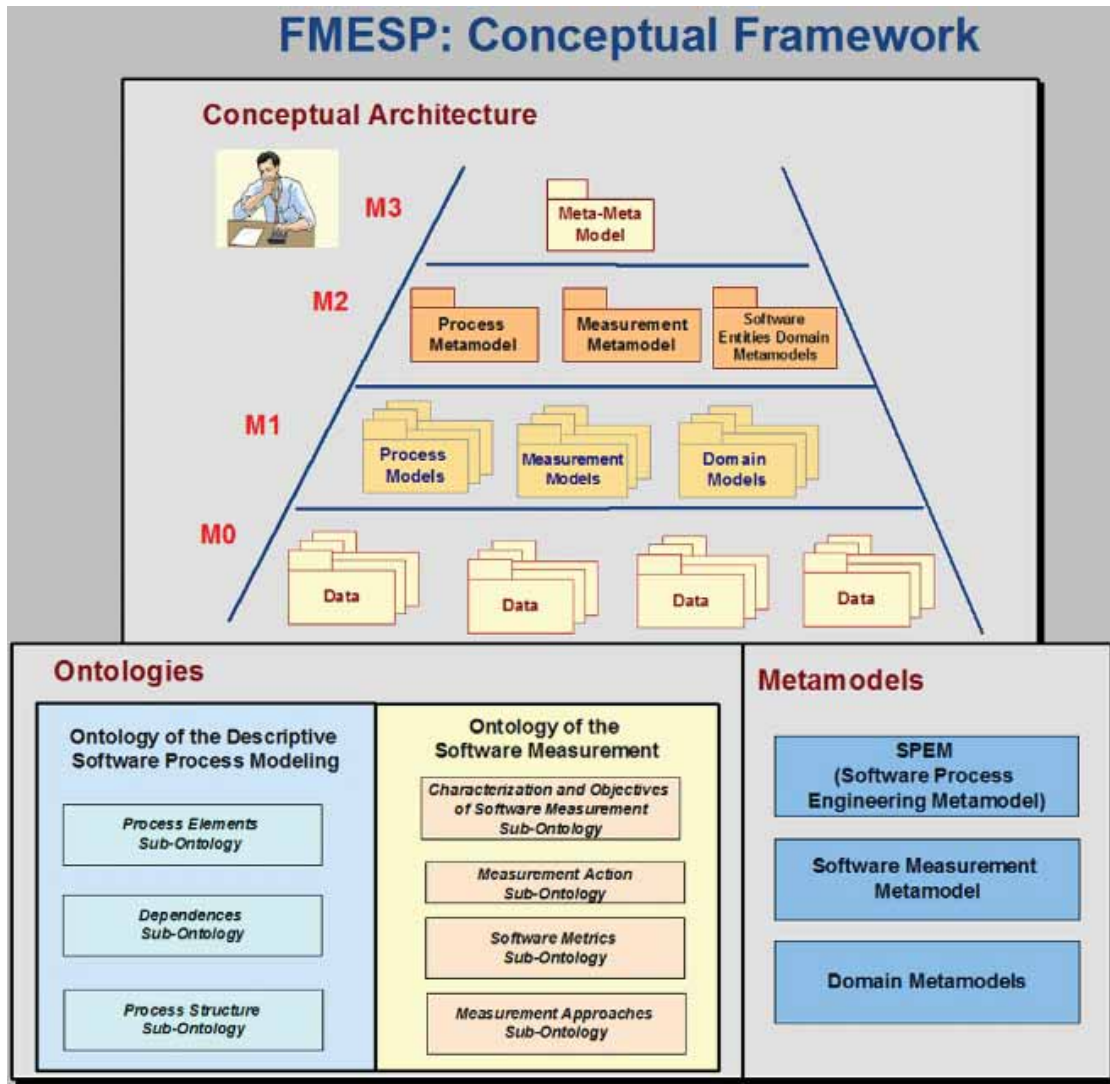


Figura 3-2: Marco conceptual de FMESP.

- II. *Ontologías*: un aspecto fundamental a considerar para lograr integrar de manera correcta el modelado y la medición del proceso de software es que todos los modelos y metamodelos deben basarse en la misma conceptualización y tal conceptualización debe estar especificada mediante ontologías. Por este motivo cuenta con las siguientes ontologías:
- ✓ Ontología de modelado descriptivo de procesos de software: si bien los autores indican que han tomado como referencia la especificación de SPEM para realizar su ontología, la misma aun no ha sido publicada o no es de público conocimiento.
 - ✓ Ontología de medición de software (SMO - *Software Measurement Ontology*): Esta ontología fue presentada en [García, y otros, 2004] y luego refinada en [García, y otros, 2006a]. Tiene ciertas similitudes con la ontología del marco conceptual C-INCAMI (comentado en la Sección 4.2), sin embargo en [Olsina, y otros, 2004] se han modelado algunos otros conceptos (como indicador elemental e indicador global) y relaciones (por ejemplo, entre medición y medida, entre métrica e indicador, etc.) que difieren semánticamente de las propuestas en SMO; además en C-INCAMI

se han incluido nuevos conceptos como contexto y propiedad de contexto, no consideradas en SMO.

III. Metamodelos: cuenta con *metamodelos de dominio* como SPEM, de medición de software, y otros, para la definición de entidades relacionadas a procesos de software.

FMESP cuenta con dos herramientas integradas que proveen el soporte tecnológico al marco conceptual. Una es METAMOD, para la definición y gestión de los metadatos de la arquitectura conceptual, y la otra es GenMETRIC, que permite definir modelos de medición y calcular las medidas definidas en dichos modelos sobre cualquier entidad de software. Además se pueden agregar nuevas herramientas que den soporte a la evaluación y mejora de los procesos. Los autores de FMESP, también proporcionan un conjunto de métricas útiles para medir procesos y artefactos de software, proveyendo de esta manera métodos y herramientas que den soporte a las actividades de medición.

En lo que respecta a la definición de un proceso de M&E en el cual se describan de manera explícita las actividades a llevarse a cabo, los autores no cuentan con una especificación pública y accesible, a pesar de que expresan la importancia de contar con modelos de proceso y evaluarlos. Por lo tanto se puede decir que este enfoque de M&E solo cubre dos de las tres capacidades fundamentales de una estrategia de M&E.

Cabe comentar, que el grupo Alarcos, además de desarrollar FMESP, también ha desarrollado CQA-Meth (*Continuos Quality Assessment Methodology*) [Rodríguez, y otros, 2010a], la cual, también es llamada *Metodología EVVE* según otra publicación [Rodríguez, y otros, 2010b]. La sigla EVVE proviene de "*Entorno para la Verificación y Validación de Especificaciones software*". Esta metodología tiene por objetivo definir un marco de trabajo que permita determinar los procesos necesarios para llevar a cabo la evaluación de modelos UML, así como facilitar la comunicación entre el cliente (patrocinador de la evaluación) y el equipo de evaluación. CQA-Meth, junto con la herramienta que la soporta (CQA-Tool), forman el entorno integrado CQA que puede ser utilizado por empresas dedicadas a ofrecer servicios de evaluación de la calidad para organizaciones de software, así como a los clientes que hayan externalizado la construcción de sus productos software, permitiendo, de este modo, la obtención de una evaluación de la calidad independiente a la que realice el fabricante. Las empresas de desarrollo de software también pueden utilizar la metodología para llevar a cabo evaluaciones de sus propios productos software.

CQA-Meth (o EVVE) está formada por tres procesos, a saber: *Proceso de Gestión de la Evaluación, Proceso de Evaluación y Proceso de Gestión de la Infraestructura*. A su vez, el Proceso de Evaluación se divide en 4 fases: *Planificación, Especificación, Ejecución y Conclusión*. Cada una de estas fases agrupa un conjunto de actividades de las cuales no se indica más que el nombre. Además, a pesar de que listan y definen los productos de trabajo (llamados "elementos" por los autores) no los asocian a ninguna actividad/tarea en particular. También vale decir que, aunque CQA-Meth surge en el mismo grupo de investigación que FMESP, no comparte su base conceptual de M&E, o al menos no lo indica de manera explícita.

3.1.5 GQM+Strategies - Goal Question Metric Plus Strategies

Uno de los enfoques más usados, a nivel académico e industrial, es *Goal Question Metric* (GQM) [Basili, y otros, 1988; Basili, y otros, 1994]. Este enfoque guía a la organización en la definición de objetivos, la especificación de datos a recolectar y su posterior análisis e interpretación. Usa implícitamente modelos que ayudan a los profesionales a interpretar los resultados de los datos según los objetivos definidos y en el contexto específico. Un beneficio de GQM es que asegura que el conjunto de métricas es lo más chico posible y que la recolección de datos es útil y direcciona a los objetivos de software definidos.

Existen muchas líneas de investigación relacionadas a GQM ([Gresse, y otros, 1995], [Solingen, y otros, 1997], [Briand, y otros, 2002], [Goethert, y otros, 2003]) que extienden y mejoran el enfoque propuesto por sus autores originales definiendo el proceso, el modelo de información o complementándolo con otros enfoques existentes. Particularmente se puede citar el enfoque GQ(I)M, *Goal Question (Indicator) Measurement* [Goethert, y otros, 2001; Goethert, y otros, 2004]. Este se distingue de GQM en que agrega un paso intermedio para asistir en el enlace entre las preguntas y los datos de medición que serán recolectados. La importancia de explicitar esta relación entre las preguntas y los datos que las responden fue el éxito del enfoque GQM, pero en GQ(I)M se sostiene que la identificación de preguntas y medidas sin la visualización de un indicador, frecuentemente, no es suficiente para obtener el éxito del programa de medición. Los “*displays*” o reportes usados para comunicar los datos (llamados indicadores en este enfoque) son un enlace clave para el éxito o falla del programa de medición. Los indicadores sirven como especificación de requerimientos para los datos que deben ser recolectados, procesados y analizados. Los autores presentan plantillas para la definición de los indicadores.

En los últimos años ha surgido GQM+Strategies [Basili, y otros, 2007b; Basili, y otros, 2010], un enfoque de medición basado en GQM, que le agrega capacidad de crear programas de medición que aseguren la alineación entre los objetivos de negocio, los objetivos específicos de software y los objetivos de medición. GQM+Strategies, crea un enlace entre los datos recolectados y los relaciona con los objetivos a diferentes niveles. El entendimiento ganado a un nivel sirve para el siguiente nivel –por ejemplo: un objetivo de proyecto, departamental o de negocio puede soportar o contribuir a satisfacer objetivos a los otros niveles sin requerir que cada nivel comparta los mismos objetivos.

Según [Basili, y otros, 2007a] y [Basili, y otros, 2010], GQM+Strategies posee una base terminológica formada por un conjunto de 18 términos agrupados en un glosario. En el resto de las publicaciones, las definiciones se van presentando a medida que se desarrolla el trabajo, en algunas ocasiones en la sección donde se presenta la estrategia. Dichos términos están divididos en dos módulos o agrupación de conceptos, que tienen que ver con dos niveles de abstracción:

- ✓ Elementos de GQM+Strategies: que involucra los conceptos pertenecientes a GQM+Strategies; y
- ✓ Grafo GQM: que involucra los conceptos pertenecientes al enfoque original de GQM (en [Briand, y otros, 1996b] y [Differding, y otros, 1996] se pueden hallar los conceptos y definiciones originales).

Con respecto al proceso de GQM+Strategies, si no se tienen en cuenta las actividades de enlace entre objetivos de diferentes niveles, sigue siendo el mismo que en GQM. Las principales actividades son: *Identificar los objetivos GQM, Producir el plan de medición, Recolectar y validar los datos y Analizar los datos*. En [Gresse, y otros, 1995] también se pueden identificar entradas, salidas y roles, e incluso en [Solingen, y otros, 1999] se presentan plantillas de los productos de trabajo.

Las diferentes actividades del proceso de GQM están soportadas por diferentes métodos, como entrevistas, cuestionarios y tormenta de ideas. Con respecto a las herramientas utilizadas durante la aplicación de GQM, existen herramientas de recolección de datos automáticas, las cuales calculan métricas específicas de acuerdo a un algoritmo predefinido. En [Solingen, y otros, 1999] se describe el desarrollo de un *Sistema de Soporte a la Medición* (MSS, por sus siglas en inglés) que sea configurable para cada programa de medición. Este sistema puede estar basado en herramientas genéricas, tales como hojas de cálculo, software estadístico, aplicaciones de base de datos o herramientas de presentación. Un MSS completo debería dar soporte a todas las actividades de medición, por lo que debería proporcionar facilidades para recolectar, almacenar, mantener, procesar, presentar y empaquetar los datos de medición. En [Solingen, y otros, 1999] se encuentra una descripción detallada de la funcionalidad requerida y una posible arquitectura.

Teniendo en cuenta lo comentado, se puede concluir que GQM puede considerarse una estrategia integrada de M&E ya que posee una base conceptual, un proceso definido explícitamente y métodos y herramientas para llevar a cabo las actividades del proceso. Aún más, en [Olsina, y otros, 2011] y en [Papa, 2012] se presenta un estudio comparativo de estrategias integradas de M&E, una de ellas GQM+Strategies. Dicho estudio permite observar el nivel de satisfacción alcanzado por cada una de las estrategias analizadas teniendo en cuenta las tres capacidades deseables en una estrategia.

3.2 Especificación de las Vistas del Proceso de M&E

Tal como se acaba de analizar, no todos los enfoques de M&E pueden ser considerados *Estrategias Integradas de M&E*. Sin embargo, el foco de este trabajo no es identificar estrategias integradas de M&E, sino más bien el modelado de las vistas de proceso de estrategias de M&E. Por lo tanto nos interesa profundizar principalmente en la capacidad *Especificación de las vistas del proceso* de los enfoques de M&E ya examinados. Consecuentemente, a continuación se analizará la manera, es decir, las vistas o perspectivas que se utilizan para especificar/modelar los procesos de M&E de las diferentes estrategias/enfoques vistos. Recordar que la vista *funcional* permite identificar qué actividades se deben llevar a cabo y qué entradas y salidas son importantes para realizar las actividades; la vista *de comportamiento* ayuda a conocer cuándo y cómo deben ejecutarse las actividades (es decir, muestra secuencias, paralelismos, iteraciones, condiciones, etc.); la vista *organizacional* tiene como fin mostrar los roles que intervienen en las actividades; y la vista *informativa*, se centra en la estructura de los productos de trabajo, y en sus interrelaciones.

3.2.1 ISO - International Organization for Standardization

En el estándar ISO 15939 se indica que son cuatro las actividades que guían el

proceso de medición de software, a saber: *Establecer y mantener el compromiso de medición, Planear el proceso de medición, Ejecutar el proceso de medición y Evaluar la medición*. Estas actividades son llevadas a cabo en un ciclo iterativo permitiendo una retroalimentación y mejora continua del proceso de medición [ISO/IEC, 2002a]. En la Figura 3-3 se pueden observar, desde una perspectiva funcional, las cuatro actividades que componen el proceso de medición, además de sus respectivas entradas y salidas. El estándar menciona que las actividades *Planear el proceso de medición* y *Ejecutar el proceso de medición* son las dos actividades centrales o principales del proceso de medición. Las otras dos actividades proveen una base para las actividades principales del proceso de medición y generan información de retroalimentación para realizar mejoras.

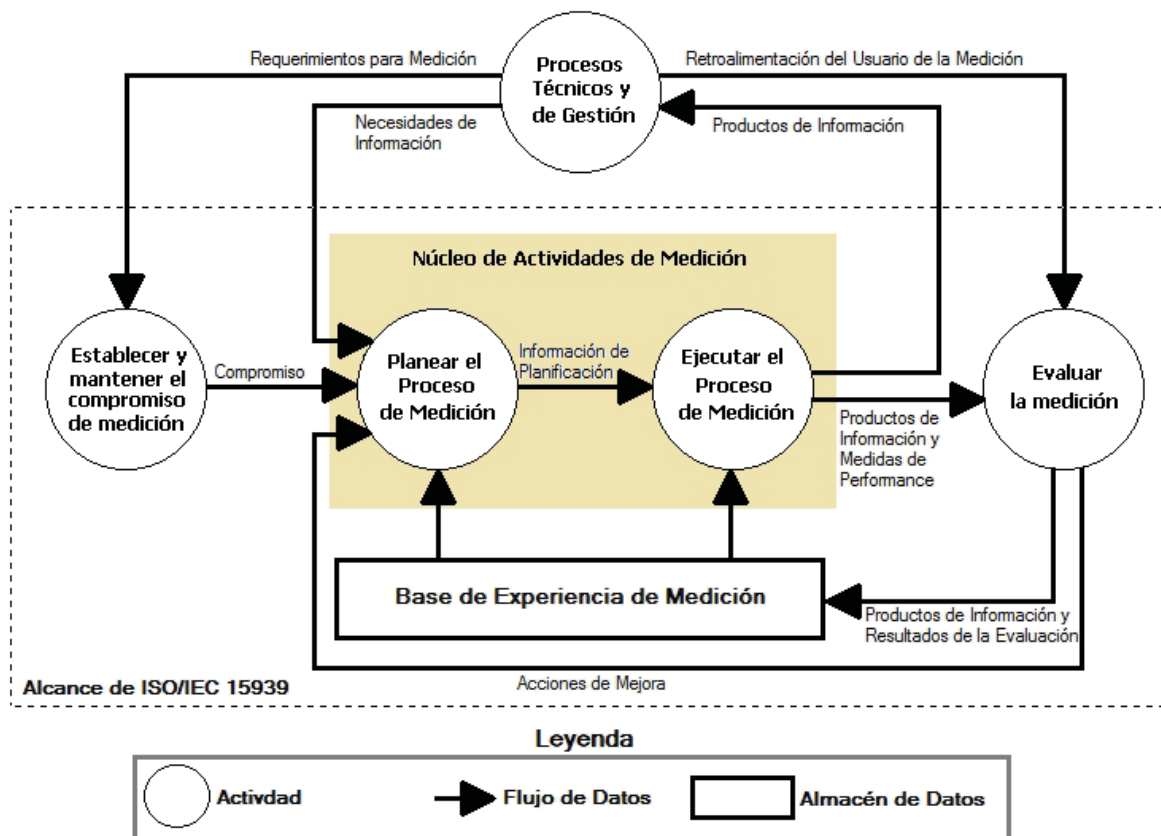


Figura 3-3: Modelo de Proceso de Medición de Software (adaptado de [ISO/IEC, 2002a]).

A su vez, las actividades mencionadas se dividen en tareas. Sin embargo, estas no se modelan desde ningún punto de vista. A continuación, se muestra el desglose del proceso de medición del estándar ISO 15939, donde en el nivel superior se encuentra el proceso, en el intermedio las actividades y en el tercer nivel las tareas:

1. Proceso de medición de software
 - 1.1. Establecer y mantener el compromiso de medición
 - 1.1.1. Aceptar los requerimientos para la medición
 - 1.1.2. Asignar recursos
 - 1.2. Planear el proceso de medición
 - 1.2.1. Caracterizar la unidad organizacional
 - 1.2.2. Identificar la necesidad de información
 - 1.2.3. Seleccionar las medidas

- 1.2.4. Definir los procedimientos de recolección de datos, análisis, e informe
- 1.2.5. Definir criterios para evaluar los productos y los procesos de medición
- 1.2.6. Revisar, aprobar, y proveer recursos para las tareas de medición
- 1.2.7. Implantar tecnologías de soporte
- 1.3. Ejecutar el proceso de medición
 - 1.3.1. Integrar procedimientos
 - 1.3.2. Recolectar datos
 - 1.3.3. Analizar datos y desarrollar productos de información
 - 1.3.4. Comunicar resultados
- 1.4. Evaluar la medición
 - 1.4.1. Evaluar los productos de información y el proceso de medición
 - 1.4.2. Identificar potenciales mejoras

Por otra parte, el proceso de evaluación descrito en el estándar ISO 14598-5 [ISO/IEC, 2001b] se basa en cinco actividades principales, a saber: *Establecimiento de los requerimientos de evaluación, Especificación de la evaluación, Diseño de la evaluación, Ejecución de la evaluación y Conclusión de la evaluación*. En la Figura 3-4 se modela el proceso de evaluación desde una perspectiva funcional, mostrando claramente las actividades involucradas y sus respectivas entradas y salidas. A su vez, cada una de estas cinco actividades se divide en sub-actividades, como se puede ver en el siguiente desglose del proceso de evaluación:

- 1. Proceso de Evaluación
 - 1.1. Establecimiento de los requerimientos de evaluación
 - 1.1.1. Elaboración de los requerimientos de evaluación
 - 1.1.2. Aprobación e informe
 - 1.2. Especificación de la evaluación
 - 1.2.1. Elaboración de la especificación de la evaluación
 - 1.2.2. Aprobación e informe
 - 1.3. Diseño de la evaluación
 - 1.3.1. Elaboración del plan de evaluación
 - 1.3.2. Aprobación e informe
 - 1.4. Ejecución de la evaluación
 - 1.4.1. Realización de las acciones del evaluador
 - 1.4.2. Revisión e informe
 - 1.5. Conclusión de la evaluación
 - 1.5.1. Revisión conjunta del informe de evaluación
 - 1.5.2. Devolución, destrucción o archivamiento de los datos y documentos de la evaluación

Cabe mencionar que, en algunas ocasiones, el estándar ISO 14598-5 indica de manera explícita que una sub-actividad se divide, a su vez, en actividades de menor nivel. Por ejemplo, la sub-actividad *Elaboración de la especificación de la evaluación* (1.2.1), se divide en *Análisis de la descripción del producto, Especificación de las mediciones y Verificación de la especificación de evaluación*. Sin embargo, este estándar no indica ninguna tarea para estas sub-actividades, de hecho, prácticamente no se definen tareas para ninguna actividad. Por otro lado, hay ocasiones en las que una sub-actividad se divide de manera explícita, pero no se especifica si es en tareas o en otras sub-actividades. Por ejemplo, la sub-actividad *Realización de las acciones del evaluador* (1.4.1) se divide en *Gestión de los componentes del producto, Gestión de los datos de la evaluación, Gestión del uso de*

herramientas, Evaluación del sitio donde se realizarán las acciones y Gestión de los requisitos sobre técnicas específicas de evaluación, pero no se indica explícitamente si estas últimas son tareas o sub-actividades.

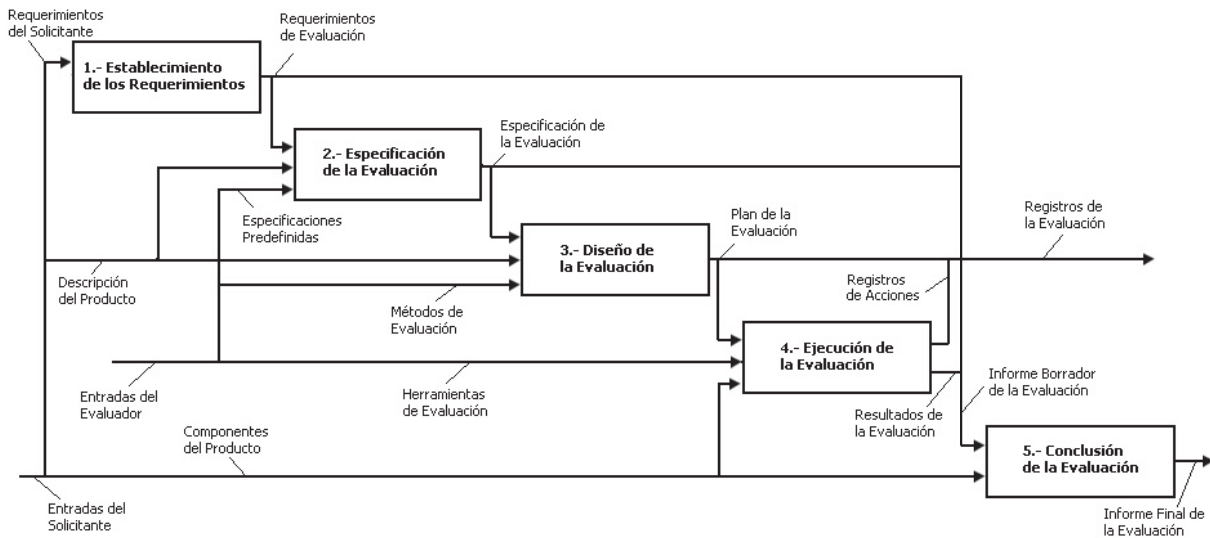


Figura 3-4. Modelo de Proceso de Evaluación de Software (adaptado de [ISO/IEC, 2001b]).

Por otra parte, si bien estos dos estándares ISO identifican las actividades necesarias para los procesos de medición y evaluación, creemos que es necesario que sean reemplazados por un único estándar en el cual se integren ambos procesos, por estar íntimamente relacionados (ver explicación en Sección 2.1.1). La integración implicaría que algunas actividades y/o tareas se fusionaran al tener los mismos propósitos y por ende evitar la redundancia. Un ejemplo de esto son la tarea *Aceptar los requerimientos para la medición* (1.1.1) del proceso de medición y la actividad *Establecimiento de los requerimientos de evaluación* (1.1) del proceso de evaluación, cuyo objetivo común es establecer y especificar los requerimientos. Otro motivo que hace pensar en la necesidad de un nuevo estándar que unifique y armonice a los estándares ISO 15939 y 14598 es la inconsistencia en el estilo adoptado para nombrar las actividades/tareas. Mientras que en ISO 15939 los nombres de las actividades/tareas comienzan con verbos en infinitivo (por ejemplo: *Ejecutar* el proceso de medición), en ISO 14598 se emplea la sustantivación de los verbos para nombrar las actividades (por ejemplo: *Ejecución* de la evaluación). Esta inconsistencia en los estilos de los nombres confunde y a veces dificulta la identificación de actividades y tareas respecto de productos de trabajo.

En lo que respecta a la dinámica del proceso, el estándar ISO 14598-5 indica que hay actividades que se realizan en paralelo. Por ejemplo, en una nota, se dice que la sub-actividad *Verificación de la especificación de evaluación* puede llevarse a cabo en paralelo con las sub-actividades *Análisis de la descripción del producto* y *Especificación de las mediciones*. Sin embargo, no existe un modelo de proceso que permita ver este comportamiento. De igual manera, existen definiciones de trabajo que son opcionales, como *Evaluación del sitio donde se realizarán las acciones* y *Gestión de los requisitos sobre técnicas específicas de evaluación*, pero no se modelan como tales.

Respecto de la vista informacional, el estándar ISO 15939, en uno de sus anexos informativos, indica cuales son los productos de trabajo finales producidos

en las diferentes tareas del proceso de medición. En la Tabla 3-1 se pueden observar los productos de trabajo producidos en cada una de las tareas de las actividades *Planear el proceso de medición* y *Ejecutar el proceso de medición*. Sin embargo, no se indica claramente qué productos de trabajo son entradas de las tareas a llevarse a cabo. Aún más, el estándar no especifica el formato ni el contenido de los productos de trabajo, ni como estos se relacionan con los que se encuentran en el modelo de proceso de la Figura 3-3. El estándar 15939 no muestra ninguna vista informacional del proceso, sino que sólo se limita a darles un nombre a los productos de trabajo.

Tabla 3-1: Productos de trabajo producidos en las tareas de las principales actividades del proceso de medición de ISO 15939.

Tarea	Productos de trabajo producidos
<i>Planear el proceso de medición</i>	
Caracterizar la unidad organizacional	Caracterización de la unidad organizacional
Identificar la necesidad de información	Necesidades de información seleccionadas
Seleccionar las medidas	Modelo de información de medición instanciado para las medidas seleccionadas
	Definición de las medidas seleccionadas
Definir los procedimientos de recolección de datos, análisis, e informe	Procedimientos para recolección, almacenamiento y verificación de datos
	Procedimientos para análisis y reporte de datos
	Procedimientos de gestión de la configuración
Definir criterios para evaluar los productos de información y los procesos de medición	Criterio para la evaluación de los productos de información
	Criterio para la evaluación del proceso de medición
Revisar, aprobar, y proveer recursos para las tareas de medición	Resultados aprobados de planificación de la medición
Implantar tecnologías de soporte	Tecnologías de soporte seleccionadas
<i>Ejecutar el proceso de medición</i>	
Integrar procedimientos	Procedimientos de recolección de datos integrados
Recolectar datos	Datos almacenados
Analizar datos y desarrollar productos de información	Resultados e interpretaciones del análisis de datos
Comunicar resultados	Productos de información

En cuanto al estándar ISO 14598-5, éste indica que las entradas al proceso de evaluación dadas por el solicitante son: la *descripción del producto* y los *componentes del producto*, y las entradas dadas por el evaluador son: *especificaciones de evaluación pre-definidas*, *métodos de evaluación* y *herramientas de evaluación*. Por otra parte, las salidas del proceso de evaluación son: *registros de evaluación* (incluyendo el *plan de la evaluación* y los *registros de las acciones de la evaluación*), el *informe borrador de la evaluación* (incluyendo los *requerimientos de la evaluación*, la *especificación de la evaluación* y los *resultados sintetizados de la evaluación*) y el *informe revisado de la evaluación*. Además de indicar las entradas y salidas, este estándar define claramente el objetivo de cada uno de los productos

de trabajo producidos. Incluso, en algunos casos hasta menciona relaciones entre los productos de trabajo. Por ejemplo, al describir el *plan de evaluación*, dice que éste debería estar “compuesto de” dos partes: la *documentación de los métodos de evaluación* y la *agenda de las acciones del evaluador*. De hecho, en la Figura 3-4 se puede apreciar que se ha intentado reflejar la composición de productos de trabajo; por ejemplo, se puede ver que la entrada del evaluador se divide en tres productos de trabajo (representados por flechas), a saber: *especificaciones de evaluación pre-definidas*, *métodos de evaluación* y *herramientas de evaluación*. Por lo tanto, se puede decir que de alguna manera cuenta con una vista informacional del proceso de evaluación.

Pasando a la vista organizacional, el estándar ISO 15939 menciona que los roles involucrados en el proceso de medición son: *interesado* (stakeholder), *patrocinador* (sponsor), *usuario de la medición* (measurement user), *analista de medición* (measurement analyst), *bibliotecario de medición* (measurement librarian), *proveedor de datos* (data provider) y *responsable del proceso de medición* (measurement process owner). Por otro lado, el estándar ISO 14598-5 indica que los roles que intervienen en la evaluación son: *solicitante* (requester) -que pueden ser a su vez: desarrollador, proveedor, usuario, entre otros- y *evaluador* (evaluator). En ambos estándares se define cada uno de estos roles, incluso en ISO 14598-5 se indican las responsabilidades de aquellos que cubran estos roles. Sin embargo, los roles no son incorporados en ninguno de los modelos de proceso, por lo que hay que leer completamente ambos estándares para poder identificar cuáles son las tareas en las que participa cada uno, lo cual requiere un esfuerzo importante.

3.2.2 CMMI - Capability Maturity Model Integration

Al describir las diferentes áreas de proceso, CMMI menciona cuáles son las actividades involucradas. Por ejemplo, el área de proceso Medición y Análisis comprende las siguientes actividades [CMMI Product Team, 2010]:

- ✓ Especificación de los objetivos de medición y análisis, los cuáles deben estar alineados con las necesidades de información y objetivos del proyecto, de la organización, o del negocio.
- ✓ Especificación de las medidas, técnicas de análisis y mecanismos de recolección de datos, almacenamiento de datos, elaboración de informes y retroalimentación.
- ✓ Aplicación de las técnicas y mecanismos de análisis para la recolección de datos, presentación de datos y la retroalimentación.
- ✓ Proporcionar resultados objetivos que puedan ser utilizados en la toma de decisiones informadas y para adoptar medidas correctivas apropiadas.

Sin embargo, CMMI, más que concentrarse en estas actividades, se centra en definir un conjunto de prácticas específicas y genéricas (*Specific Practices –SP–* y *Generic Practices –GP–*) que permiten lograr ciertas metas específicas y genéricas (*Specific Goals –SG–* y *Generic Goals –GG–*). Un meta genérica es aquella que es común a múltiples áreas de proceso, mientras que una meta específica es aquella que debe lograrse para satisfacer un determinado área de proceso. Por otro lado, las prácticas genéricas aplican a múltiples áreas de proceso, describen las actividades que se consideran importantes en el logro de una meta genérica y contribuyen a la institucionalización de los procesos asociados a un área de

proceso. En contraposición, una práctica específica es la descripción de una actividad que se considera importante en la consecución de la meta específica asociada a un área de proceso. Además, existen sub-prácticas que proporcionan guías para interpretar e implementar una práctica específica o genérica. Particularmente para el área de proceso Medición y Análisis, las metas y prácticas específicas, junto a las sub-prácticas, son:

1. Alinear las actividades de Medición y Análisis (SG)
 - 1.1 Establecer los objetivos de la medición (SP)
 - 1.1.1 *Documentar las necesidades de información y de objetivos.*
 - 1.1.2 *Priorizar las necesidades de información y los objetivos.*
 - 1.1.3 *Documentar, revisar y actualizar los objetivos de medición.*
 - 1.1.4 *Proporcionar realimentación para refinar y clarificar las necesidades de información y los objetivos en la forma que sea necesario*
 - 1.1.5 *Mantener la trazabilidad de los objetivos de medición para las necesidades de información y los objetivos identificados.*
 - 1.2 Especificar medidas (SP)
 - 1.2.1 *Identificar las medidas candidatas en base a los objetivos de medición documentados.*
 - 1.2.2 *Identificar las medidas existentes que ya tratan los objetivos de medición.*
 - 1.2.3 *Especificar las definiciones operativas para las medidas.*
 - 1.2.4 *Priorizar, revisar y actualizar las medidas.*
 - 1.3 Especificar procedimientos de recolección y almacenamiento de datos (SP)
 - 1.3.1. *Identificar las fuentes de datos existentes que se generan a partir de los productos de trabajo, los procesos o las transacciones actuales.*
 - 1.3.2. *Identificar las medidas para las que son necesarios los datos, aunque no se encuentren disponibles en la actualidad.*
 - 1.3.3. *Especificar cómo recoger y almacenar los datos para cada medida requerida.*
 - 1.3.4. *Crear mecanismos y guías de proceso de recogida de datos.*
 - 1.3.5. *Soportar la recogida automática de los datos donde sea apropiado y factible*
 - 1.3.6. *Priorizar, revisar y actualizar los procedimientos de recogida y de almacenamiento de datos.*
 - 1.3.7. *Actualizar en la forma en que sea necesaria las medidas y los objetivos de medición.*
 - 1.4 Especificar procedimientos de análisis (SP)
 - 1.4.1 *Especificar y priorizar los análisis que se van a realizar y los informes que se van a preparar.*
 - 1.4.2 *Seleccionar los métodos y las herramientas apropiados de análisis de datos*
 - 1.4.3 *Especificar los procedimientos administrativos para analizar los datos y para comunicar los resultados.*
 - 1.4.4 *Revisar y actualizar el contenido y el formato propuesto de los análisis e informes especificados.*
 - 1.4.5 *Actualizar las medidas y los objetivos de medición según las necesidades.*
 - 1.4.6 *Especificar los criterios para evaluar la utilidad de los resultados de análisis y para evaluar el comportamiento de las actividades de medición y análisis.*

2. Proveer Resultados de la Medición (SG)
 - 2.1 Obtener datos de la medición (SP)
 - 2.1.1 *Obtener los datos para las medidas base.*
 - 2.1.2 *2. Generar los datos para las medidas derivadas.*
 - 2.1.3 *3. Realizar las comprobaciones de integridad de datos lo más cerca posible a la fuente de los mismos.*
 - 2.2 Analizar datos de la medición (SP)
 - 2.2.1 *Llevar a cabo los análisis iniciales, interpretar los resultados y sacar las conclusiones preliminares.*
 - 2.2.2 *Llevar a cabo mediciones y análisis adicionales según sea necesario y preparar los resultados para su presentación.*
 - 2.2.3 *Revisar los resultados iniciales con las partes interesadas relevantes.*
 - 2.2.4 *Refinar los criterios para análisis futuros.*
 - 2.3 Almacenar datos y resultados (SP)
 - 2.3.1 *Revisar los datos para asegurar que sean completos, íntegros, precisos y actuales.*
 - 2.3.2 *Almacenar los datos conforme a los procedimientos de almacenamiento de datos.*
 - 2.3.3 *Poner los contenidos almacenados disponibles para uso exclusivo de los grupos y el personal apropiado.*
 - 2.3.4 *Prevenir que la información almacenada no sea utilizada inapropiadamente.*
 - 2.4 Comunicar los resultados (SP)
 - 2.4.1 *Mantener informadas regularmente a las partes interesadas relevantes de los resultados de la medición.*
 - 2.4.2 *Ayudar a las partes interesadas relevantes a entender los resultados*

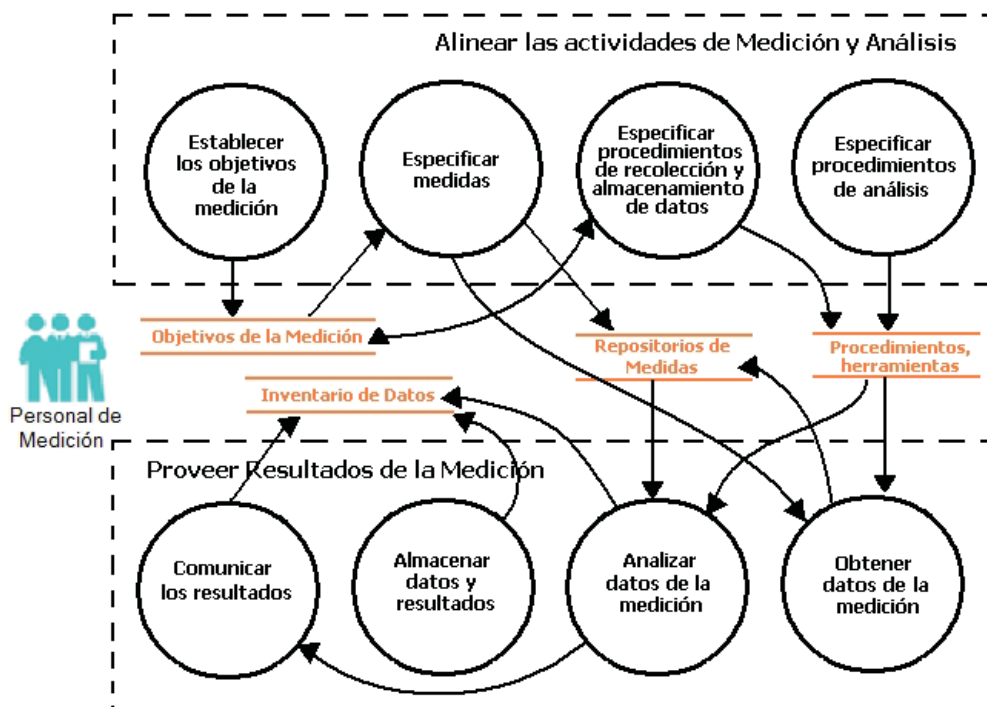


Figura 3-5: Diagrama de contexto de Medición y Análisis en CMMI (adaptado de [Ahern, y otros, 2008]).

En la Figura 3-5 se puede observar un diagrama de flujo de datos (adaptado de [Ahern, y otros, 2008]) en el cual se modelan las prácticas específicas para el área de proceso Medición y Análisis. Sin embargo CMMI no define ningún modelo de proceso. De hecho, en [CMMI Product Team, 2010] se indica que los modelos CMMI no son procesos ni descripciones de procesos. La razón, según indican, es que los procesos reales, los utilizados en una organización, dependen de muchos factores, entre ellos el dominio de aplicación y la estructura y tamaño de la organización. Además, aclaran que las áreas de proceso de un modelo CMMI típicamente no mapean uno a uno con los procesos usados en las organizaciones. Por lo tanto, CMMI se limita a fijar metas que deben alcanzarse para mostrar madurez y prácticas (como actividades/tareas) para alcanzar tales metas, pero no se establecen secuencias, paralelismos, puntos de control, entre otros aspectos relacionados al comportamiento del proceso que subyace.

Tabla 3-2: Productos de trabajo producidos en las prácticas específicas para el área de proceso Medición y Análisis.

Prácticas específicas	Productos de trabajo típicos
<i>Alinear las actividades de Medición y Análisis (SG)</i>	
Establecer los objetivos de la medición	Objetivos de Medición
Especificar medidas	Especificaciones de medidas base y derivadas
Especificar procedimientos de recolección y almacenamiento de datos	Procedimientos de recolección y almacenamiento de datos
	Herramientas de recolección de datos
Especificar procedimientos de análisis	Especificaciones y procedimientos de de análisis
	Herramientas de recolección de datos
<i>Proveer Resultados de la Medición (SG)</i>	
Obtener datos de la medición	Conjuntos de datos de mediciones base y derivadas
	Resultados de los test de integridad de datos
Analizar datos de la medición	Resultados e informes borradores del análisis
Almacenar datos y resultados	Inventario de los datos almacenados
Comunicar los resultados	Informes entregados y resultados de los análisis relacionados
	Información contextual o guía para ayudar a interpretar los resultados del análisis

No obstante, cabe destacar que, si bien CMMI no presenta una vista de comportamiento, al describir la meta específica *Alinear las actividades de Medición y Análisis*, [CMMI Product Team, 2010] indica que las prácticas específicas bajo esta meta pueden ser conducidas concurrentemente o en cualquier orden. Pero, si pensamos de manera lógica, esto no puede ser cierto. Por ejemplo, no se podría *Especificar procedimientos de análisis* si previamente no se realizó la práctica *Especificar medidas*⁵, ya que para decidir los procedimientos de análisis de datos a

⁵ El término *medida* utilizado en CMMI hace referencia al término *métrica* de nuestro marco conceptual C-INCAMI (ver Sección 4.2.4).

utilizar es necesario conocer el tipo de valores que producirán las medidas seleccionadas. Y tampoco se podría especificar correctamente las medidas que se van a utilizar sin *Establecer los objetivos de la medición*, debido a que dependiendo del objetivo que se desee alcanzar, será más apropiado escoger una u otras medidas.

En cuanto a lo que respecta a la vista informacional, si bien no se presenta ningún modelo, [CMMI Product Team, 2010] cuenta con secciones en las cuales se listan ejemplos de posibles productos de trabajo producidos en cada una de las prácticas específicas. En la Tabla 3-2 se pueden apreciar los posibles productos de trabajo producidos en el área de proceso Medición y Análisis. Es importante remarcar que si bien se indican los productos de trabajo resultantes de ejecutar las prácticas específicas, nada se dice acerca de las relaciones entre estos ni cuáles son los productos de trabajo insumidos en cada una de las prácticas. Cabe recordar que si bien es importante saber cuál será la salida de un a tarea, también es muy importante identificar cuáles son las entradas para poder llevarlas a cabo.

Además de ser importante conocer las entradas y salidas de las tareas, ya que en el modelado de proceso contribuyen a la vista funcional, también es de gran valor identificar los roles involucrados en la realización de las diferentes prácticas. Conocer los roles, es decir el conjunto de habilidades necesarias para llevar a cabo una tarea, es fundamental para seleccionar al personal apropiado. Sin embargo CMMI no especifica ningún rol para el área de proceso Medición y Análisis, careciendo, por lo tanto, de información para modelar una vista organizacional.

3.2.3 PSM - Practical Software and Systems Measurement

PSM describe el proceso de medición como un proceso sistemático, pero flexible. Este proceso está basado en la típica secuencia de gestión “*Plan-Do-Check-Act*”, pero adaptado para soportar actividades y tareas específicas de medición. En [PSM, 2000] se presenta una vista de alto nivel del proceso de medición.

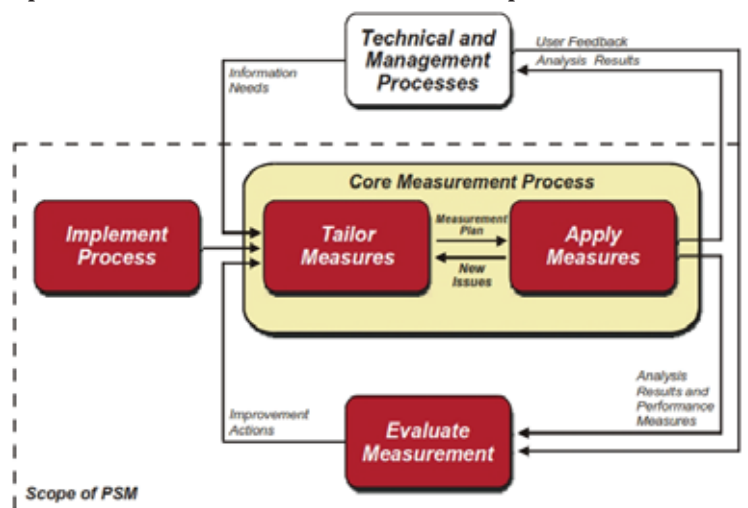


Figura 3-6: Proceso de Medición de PSM.

La Figura 3-6 muestra esta vista de alto nivel en la que se aprecian las cuatro actividades de PSM, a saber: 1) *Ajustar las medidas* (Tailor Measures), 2) *Aplicar las medidas* (Apply Measures), 3) *Implementar el proceso* (Implement Process) y 4) *Evaluar la medición* (Evaluate Measurement). Las dos primeras actividades forman el núcleo del proceso de medición, es decir, son las actividades que permiten

adquirir datos útiles para la toma de decisiones. La actividad *Implementar el proceso*, incluye las tareas para establecer un programa de medición en una organización. Y la actividad *Evaluar la medición* identifica tareas para evaluar y mejorar el programa de medición.

A su vez, cada una de estas actividades se divide en tareas. Particularmente, las actividades que forman el núcleo del proceso cuentan con las siguientes tareas:

1. Ajustar las medidas
 - 1.1. *Identificar y priorizar cuestiones del proyecto* (Identify and Prioritize Project Issues)
 - 1.2. *Seleccionar y especificar las medidas del proyecto* (Select and Specify Project Measures)
 - 1.3. *Integrar en los procesos técnicos y de gestión* (Integrate into the technical and management processes)
2. Aplicar las medidas
 - 1.4. *Recolectar y procesar los datos* (Collect and Process Data)
 - 1.5. *Analizar las cuestiones* (Analyze Issues)
 - 1.6. *Hacer recomendaciones* (Make Recommendations)

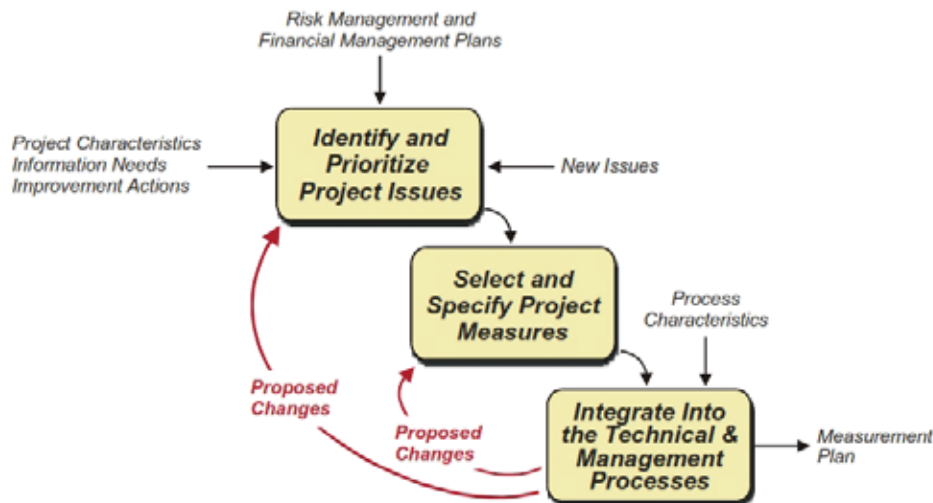


Figura 3-7: Vista funcional de la actividad *Ajustar las medidas*.

En [PSM, 2000] puede hallarse la vista funcional del proceso de medición, tal como se muestra en Figura 3-6, como así también de cada una de las cuatro actividades de PSM. Por ejemplo, en las Figuras 3-7 y 3-8 pueden observarse la vista funcional de la actividad *Ajustar las medidas* y *Aplicar las medidas*, respectivamente.

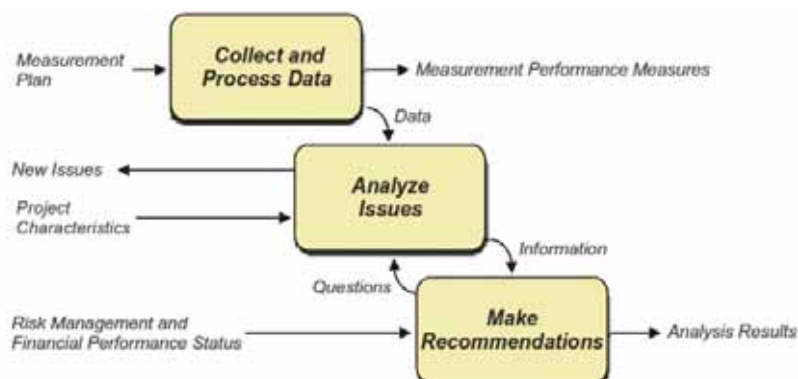


Figura 3-8: Vista funcional de la actividad *Aplicar las medidas*.

Cabe mencionar que, con la creación del estándar ISO 15939, los autores decidieron adaptar los nombres de las actividades de PSM con el objetivo de compartir la terminología con tal estándar. De esta manera, las cuatro actividades de PSM pasaron a llamarse: 1) *Planificar la Medición* (Plan Measurement), 2) *Ejecutar la Medición* (Perform Measurement), 3) *Establecer y Mantener el Compromiso* (Establish and Sustain Commitment), y 4) *Evaluar la Medición* (Evaluate Measurement). En la Figura 3-9 se observan estos cambios. Incluso, las tareas han sido modificadas con el pasar del tiempo. Según [Cree, y otros, 2008] los nombres de las tareas correspondientes a las actividades núcleo son:

1. Planificar la Medición
 - 1.1. Identificar las necesidades de información
 - 1.2. Seleccionar, priorizar, y especificar las medidas correspondientes
 - 1.3. Preparar un plan de medición
2. Ejecutar la Medición
 - 2.1. Recolectar y validar los datos
 - 2.2. Procesar y analizar los datos
 - 2.3. Generar informes
 - 2.4. Hacer recomendaciones

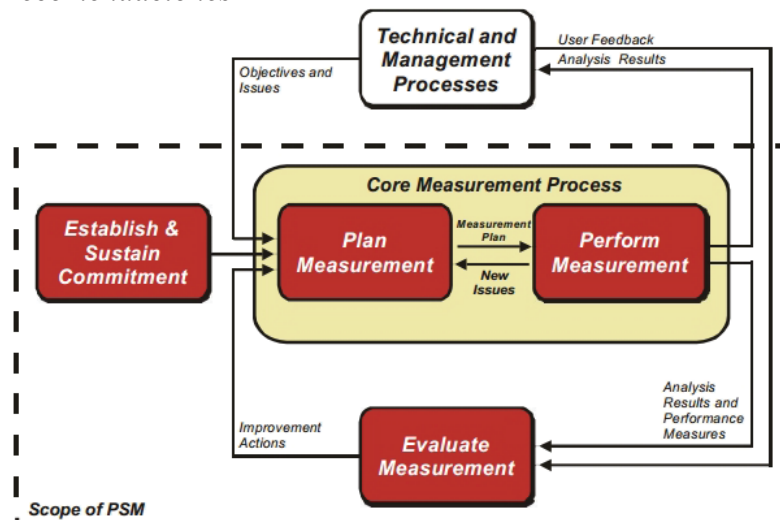


Figura 3-9: Proceso de Medición de PSM usando la terminología de ISO 15939.

En lo referido a definición de roles, en [PSM, 2000] se indica que es importante definir quién es responsable de cada parte del proceso de medición para alcanzar una implementación exitosa. Los roles definidos en este documento son: *Director ejecutivo* (Executive manager), *Administrador técnico o de proyecto* (Project or technical manager), *Analista de medición* (Measurement Analyst) y *Equipo de proyecto* (Project team). Por otro lado, en un documento más reciente [Cree, y otros, 2008] se señalan los siguientes roles: *Administrador de proyecto* (Project Manager), *Miembro del equipo* (Team Member), *Analista de medición* (Measurement Analyst) y *Director ejecutivo* (Executive Manager). En ambos documentos se indican las responsabilidades que comprende cada rol y, según el último documento citado, todos los roles intervienen, de una u otra manera, en todas las actividades. Sin embargo, no se mencionan puntualmente las tareas en las que participa cada rol -aunque quizás se puedan llegar a identificar mediante las responsabilidades- y tampoco se presenta un modelado del proceso desde una perspectiva organizacional.

Software Measurement Plan Outline
<p>Part 1 – Introduction</p> <ul style="list-style-type: none"> • Purpose and Scope <p>Part 2 – Project Description</p> <ul style="list-style-type: none"> • Technical and Project Management Characteristics <p>Part 3 – Measurement Roles, Responsibilities, and Communications</p> <ul style="list-style-type: none"> • How Measurement is Integrated into the Technical and Management Processes • Measurement Points of Contact (acquirer, supplier, subcontractors) • Measurement Responsibilities • Organizational Communications and Interfaces • Tools and Databases • Phased Implementation (if applicable) • Evaluation Criteria <p>Part 4 – Description of Project Issues</p> <ul style="list-style-type: none"> • Organizational Goals / Issues • Prioritized List of Issues and Objectives <p>Part 5 – Measurement Specifications</p> <ul style="list-style-type: none"> • Include for Each Selected Measure (for each organization) <ul style="list-style-type: none"> – Measure Name – Project-specific issue the Measure Maps to – Data Items – Attributes – Aggregation Structures – Collection Level – Criteria for Counting Actuals – Data Definitions – Estimations Methodology, Models, and Historical Data – Collection and Reporting Mechanisms – Source of Data – Collection and Reporting Periodicity – Applicable Phases and Activities <p>Part 6 – Project Aggregation Structures</p> <ul style="list-style-type: none"> • Component Aggregation Structure, such as Class, Units • Activity Aggregation Structure, such as Requirements Analysis, Design, Implementation, and Integration and Test • Functional Aggregation Structure <p>Part 7 – Initial Indicators</p> <ul style="list-style-type: none"> • Include for Each Proposed Indicator: <ul style="list-style-type: none"> – Indicator Name – Project-Specific Issue the Indicator Maps to – Measures Used to Construct the Indicator – Sample Display Format – Interpretation and Decision Related to the Indicator <p>Part 8 – Reporting Mechanisms and Periodicity</p> <ul style="list-style-type: none"> • Reporting Mechanism and Periodicity • Content Reports

Figura 3-10: Ejemplo de composición del Plan de Medición (tomado de [PSM, 2000]).

Respecto a la vista informacional, en los documentos observados de PSM no se ha hallado modelos en los que se aprecie composición, o relaciones entre los diferentes productos de trabajo consumidos o producidos. Sin embargo se presentan plantillas o ejemplos, como el que se ve en la Figura 3-10, donde se pueden apreciar la composición de los mismos. Finalmente, cabe mencionar que tampoco se ha hallado una vista de comportamiento del proceso de PSM. Si bien, observando el flujo de los productos de trabajos mostrados en los modelos de proceso se puede llegar a identificar secuencias e iteraciones, como por ejemplo en la Figura 3-7, lo cierto es que en la documentación observada de PSM no hay

modelos específicos que describan la dinámica del proceso.

3.2.4 Alarcos Quality Center

La Metodología CQA-Meth (*Continuos Quality Assessment Methodology*) [Rodríguez, y otros, 2010a] o EVVE (*Entorno para la Verificación y Validación de Especificaciones software*) [Rodríguez, y otros, 2010b] está formada por tres procesos principales (ver Figura 3-11), a saber: *Gestión de la Evaluación*, *Evaluación* y *Gestión de la Infraestructura*.



Figura 3-11: Proceso y fases de la metodología de evaluación EVVE (tomado de [Rodríguez, y otros, 2010b]).

A su vez, el proceso de Evaluación se divide en cuatro fases: Planificación, Especificación, Ejecución y Conclusión. Cada una de estas fases agrupa un conjunto de actividades de las cuales sólo se indica el nombre. A continuación se presenta la estructura de desglose del proceso Evaluación (raíz), donde los elementos hojas son actividades y los intermedios son fases:

1. Evaluación

1.1 Planificación

- 1.1.1 *Contratación.*
- 1.1.2 *Arranque del proyecto*
- 1.1.3 *Planificación detallada*
- 1.1.4 *Consolidación del plan*

1.2 Especificación

- 1.2.1 *Obtención y análisis de artefactos a evaluar*
- 1.2.2 *Selección del modelo de calidad y las técnicas*
- 1.2.3 *Planificación interna de la evaluación*
- 1.2.4 *Verificación de la especificación*

1.3 Ejecución

- 1.3.1 *Aplicación de técnicas de evaluación*
- 1.3.2 *Análisis de la ejecución*
- 1.3.3 *Unificación de resultados*

1.4 Conclusión

- 1.4.1 *Elaboración del informe de evaluación*
- 1.4.2 *Presentación de resultados*
- 1.4.3 *Corrección del informe y finalización de la evaluación*

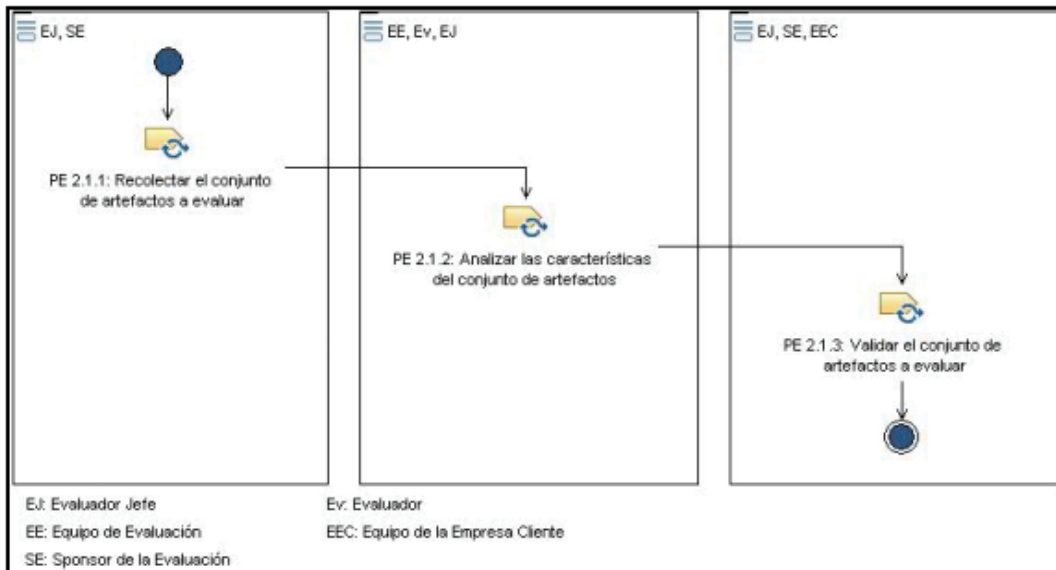


Figura 3-12: Vista organizacional y de comportamiento de la actividad *Obtención y Análisis de Artefactos a Evaluar* (tomado de [Rodríguez, y otros, 2010b]).

Si bien en la literatura consultada no se han encontrado las tareas que forman parte de cada una de las actividades mencionadas, en [Rodríguez, y otros, 2010b] se puede observar el modelo de la Figura 3-12 en el cual se aprecian las tareas de la actividad *Obtención y Análisis de Artefactos a Evaluar* (1.2.1). El modelo no solo presenta las tareas, sino que también modela la dinámica (secuencia de tareas) y los roles involucrados. Claramente los autores en un mismo modelo han tenido en cuenta la vista de comportamiento y organizacional del proceso de Evaluación del EVVE.

Es digno de mención que, además de identificarse los roles involucrados en cada tarea, también se modela la relación que existe entre ellos. Según la Figura 3-13, se puede observar que la comunicación entre las empresas (cliente y evaluadora) es realizada a través del patrocinador (o Sponsor) y del evaluador jefe, y a su vez, éstos serán los que se comuniquen directamente con sus equipos internos.

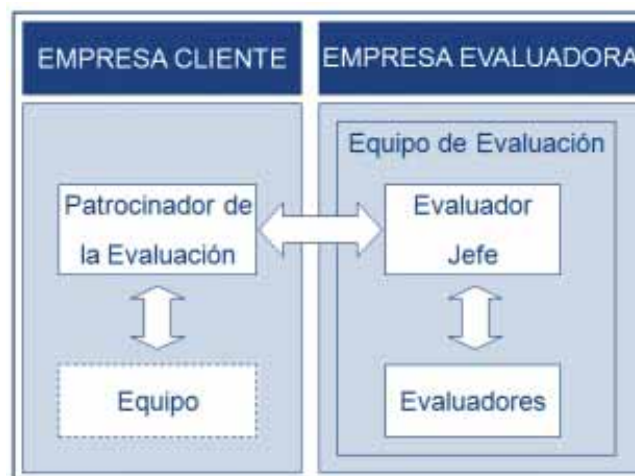


Figura 3-13: Roles y relaciones en el proceso de evaluación.

Por otro lado, es importante destacar que EVVE cuenta con un catálogo de productos de trabajo, o elementos como los llaman los autores. Algunos de estos

productos de trabajo son: *Contrato de evaluación, Plan de evaluación, Especificación de la evaluación, Informe de evaluación*, entre otros. Del producto de trabajo *Contrato de evaluación* se dice que está dividido en dos partes (*Propuesta técnica de evaluación y Propuesta económica de evaluación*), que bien podrán estar incluidas en el mismo documento o en dos documentos separados. Sin embargo, a pesar de mencionarse relaciones como la citada entre los productos de trabajo, no se ha hallado una vista informacional. Por otro lado, si bien en la documentación consultada se indica la fase en la cual son producidos algunos de los productos de trabajo, no se indican específicamente las actividades o tareas involucradas, ni se modela el proceso desde una vista funcional en la cual se pueda ver el flujo de los productos de trabajo.

3.2.5 GQM+Strategies - Goal Question Metric Plus Strategies

El proceso de GQM+Strategies, si no se tienen en cuenta las actividades de enlace entre objetivos de diferentes niveles, es el mismo que en GQM. Las actividades para llevar a cabo un programa de medición basado en GQM pueden observarse en el siguiente diagrama de desglose:

1. GQM process
 - 1.1. Prestudy
 - 1.1.1. Identify available inputs, preconditions & constraints
 - 1.1.2. Characterize organization and identify organizational improvement goals
 - 1.1.3. Identify and characterize candidate application projects
 - 1.1.4. Select project and identify project goals
 - 1.2. Identify GQM goals
 - 1.2.1. Specify measurement goals informally
 - 1.2.2. Specify GQM goals
 - 1.2.3. Rank and select GQM goals
 - 1.3. Produce GQM plan
 - 1.3.1. Hold first interviews
 - 1.3.2. Merge abstraction sheets
 - 1.3.3. Refine GQM plan
 - 1.3.4. Hold follow-up interviews
 - 1.3.5. Review GQM plan
 - 1.4. Produce measurement plan
 - 1.4.1. Define collection procedures
 - 1.4.2. Check measurement plan
 - 1.5. Collect and validate data
 - 1.5.1. Collect data
 - 1.5.2. Validate data
 - 1.5.3. Store data
 - 1.6. Analyze data
 - 1.6.1. Prepare presentation material
 - 1.6.2. Self study of presentation material
 - 1.6.3. Perform feedback session
 - 1.7. Package

En [Gresse, y otros, 1995] se describe extensamente el proceso de GQM. Al describir cada actividad, se incluye un diagrama de flujo de producto (utilizando la

notación MVP-L [Bröckers, y otros, 1995b]), una breve descripción de la actividad, una detallada descripción de la “interfaz” (productos consumidos, modificados y producidos, criterios de entrada y salida –pre y pos condiciones-, y roles involucrados) y, en muchos casos, una descripción de la actividad refinada. En la Figura 3-14 se puede ver un diagrama de flujo de producto que modela la vista funcional del proceso general de GQM.

En la documentación consultada también se incluyen diagramas de tiempo (*timeline diagram*) que permiten apreciar secuencias y paralelismos de las actividades. A modo de ejemplo, en la Figura 3-15 se puede observar un diagrama de tiempo del proceso general de GQM que permite ver, entre otras cosas, que la actividad *Producir el plan de medición* (Produce measurement plan) debe realizarse luego de *Producir el plan GQM* (Produce GQM plan), y que *Analizar los Datos* (Analyze data) puede hacerse (en cierta medida) en paralelo con la actividad *Recolectar y validar los datos* (Collect and validate data). Además es destacable comentar que en GQM también se exponen los criterios de entrada y salida, es decir, pre y pos condiciones, que deben tenerse en cuenta al inicio y final de cada actividad, respectivamente. Esto, junto con los diagramas de tiempo, contribuye al modelado de la vista de comportamiento del proceso.

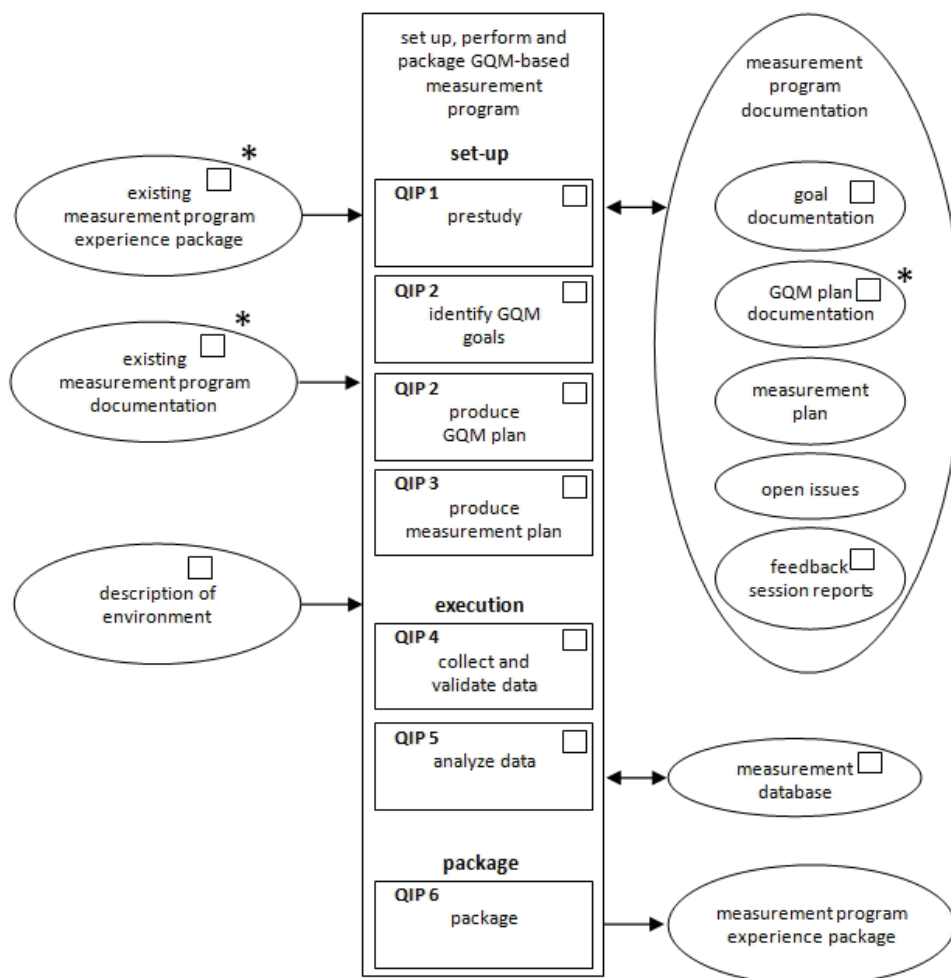


Figura 3-14: Vista funcional del proceso general de GQM (tomado de [Gresse, y otros, 1995]).

Básicamente, existen dos roles en GQM [Gresse, y otros, 1995]: *equipo GQM* (GQM team) y *equipo de proyecto* (Project team). Los miembros del *equipo GQM*

son los responsables de la planificación y ejecución del programa de medición. El *equipo de proyecto* solo participa en algunos pasos del proceso GQM. A su vez, en [Solingen, y otros, 1999] se menciona que en el *equipo GQM* se encuentran los siguientes roles: *encargado* (manager), *entrenador* (coach) e *ingeniero de soporte* (support engineer), y además se mencionan otros como *equipo de definición GQM* (GQM definition team) y *equipo de análisis GQM* (GQM analysis team). Sin embargo no se han encontrado modelos que presenten una vista organizacional, es decir, que permitan identificar las actividades en las cuales los roles están involucrados.

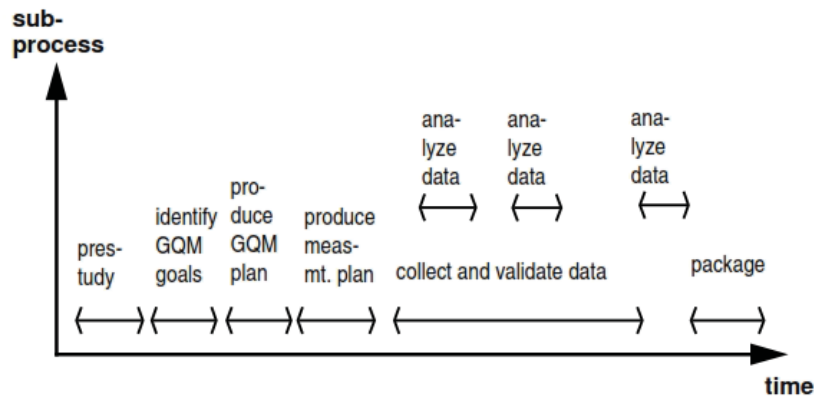


Figura 3-15: Vista de comportamiento del proceso general de GQM utilizando un diagrama de tiempo.

Finalmente, en lo que respecta a la vista informacional, se puede observar que existen modelos que permiten ver relaciones de composición entre los productos de trabajo. Por ejemplo, en la Figura 3-14 se puede ver que la *documentación del programa de medición* (measurement program documentation) está compuesta por la *documentación de objetivos* (goal documentation), la *documentación del plan GQM* (GQM plan documentation) y el *plan de medición* (measurement plan), entre otros. A su vez, algunos de estos productos de trabajo se componen de otros, como por ejemplo la *documentación del plan GQM* (ver Figura 3-16). Incluso en [Solingen, y otros, 1999] se presentan plantillas de los productos de trabajo.

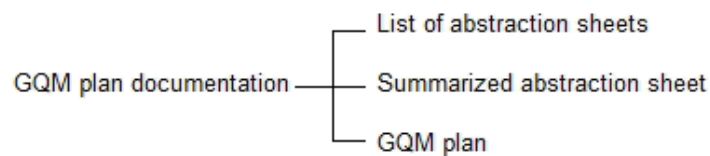


Figura 3-16: Vista informacional del producto de trabajo *documentación del plan GQM* (tomado de [Gresse, y otros, 1995]).

3.3 Marcos Conceptuales de Proceso

Son muchos los trabajos relacionados a procesos, y de igual manera son muchas las interpretaciones que se dan a cada uno de los términos del dominio de proceso. Esto se debe a que, a la fecha, aún no hay amplio consenso sobre todos los términos y su significado. Con el fin de mostrar tal falta de consenso, a continuación se comentan algunos de los trabajos más relevantes en lo que respecta a definición de una base o marco conceptual de proceso.

Entre los pioneros en la definición de una terminología para proceso, se puede mencionar a [Feiler, y otros, 1993]. Los autores definen un conjunto de

términos básicos e indican que suponen que estos términos sufrirán un cambio evolutivo y que muchos términos más se añadirán en el futuro. Los términos presentados están agrupados en cuatro categorías: marco para la definición de procesos, ingeniería de proceso, ejecución del proceso y propiedades del proceso. Introducen los conceptos “paso de proceso” (*process step*) y “elemento de proceso” (*process element*). El primer término representa a una actividad discreta, de duración finita y con un nivel de abstracción que depende del contexto. Según se indica en el mismo trabajo, una tarea es un paso de proceso. El segundo término se define simplemente como un componente de un proceso, y los autores mencionan que un elemento de proceso varía desde pasos de proceso individuales hasta partes muy grandes del proceso. Por lo tanto, un “elemento de proceso” parece englobar desde tareas hasta sub-procesos. Sin embargo, al revisar la definición de proceso propuesta en [Conradi, y otros, 1993], a saber: “todos los *elementos* del ‘mundo real’ involucrados en el desarrollo y mantenimiento de un producto, por ejemplo, artefactos, tecnologías de producción, actividades, agentes y soporte de proceso” (cursivas agregadas), se puede apreciar que el término “elemento de proceso” aquí es más abarcador. Por otro lado, en [Conradi, y otros, 1993], a diferencia de [Feiler, y otros, 1993], se define el término actividad y se indica que es un paso o una operación atómica o compuesta de un proceso, pero no define el término tarea. Otros de los conceptos que define son agente, herramienta, rol, proyecto y artefacto, entre otros.

Además de los trabajos anteriores, se puede citar a [Lonchamp, 1993], que, aparte de definir muchos de los términos básicos del dominio de proceso, realiza una clara diferenciación entre los términos tarea y actividad, algo que en muchos trabajos, incluso actuales, no se tiene en cuenta. Según el autor, a una tarea, a diferencia de una actividad, se le asignan recursos, más precisamente, recursos humanos, tecnológicos, temporales y monetarios. Por lo tanto, a una tarea se le programa, ejecuta y controla, mientras que a una actividad se la planifica.

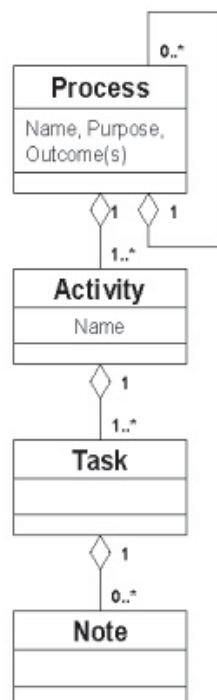


Figura 3-17: Constructores de proceso según ISO/IEC 12207/15288 (tomado de [ISO/IEC, 2004]).

Entre los trabajos más actuales podemos citar al estándar ISO 12207 [ISO/IEC, 2004], el cual define los procesos principales, de soporte y organizacionales para organizaciones de Software. Al igual que los trabajos anteriores, éste cuenta con un glosario de términos, pero además presenta un diagrama en el cual se pueden ver las relaciones entre los términos proceso, actividad y tarea. Como se observa en la Figura 3-17, un proceso puede agrupar a otros procesos y, además, contiene al menos una actividad. A su vez, una actividad agrupa a una o más tareas relacionadas. El diagrama también muestra que un proceso requiere contar con un propósito y salidas. En el modelo de la Figura 3-17 también aparece el concepto “nota”, el cual se utiliza para agregar información con el fin de describir mejor una tarea, aunque claramente no es un concepto propio del dominio de proceso.

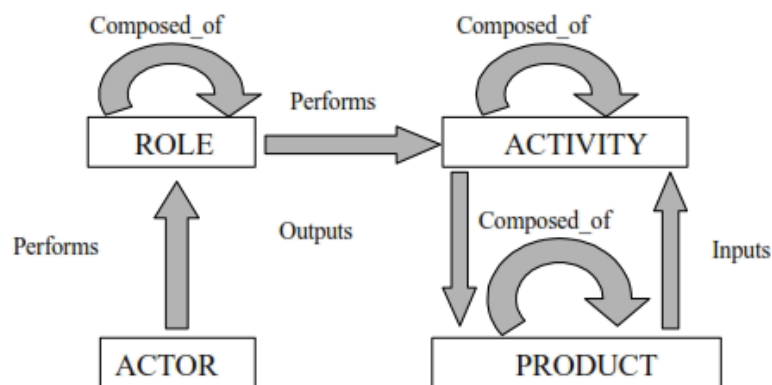


Figura 3-18: Componentes básicos de modelado de proceso según [Acuña, y otros, 2001b].

Existen también trabajos que no se han conformado con realizar simplemente un glosario de términos, sino que han intentado identificar las relaciones que existen entre los diferentes conceptos del dominio de proceso. Un ejemplo de esto es [Acuña, y otros, 2001b] donde los autores definen un conjunto de términos básicos y muestran las relaciones entre los elementos (ver Figura 3-18). Si bien no definen estas relaciones, indican que las mismas pueden hallarse en [Benali, y otros, 1992]. A pesar de ser pocos los términos utilizados, es un buen avance hacia un marco conceptual más robusto como lo es una ontología.

En esta dirección se puede citar el trabajo [Esteban, y otros, 2003], donde los autores, no solo identifican conceptos y relaciones entre estos, sino que también identifican atributos para cada uno de los elementos del dominio (ver Figura 3-19). Este trabajo tiene como fuente algunos de los escritos ya comentados. Una particularidad es que en este artículo aparece un concepto no tenido en cuenta en los trabajos revisados, a saber: método. Este término está ligado al dominio de procesos, ya que cuando hablamos de procesos hablamos de un conjunto de actividades que deben implementarse, y justamente un método es la manera de implementar o llevar a cabo la descripción de una actividad. En cuanto a los atributos, estos están identificados pero no definidos. También cabe comentar que los autores hacen uso de una plantilla derivada a partir del modelo conceptual con el objetivo de contener de manera consistente toda la información referente a una actividad y así facilitar el almacenamiento, consulta, explotación y reutilización de la misma.

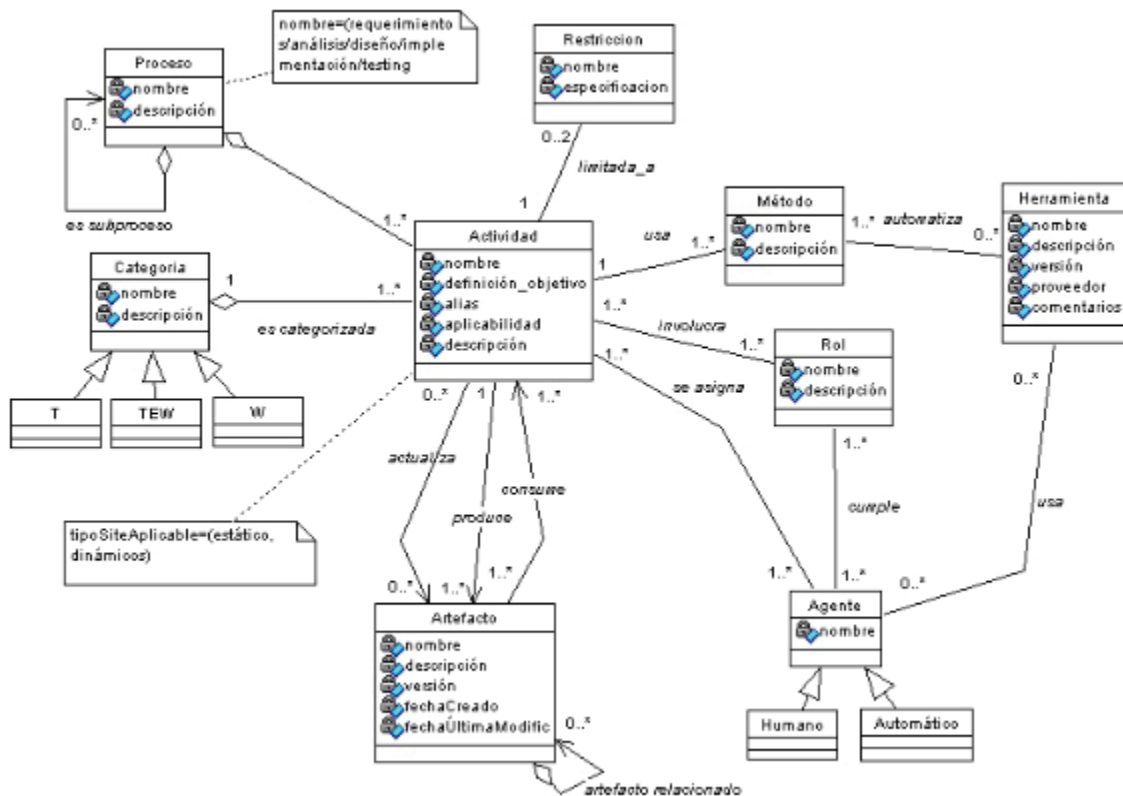


Figura 3-19: Modelo conceptual para el dominio de procesos Web (tomado de [Esteban, y otros, 2003]).

En 2002, en un intento por estandarizar los términos del dominio de proceso, la OMG (*Object Management Group*) lanza SPEM (*Software & Systems Process Engineering Meta-Model Specification*), un metamodelo de ingeniería de proceso como así también un marco conceptual, el cual provee los conceptos necesarios para modelar, documentar y gestionar procesos [OMG-SPEM, 2008]. Sin embargo la primera versión no tuvo mucho éxito. Fue recién con la versión 2.0 cuando SPEM logró ser adoptada por la comunidad. Uno de los cambios introducidos en SPEM 2.0 respecto a versiones anteriores fue la modificación de algunos términos. Por ejemplo, “goal” fue renombrado por “postcondition”, ya que una “definición de trabajo” (*work definition*) tenía asociada “restricciones” (*constraints*) las cuales eran “preconditions” y “goals”, y esto causaba confusión. Otro concepto sustituido fue “process performer” por “role use” y, además, se agregaron los conceptos “composite role” y “team profile” para hacer referencia a roles más generales o grupos. En la versión 1.1 de SPEM un “proceso” (*process*) era un “componente de proceso” (*process component*), el cual a su vez era un “paquete” (*package*) -ver Figura 3-20-; estas relaciones de especialización (de UML) se modificaron, ya que pensar que un proceso “es un” paquete o algún tipo de contenedor físico es algo extraño. Un concepto que se eliminó fue “ciclo de vida” (*life cycle*) debido a que con los cambios introducidos en la nueva versión este pasaba a ser sinónimo de “proceso” y por lo tanto era redundante. Estos cambios, junto con la notación visual atractiva y la adopción de UML 2.0 hicieron que SPEM llegue a utilizarse en gran medida como lenguaje estándar para modelar procesos de software.

Sin embargo, un punto en contra de SPEM a la hora de verlo como un marco conceptual para el dominio de proceso es que su estructura orientada a objetos

hace que existan conceptos que no son propios del dominio de proceso. Por ejemplo, “rol compuesto” (*composite role*) y “producto de trabajo” (*work product use*) son una especialización (de UML) de “elemento de desglose” (*breakdown element*), obviamente este último término no pertenece al dominio, sino que fue introducido con el objetivo de poder indicar que tanto un rol compuesto como un producto de trabajo pueden expresarse como una estructura jerárquica de componentes y así aumentar las capacidades del metamodelo. También existen distinciones entre conceptos como “*task definition*” y “*task use*” que agregan complejidad innecesaria desde el punto de vista conceptual. Mientras que el primer concepto hace referencia a la definición de una tarea, por ejemplo para ser almacenada en un repositorio de tareas, el segundo concepto hace referencia a una tarea concreta utilizada en un proceso real. Claramente esta diferenciación no tiene que ver con el dominio. Lo mismo ocurre con conceptos tales como “*work product definition*” y “*work product use*” o “*role definition*” y “*role use*”.

También cabe mencionar que en SPEM 2.0 se define un “proceso” como un tipo especial de “actividad”, siendo así “actividad” un concepto genérico que incluye a “proceso”, junto a los conceptos “iteración” (*iteration*) y “fase” (*phase*), como especialización. Por lo tanto, esta jerarquía entre proceso y actividad no sigue los tres niveles de descomposición utilizados en el modelo de proceso de ISO 12207 (recordar Figura 3-17): procesos agrupan actividades, y estas agrupan tareas.

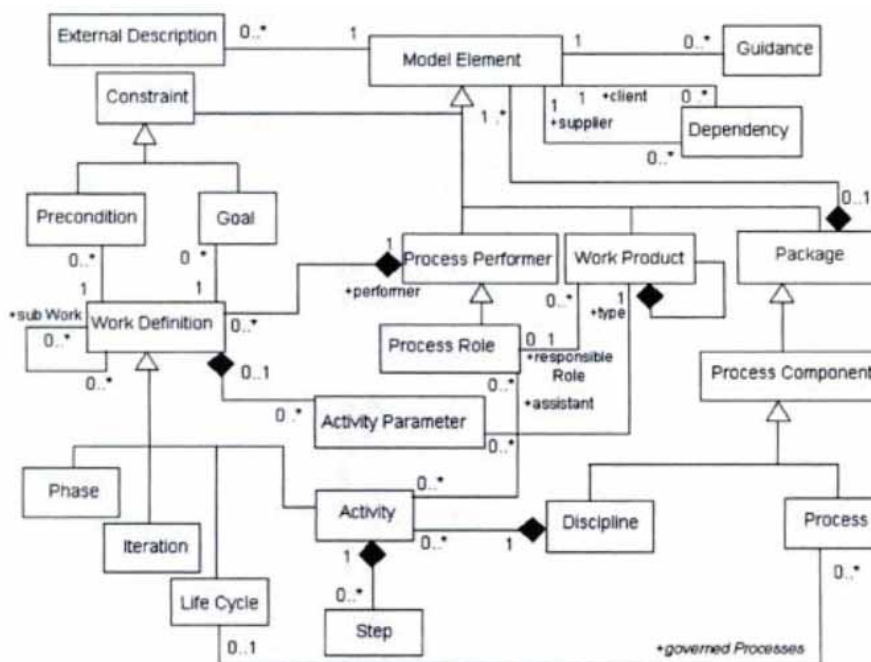


Figura 3-20: Principales conceptos del metamodelo de SPEM 1.1.

Otro trabajo digno de mencionar es [Bringunte, y otros, 2011] en el cual los autores definen una *Ontología de Proceso de Software* (SPO, por sus siglas en inglés). Realmente es un trabajo muy completo; contiene conceptos como “Organización” y “Proyecto” y, además, tiene en cuenta la idea de planificación y programación de proyectos. Su ontología se apoya sobre una ontología fundacional llamada UFO (*Unified Foundational Ontology*). Si bien este hecho puede darle robustez a la ontología de proceso, tal como sus creadores lo indican, al mismo tiempo les juega en contra. Por ejemplo, en versiones previas de SPO [Guizzardi, y otros, 2008] se muestra que “recurso hardware”, “recurso software” y “recurso

humano” son especializaciones (de UML) de “recurso”. Sin embargo en la ontología presentada en [Bringunte, y otros, 2011] se aprecia que un “recurso humano” ya no es un “recurso” (ver Figura 3-21). Este cambio se debe a que un “recurso” de SPO es un “objeto” de UFO, y como desde el punto de vista semántico un “recurso humano” no puede ser un “objeto” se debió quitar tal relación. Además no tiene de manera explícita los conceptos fase (las llama macro-actividades), tarea (aunque usa el término “actividad atómica”) y pre y post condición. Tampoco asocia atributos a los conceptos, lo cual le daría mayor robustez.

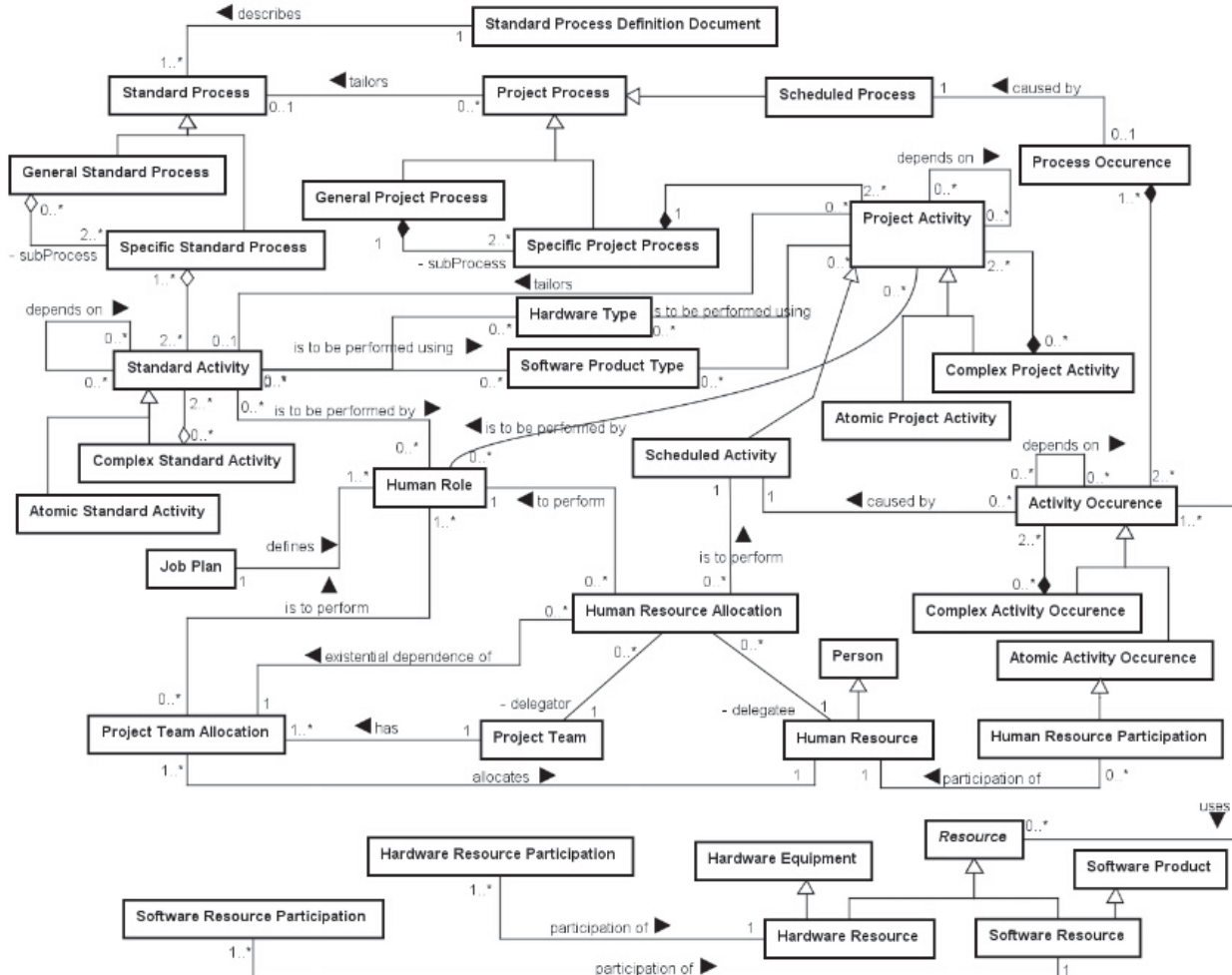


Figura 3-21: Principales conceptos de la *Ontología de Proceso de Software* (tomado de [Bringunte, y otros, 2011]).

Finalmente, se puede mencionar *OPEN Proces Framework* (OPF) [OPF], un repositorio extensible de componentes de proceso predefinidos. Algunos de los conceptos del metamodelo de OPF se aprecian en la Figura 3-22. A diferencia de los trabajos presentados, utiliza el término “productor” (*producer*) para hacer referencia a los agentes que intervienen en la realización de una actividad o tarea. En la definición de “productor” se indica que produce (crea, evalúa, itera o mantiene) versiones de uno o más productos de trabajo relacionados y provee uno o más servicios relacionados. Sin embargo el concepto “servicio” no existe en el metamodelo, aunque se define en un glosario. Tampoco cuenta con los conceptos “recurso” y “pre” y “post condición”. En cuanto a la definición de “herramienta” (*tool*) se indica que es una aplicación de software que es usada por una o más personas. Sin embargo una herramienta no solo es una aplicación de software, sino que también pueden ser elementos físicos (por ejemplo, hardware), o una

combinación de ambos. Cabe mencionar que, si bien indican las relaciones que existen entre los conceptos, no se expresa o define el significado de las mismas, al menos de manera explícita.

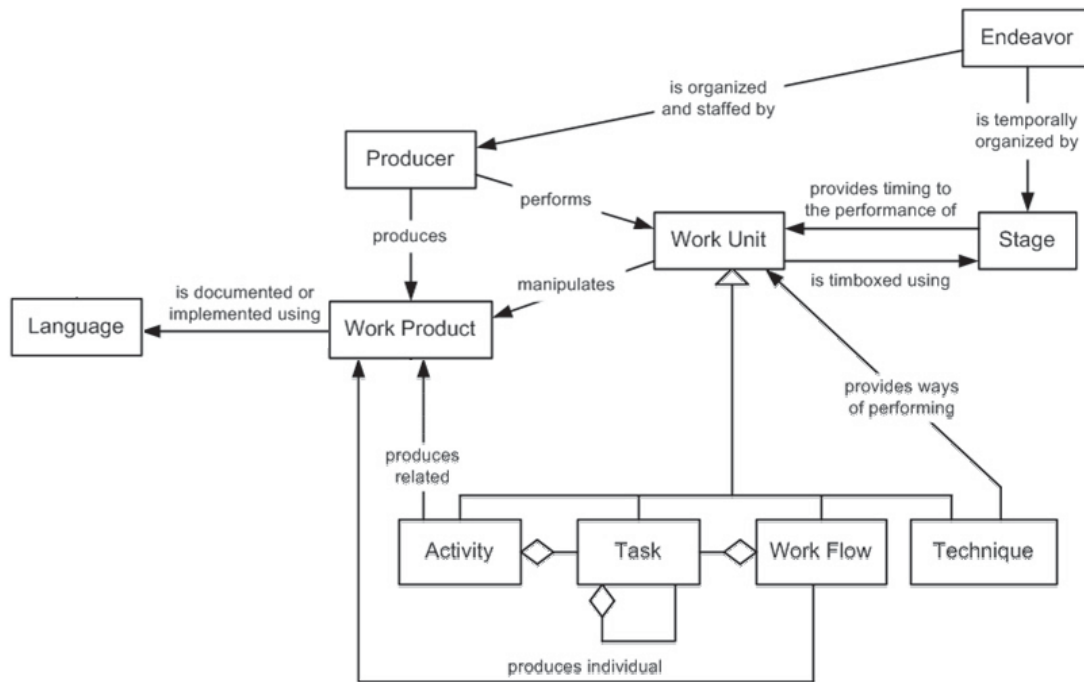


Figura 3-22: Principales componentes de proceso de OPF y sus relaciones (adaptado de [OPF]).

3.3.1 Marco Conceptual de Proceso Propuesto

En los trabajos antes comentados se puede notar la falta de consenso que existe respecto a los términos y el significado de los mismos para el dominio de proceso. Por ejemplo, hay trabajos en los cuales los términos “actividad” y “tarea” son sinónimos, o directamente no se definen aunque son ampliamente utilizados (como sucede en el caso de [CMMI Product Team, 2010]). Incluso existen trabajos que hacen uso de términos no tan frecuentemente usados. En este sentido se puede citar [Bringunte, y otros, 2011] en el cual las actividades son llamadas “actividades complejas” (*complex activities*) y las tareas “actividades atómicas” (*atomic activities*). También existen términos que dan lugar a confusión, por ejemplo el término “elemento de proceso” (*process element*). Este término, en [CMMI Product Team, 2010], es utilizado para referirse a “la unidad fundamental de un proceso” (es decir, representa una tarea o una actividad según se indica en la nota del glosario), sin embargo “elemento de proceso”, en [OMG-SPEM, 2008] y en [IEEE, 2004], no solo hace referencia a tareas o actividades, sino también a agentes y productos de trabajo, lo cual es lógico ya que todos ellos son elementos que intervienen en un proceso.

En vista de esta falta de amplio consenso en el dominio de procesos, realizamos un nuevo marco conceptual para este dominio [Becker, y otros, 2013; Becker, y otros, 2014]. Este nuevo marco explicita con claridad el significado de los términos relacionados a proceso que se utilizarán en este trabajo. Cabe acotar, que el marco conceptual propuesto toma como base un trabajo previo de nuestro grupo de I+D [Olsina, 1997; Olsina, 1998], el cual se apoyaba a su vez en trabajos seminales tales como [Humphrey, 1989], [Feiler, y otros, 1993] y [Lonchamp, 1993], entre otros. Además, el nuevo marco también está construido teniendo en

cuenta trabajos más recientes en el área de proceso como son SPEM, CMMI e ISO 12207.

Para la construcción de la base conceptual de proceso se han usado elementos de una ontología. Esta incluye los principales conceptos del dominio de proceso, sus definiciones, relaciones entre los conceptos y propiedades de los conceptos. Una ontología con estas características es denominada ontología liviana o ligera, en contraposición con las ontologías pesadas, que además de lo mencionado cuentan con axiomas y restricciones [Corcho, y otros, 2003]. En la Figura 3-23 se muestra un diagrama de Clases de UML donde se pueden apreciar los términos fundamentales del dominio de proceso (ver definiciones en Tabla 3-3), junto con sus relaciones (ver definiciones en Tabla 3-4) y atributos o propiedades principales (ver definiciones en Tabla 3-5).

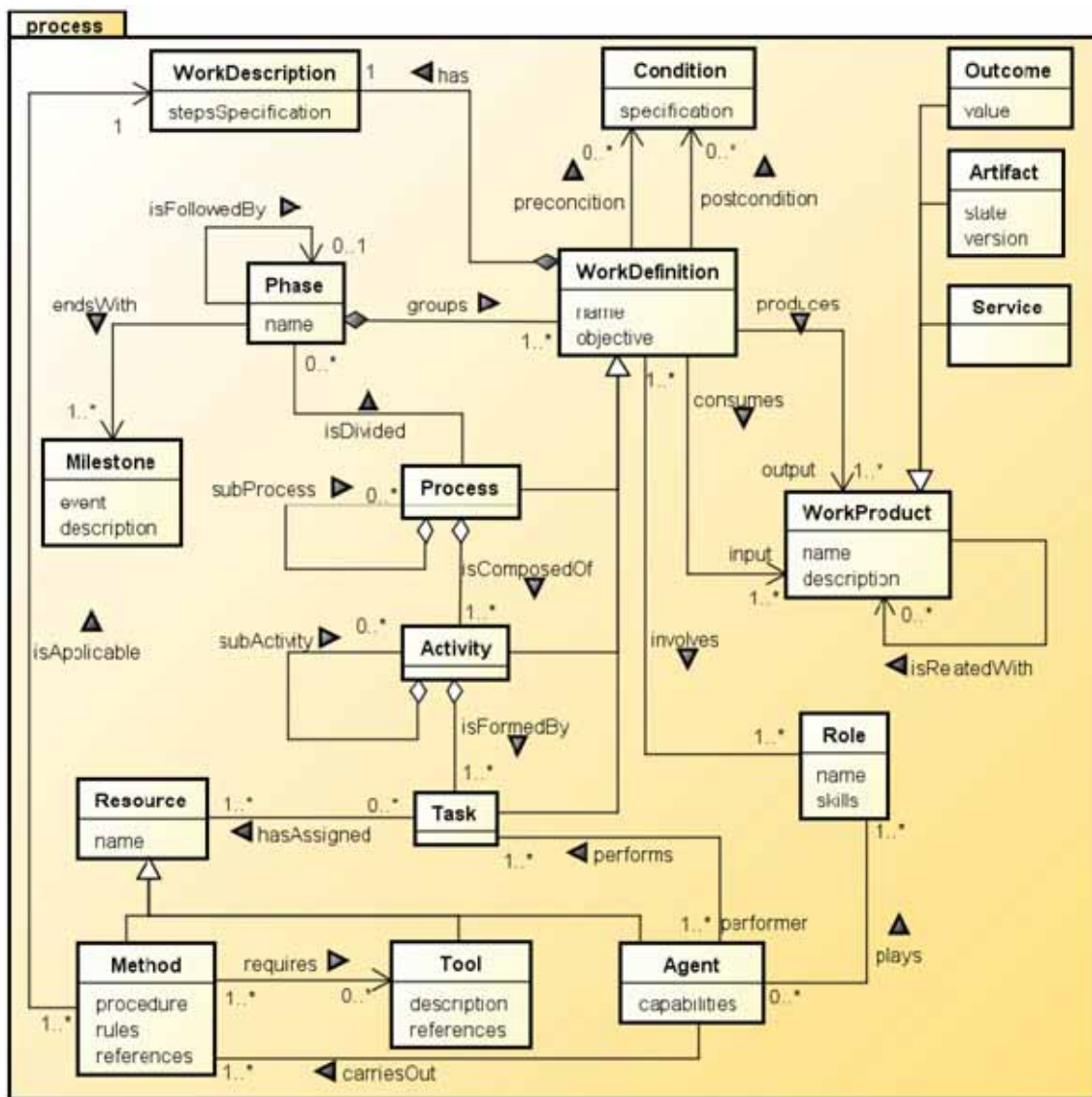


Figura 3-23: Modelo conceptual propuesto para el dominio de proceso.

Los términos fundamentales de la ontología de proceso son *Proceso (Process)*, *Actividad (Activity)* y *Tarea (Task)*, donde un proceso está compuesto por actividades y una actividad está formada por un conjunto de tareas, siendo este último un elemento atómico (es decir, que no se puede descomponer). La semántica que hemos dado a estos tres términos está en conformidad con la dada

en ISO 12207 [ISO/IEC, 2008] (recordar Figura 3-17 en Sección 3.3). También es importante remarcar que una tarea se programa, es decir, se le asignan recursos como por ejemplo un agente, mientras que a las actividades y procesos no (aunque sí deben planificarse en un proyecto). Por otro lado, teniendo en cuenta que una actividad se puede descomponer en actividades de menor nivel de granularidad que no llegan a ser tareas, se ha incluido el concepto *sub-actividad* (*subActivity*). De igual manera también se hace uso del concepto *sub-proceso* (*subProcess*) para hacer referencia a procesos de baja granularidad.

Si bien los términos proceso, actividad y tarea tienen una semántica ligeramente diferente, tal como se ha comentado, cabe mencionar que comparten no obstante ciertas características comunes como por ejemplo:

- ✓ un nombre (*name*),
- ✓ un objetivo (*objective*),
- ✓ una *Descripción de trabajo* (*Work Description*) a realizar –es decir, la especificación de los pasos para alcanzar el objetivo; o en otras palabras, “qué” debe hacerse,
- ✓ involucran *roles* (*Role*),
- ✓ consumen y producen *Productos de trabajo* (*Work Product*), y
- ✓ se realizan bajo ciertas *Condiciones* (*Condition*), llamadas precondiciones (*precondition*) si deben cumplirse al inicio, o postcondiciones (*postcondition*) si deben cumplirse al final del proceso/actividad/tarea.

Teniendo esto en cuenta se decidió hacer uso de un concepto de alto nivel que identifique tanto al concepto proceso, como a actividad y tarea. El concepto abstracto que se decidió utilizar es *Definición de trabajo* (*Work Definition*). Dicho concepto también es utilizado en SPEM.

También cabe destacar la inclusión del concepto *Fase* (*Phase*), el cual representa la agrupación de procesos o actividades fuertemente relacionadas y definidas en cierto orden, y que finalizan con un *Hito* (*Milestone*), es decir, un evento significativo o importante. A modo de ejemplo, el *Proceso Unificado de Rational* (RUP, por sus siglas en inglés) se divide en cuatro fases, a saber: i) Inicio, ii) Elaboración, iii) Construcción y iv) Transición, y sus hitos, respectivamente, podrían ser: i) definición de los objetivos (alcance y requerimientos), ii) definición de la arquitectura, iii) obtención de un producto beta con capacidad operacional, y iv) liberación del producto.

Tabla 3-3: Definiciones de los conceptos del marco conceptual de proceso.

Concepto	Definición
Activity (Actividad)	Es una definición de trabajo (Work Definition) que está formada por un conjunto interrelacionado de sub-actividades y tareas (Task). <u>Nota 1:</u> Una sub-actividad es una actividad de más baja granularidad. <u>Nota 2:</u> En proyectos de ingeniería se planifican actividades en tanto que se programan y ejecutan tareas.
Agent (Agente)	Ente ejecutor asignado a una tarea (Task) en cumplimiento de un rol (Role). <u>Nota 1:</u> Un agente puede ser humano o automático.

Artifact (Artefacto)	Es un producto de trabajo (Work Product) tangible o intangible, versionable, y que puede ser entregado.
Condition (Condición)	Circunstancia que debe cumplirse al iniciar o al finalizar la realización de una definición de trabajo (Work Definition). <u>Nota 1</u> : La condición de inicio se denomina precondition, mientras que la de finalización se denomina postcondición.
Method (Método)	Modo específico y particular de realizar los pasos especificados en la descripción (Work Description) de una definición de trabajo (Work Definition). <u>Nota 1</u> : El modo específico y particular de un método (es decir, <i>cómo</i> deben realizarse los pasos descritos en una definición de trabajo) se representa por un procedimiento y reglas.
Milestone (Hito)	Evento significativo. <u>Nota 1</u> : Un hito representa, por ejemplo, la finalización de una fase.
Outcome (Resultado)	Es un producto de trabajo (Work Product) intangible, almacenable y procesable.
Phase (Fase)	Agrupación de definiciones de trabajo (Work Definition) fuertemente relacionadas definidas en cierto orden. <u>Nota 1</u> : Una fase finaliza con un hito. <u>Nota 2</u> : En una fase, las definiciones de trabajo son procesos y/o actividades.
Process (Proceso)	Es una definición de trabajo (Work Definition) que está compuesta por un conjunto interrelacionado de sub-procesos y actividades (Activity). <u>Nota 1</u> : Un sub-proceso es un proceso de más baja granularidad.
Resource (Recurso)	Activo asignado para realizar una tarea (Task). <u>Nota 1</u> : Un activo es una entidad con valor añadido para una organización.
Role (Rol)	Conjunto de habilidades que debe poseer un agente (Agent) para realizar una definición de trabajo (Work Definition). <u>Nota 1</u> : Habilidades comprende capacidades, competencias y responsabilidades.
Service (Servicio)	Es un producto de trabajo (Work Product) intangible, no almacenable y entregable.
Task (Tarea)	Es una definición de trabajo (Work Definition) atómica, es decir, que no se la puede descomponer. <u>Nota 1</u> : A diferencia de una actividad y de un proceso, a una tarea se le asignan recursos, por ejemplo un agente.
Tool (Herramienta)	Instrumento utilizado para automatizar, parcial o totalmente, un método (Method). <u>Nota 1</u> : Un instrumento puede ser de tipo físico (hardware), computarizado (software) o una combinación de ambos.
Work Definition (Definición de trabajo)	Entidad abstracta que describe al trabajo en base a productos de trabajo (Work Product) consumidos y producidos, condiciones (Conditions) y roles (Role) involucrados. <u>Nota 1</u> : Trabajo representa un proceso, una actividad o una tarea.
Work Description (Descripción de trabajo)	Especificación de los pasos para alcanzar el objetivo de una definición de trabajo (Work Definition). <u>Nota 1</u> : La especificación de los pasos es un conjunto de acciones (sean tanto actividades como tareas) o una función de transformación. Representa <i>qué</i> debería hacerse y no <i>cómo</i> debe realizarse. <u>Nota 2</u> : La especificación de la descripción de una definición de trabajo puede ser formal, semiformal o informal, como por ejemplo, en lenguaje natural.
Work Product (Producto de trabajo)	Producto consumido o producido en una definición de trabajo (Work Definition).

Otro concepto clave en la ontología de proceso es *Producto de trabajo (Work Product)*, ya que cualquier definición de trabajo (proceso/actividad/tarea)

consume, modifica y/o produce productos de trabajo. Con el objetivo de ser más específicos respecto a este concepto, se decidió incluir tres tipos de productos de trabajo, a saber: *Resultado (Outcome)*, *Artefacto (Artifact)* y *Servicio (Service)*.

Teniendo en cuenta las definiciones de la Tabla 3-3, se puede notar que si bien un resultado y un servicio son productos de trabajo intangibles, se diferencian en que el primero es procesable (por ejemplo, datos, información, lecciones aprendidas, medidas, etc.), mientras que el segundo no lo es y la organización que lo brinda no es aquella que lo requiere (por ejemplo, una empresa de desarrollo de software podría contratar servicios de capacitación de personal y de evaluación de la calidad de sus productos). Esta definición de servicio está acorde con la dada en CMMI. Por otro lado, se encuentran los artefactos, que pueden ser tanto tangibles como intangibles, y se caracterizan por ser versionables (ejemplos de artefactos son: el software, los modelos de calidad -como los incluidos en ISO 25010- y los diagramas de casos de uso, entre otros). Cabe acotar que en [PMI, 2009] se indica que un proyecto se lleva a cabo para crear: i) un producto -es decir, un artículo producido, que es cuantificable y que puede ser un elemento terminado o un componente-, ii) la capacidad de realizar un servicio -por ejemplo, una función comercial que brinda apoyo a la producción o distribución-, o iii) un resultado -es decir, una salida de la ejecución de procesos y actividades, tal como un producto o un documento. Al igual que en nuestra ontología, en [PMI, 2009] se hace la distinción de tres tipos de productos de trabajo. Sin embargo la definición del concepto "resultado" dada en [PMI, 2009] es muy amplia y se solapa con la de "producto". Incluso, pensando en una organización que desarrolla software y lleva a cabo un proyecto de M&E para conocer la calidad de alguno de sus productos, el valor de un indicador producido en una actividad de evaluación no encajaría en ninguno de estos tres conceptos, según las definiciones obtenidas del glosario de [PMI, 2009].

Al definir y especificar un proceso, además de ser necesario indicar qué actividades y tareas deben llevarse a cabo, y qué productos de trabajo serán insumidos, modificados y/o producidos, es importante indicar los roles involucrados. Un *Rol (Role)*, indica el conjunto de habilidades (es decir, capacidades, competencias y responsabilidades) que debe poseer un *Agente (Agent)*, quien es el ente ejecutor asignado a una tarea para llevarla a cabo. Notar que un agente puede ser humano, es decir, una persona, o automatizado, como un robot. Cabe acotar que a una tarea se pueden asociar más de un agente, tal es el caso de la programación de pares.

Aparte de asignarse agentes a las diferentes tareas, también deben asignarse otros *Recursos (Resource)*, tales como los *Métodos (Method)* y las *Herramientas (Tool)* que se utilizarán para llevar a cabo la tarea. Los métodos son el modo específico y particular de realizar los pasos especificados en la descripción de trabajo de una tarea. El modo específico y particular de un método (es decir, "cómo" deben realizarse los pasos descritos en una definición de trabajo) se representa por un procedimiento (*procedure*) y reglas (*rules*) asociadas (ver elemento *Method* en Figura 3-23). Por otra parte, las herramientas son instrumentos -ya sean de tipo físico (hardware), computarizado (software) o una combinación de ambos- que automatizan, parcial o totalmente, a un método.

Como conclusión, el marco conceptual de proceso presentado contiene los conceptos necesarios para que el lector pueda comprender mejor la terminología

(del dominio de proceso) utilizada en el resto de este trabajo. Además, teniendo presente que una ontología general puede servir de base o para enriquecer una ontología más específica, la ontología de proceso es utilizada para darle mayor semántica a los conceptos del marco conceptual de M&E llamado C-INCAMI. El marco C-INCAMI se presenta en la Sección 4.2, y su enriquecimiento semántico se explica en la Sección 4.3.

Tabla 3-4: Definiciones de las relaciones del marco conceptual de proceso.

Relación	Definición
carriesOut (realiza)	Un agente realiza, lleva a cabo uno o más métodos.
consumes (consume)	Una definición de trabajo, para lograr su objetivo, consume uno o más productos de trabajo.
endsWith (finalizaCon)	Una fase finaliza con la ocurrencia de uno o más hitos.
groups (agrupa)	Una fase agrupa una o más definiciones de trabajo.
has (posee)	Una definición de trabajo posee una descripción de trabajo en la cual se especifica qué hacer para lograr su objetivo.
hasAssigned (tieneAsignada)	Una tarea tiene asignados recursos para su ejecución.
involves (involucra)	Una definición de trabajo involucra a uno o más roles. A su vez, un rol puede participar en una o más definiciones de trabajo.
isApplicable (esAplicable)	Un método es aplicable a una descripción de una definición de trabajo. A su vez, a una descripción de trabajo se le aplica ninguno, uno o más métodos.
isComposedOf (seComponeDe)	Un proceso se compone de una o más actividades.
isDivided (seDivide)	Un proceso puede dividirse en fases.
isFollowedBy (esSeguidaPor)	Una fase puede suceder a otra fase.
isFormedBy (estaFormadoPor)	Una actividad está formada por una o más tareas.
isRelatedWith (seRelacionaCon)	Un producto de trabajo puede relacionarse con otros productos de trabajo. <u>Nota 1:</u> la relación entre productos de trabajo puede ser de distintos tipos como herencia, composición, agregación, etc.
performs (ejecuta)	Un agente ejecuta una o más tareas que tiene asignada. A su vez, una tarea es ejecutada por uno o más agentes.
plays (ocupa)	Un agente ocupa o juega uno o más roles. A su vez, un rol puede ser jugado por varios agentes.
postcondition (postcondición)	Una definición de trabajo puede tener asociada condiciones que deben cumplirse al final de su realización para ser considerada terminada. <u>Nota 1:</u> El cumplimiento de la condición implica que la evaluación de la misma debe ser verdadera.
precondition (precondición)	Una definición de trabajo puede tener asociada condiciones que deben cumplirse antes de ser iniciada. <u>Nota 1:</u> El cumplimiento de la condición implica que la evaluación de la misma debe ser verdadera.
produces (produce)	Una definición de trabajo produce (crea o modifica) uno o más productos de trabajo.
requires (requiere)	Un método puede requerir el uso de herramientas.
subActivity (subActividad)	Una actividad puede estar formada por otras actividades más específicas, llamadas sub-actividades.
subProcess (subProceso)	Un proceso puede estar formado por otros procesos más específicos, llamados sub-procesos.

Tabla 3-5: Definiciones de los atributos del marco conceptual de proceso.

Concepto	Atributo	Definición
Agent	capabilities (capacidades)	Conjunto de capacidades o habilidades que tiene el agente.
Artifact	state (estado)	Situación en la que se encuentra el artefacto.
	version (version)	Identificador único que indica el nivel de evolución del artefacto.
Condition	specification (especificación)	Especificación no ambigua de las circunstancias que deben cumplirse.
Method	procedure (procedimiento)	Conjunto establecido y ordenado de instrucciones (u operaciones) de un método que indican cómo deben llevarse a cabo los pasos de la descripción de una definición de trabajo.
	rules (reglas)	Conjunto de condiciones, heurísticas, axiomas, etc. asociadas al procedimiento.
	references (referencias)	Cita de recursos bibliográficos, URLs, etc., con información adicional sobre el método.
Milestone	event (evento)	Denominación del suceso previsto que indica un hito.
	description (descripción)	Una declaración textual no ambigua que describe al evento.
Outcome	value (valor)	Resultado numérico o categórico.
Phase	name (nombre)	Denominación que identifica a una fase.
Resource	name (nombre)	Denominación que identifica a un recurso.
Role	name (nombre)	Denominación que identifica a un rol.
	skills (habilidades)	Conjunto de capacidades, competencias y responsabilidades del rol.
Tool	description (descripción)	Una declaración textual no ambigua que describe a la herramienta.
	references (referencias)	Cita de recursos bibliográficos, URLs, etc., con información adicional sobre la herramienta.
Work Definition	name (nombre)	Denominación que identifica a una definición de trabajo.
	objective (objetivo)	Finalidad que se desea alcanzar.
Work Description	stepsSpecification (especificaciónDePasos)	Especificación de pasos a seguir para alcanzar el objetivo de la definición de trabajo. <i>Nota:</i> Los pasos pueden ser acciones genéricas (por ej. para un proceso) o instrucciones concretas (por ej. para una tarea).
Work Product	name (nombre)	Denominación que identifica a un producto de trabajo.
	description (descripción)	Una declaración textual no ambigua que describe al producto de trabajo.

Por último, cabe mencionar que el presente marco conceptual de proceso dispone de los conceptos claves para modelar y especificar procesos desde las diferentes vistas de modelado. Por ejemplo, conceptos tales como *Activity*, *Task* y *Work Product*, junto con las relaciones *consumes* y *produces*, permiten modelar la vista funcional y de comportamiento; la vista organizacional está soportada por conceptos como *Activity*, *Task* y *Role*; y la vista informacional puede modelarse teniendo en cuenta conceptos como *Work Product* y *Artifact*, junto con la relación *is_related_with*.

3.4 Solución Propuesta

Como se analizó en este capítulo, si bien existen muchas estrategias de M&E, no todas cuentan, de una manera integrada, con las tres capacidades fundamentales de una estrategia, a saber: *i)* un *marco conceptual* en el cual se definan los principales términos para el dominio de M&E, *ii)* una *especificación de las vistas del proceso* en la cual se describan las principales actividades que deben realizarse, junto con sus entradas, salidas, roles y otros aspectos, y *iii)* *métodos y herramientas* específicos que permitan llevar a cabo las actividades del proceso. Con el fin de contribuir en este sentido se propone una estrategia integrada de M&E de propósito general llamada GOCAME (*Goal-Oriented Context-Aware Measurement and Evaluation*). Esta estrategia cuenta con las tres capacidades mencionadas ya que posee: *i)* un marco conceptual de M&E, con base ontológica, denominado C-INCAMI (*Context - Information Need, Concept Model, Attribute, Metric and Indicador*) (ver Sección 4.2), *ii)* especificaciones del proceso de M&E teniendo en cuenta diferentes vistas (ver Sección 4.4), y *iii)* una metodología llamada WebQEM (*Web Quality Evaluation Methodology*) junto a la herramienta C-INCAMI_Tool que permiten llevar a cabo las actividades del proceso (ver Sección 4.5).

GOCAME se ha basado en los procesos que subyacen en la metodología WebQEM, los cuales consideran de alguna manera el proceso para evaluadores y el proceso de medición de software de la ISO. Por lo tanto GOCAME se apega a dichos estándares, y además proporciona una visión integrada de ambos, no sólo desde el punto de vista de las actividades que componen los procesos sino también desde el punto de vista de los términos empleados al usar el marco conceptual C-INCAMI.

Otra debilidad encontrada en la mayoría de los enfoques de M&E analizados es que generalmente no tienen en cuenta las vistas del proceso (recordar Sección 3.2). Esto trae aparejado especificaciones del proceso poco detalladas, y por lo tanto, a veces, difíciles de seguir debido a la falta de información. Por ejemplo, en los estándares ISO analizados solo se hace uso de la vista funcional a nivel global, y en el estándar *de facto* CMMI directamente no se presenta ningún modelo de proceso. Teniendo presente la importancia de contar con procesos bien definidos para que los agentes puedan lograr los objetivos perseguidos, esto es, que se pueda identificar claramente las actividades que deben llevarse a cabo, así como la secuencia de las mismas, entre otros aspectos, se propone modelar el proceso de M&E de GOCAME desde diferentes vistas o perspectivas, a saber: funcional, de comportamiento, informacional y organizacional. Tales vistas, o modelos, permiten conocer qué actividades y tareas deben llevarse a cabo y en qué orden, cuáles son los roles involucrados, y qué productos de trabajo son requeridos y cuáles son producidos, así como las relaciones que existen entre los productos de trabajo. Con el objetivo de modelar todos estos aspectos del proceso de M&E de GOCAME se

hará uso del lenguaje de modelado SPEM 2.0 (ver Sección 2.2.4 para consultar los conceptos básicos de SPEM, o el Apéndice A para una revisión más detallada del lenguaje).

Un aspecto a tener en cuenta al momento de modelar procesos es la granularidad del modelo. La granularidad de un modelo de proceso es el nivel de profundidad o detalle al que se va a llegar al momento de describir y modelar el proceso [Pérez-Jiménez, y otros, 2009]. Se pueden distinguir entre modelos de grano fino y modelos de grano grueso [Dowson, 1993; Rolland, 1998]. Los primeros muestran más detalles, mientras que los de grano grueso son abstracciones de más alto nivel. Como se indica en [ISO/IEC, 2001c], el nivel de detalle en la especificación del proceso debería ser tal que la repetibilidad y la reproducibilidad sean aseguradas. Con este fin, se ha modelado el proceso de M&E de GOCAME con una profundidad tal que permite dejar al descubierto cuáles son todas las tareas involucradas junto a sus entradas y salidas, así como secuencias y otros aspectos relevantes. De esta manera se proporciona un nivel de detalle que favorece el aseguramiento de la repetibilidad y reproducibilidad.

También cabe mencionar que en el estándar ISO 12207 [ISO/IEC, 2008] se describe cada proceso con la siguiente información: el título (o nombre, el cual transmite el alcance del proceso en su conjunto), el propósito (que describe el objetivo de realizar el proceso), las salidas (mediante un conjunto de los resultados esperados de la realización eficaz del proceso) y una lista de las actividades y tareas. Por otro lado, en [CMMI Product Team, 2010] se menciona que un proceso definido debe indicar claramente el propósito, las entradas y salidas, criterios de entrada y salida (es decir, pre y pos condiciones), actividades, roles, medidas, y pasos de verificación. Como puede observarse, hay similitudes y diferencias en estos documentos respecto a la información necesaria para describir un proceso. Sin embargo ninguno menciona como requisito contar con modelos del proceso. Recordemos que los modelos facilitan la comunicación y el entendimiento y, además ayudan a identificar fácilmente ciertos aspectos como, por ejemplo, el orden en el cual deben llevarse a cabo las actividades/tareas y puntos de control.

Para hacer frente a esto, proponemos una plantilla de especificación de proceso (Sección 5.4). La misma se basa en el principio W5H (*Who, What, Where, When, Why, How*) y hace uso de la terminología del marco conceptual de proceso (visto en Sección 3.3.1). La plantilla propuesta facilita conocer: el objetivo de la actividad (*why*), la descripción de la actividad y productos de trabajo necesarios (*what*), la dinámica de las actividades (*when*), los roles involucrados en la actividad (*who*), y los métodos, técnicas, guías y herramientas asociados a la actividad (*how*). Toda esta información permite especificar de manera completa un proceso, y sus actividades y tareas. La plantilla hace uso de descripciones textuales junto con modelos gráficos, lo cual aumenta la facilidad de entendimiento y la comunicabilidad. Además, los modelos utilizados integran aspectos de las vistas funcional, de comportamiento y organizacional, permitiendo ver el proceso desde diferentes perspectivas a la vez. Todo esto ayuda a que los interesados dispongan de una visión más práctica, completa, detallada e integrada del proceso.

Además de contar con una especificación del proceso, una estrategia de M&E robusta debe poseer, como ya se dijo, un marco conceptual. Los diferentes enfoques de M&E analizados cuentan, al menos, con algún glosario de los términos utilizados. Sin embargo es importante hacer uso de un vocabulario más ricamente

estructurado, definido por ejemplo mediante una ontología. En vista de esto, GOCAME cuenta con una ontología de M&E (base del marco conceptual C-INCAMI) en la cual se definen los términos, atributos y relaciones concernientes a requerimientos no funcionales, contexto, medición y evaluación (ver Sección 4.2). A su vez, esta ontología ha sido enriquecida con los términos de la ontología de proceso (comentada en Sección 3.3.1) tal como se muestra en la Sección 4.3.

Finalmente, cabe mencionar también que la estrategia GOCAME ha sido utilizada como una estrategia base que permite definir nuevas estrategias de M&E. Por ejemplo, a partir de GOCAME se ha desarrollado SIQinU (*Strategy to Improve Quality in Use*), una estrategia de propósito específico, la cual también cuenta con las tres capacidades consideradas esenciales de una estrategia de M&E. En el Capítulo 5 se describe esta estrategia poniendo especial énfasis en la especificación de las vistas del proceso y en el Capítulo 6 se presenta un caso de estudio utilizando SIQinU.

Capítulo 4: Estrategia GOCAME

En este capítulo se presenta la estrategia de M&E llamada GOCAME. Luego de la introducción se analizan las tres capacidades que la distinguen como una estrategia integrada, poniendo el foco en la capacidad denominada “especificación de las vistas del proceso de M&E”, por ser el tema central de esta tesis.

Por otro lado, respecto a la capacidad marco conceptual de M&E, además de describir a C-INCAMI, se discute cómo los términos de esta base conceptual pueden enriquecerse utilizando la base conceptual de proceso presentada en la Sección 3.3.1.

4.1. Introducción

GOCAME (*Goal-Oriented Context-Aware Measurement and Evaluation*) es una estrategia de M&E desarrollada previamente en nuestro grupo de I+D [Olsina, y otros, 2008b]. Dicha estrategia sigue un enfoque *top-down* u orientado a objetivos. En este tipo de enfoques, en primer lugar se definen los objetivos de la M&E, y a partir de estos se derivan las medidas a obtener. Por el contrario, en un enfoque *bottom-up* se comienza realizando mediciones y luego se trata de relacionar las medidas obtenidas con ciertos objetivos de alto nivel (de proyecto, de negocio, etc.). Según se comenta en [Heidrich, y otros, 2006], un enfoque *top-down* permite el desarrollo de planes de M&E más adecuados, consistentes y completos.

Particularmente, el enfoque *top-down* en GOCAME se sigue de la siguiente manera: en primer lugar se define el propósito de la M&E, a partir de este se instancia un modelo de calidad adecuado, luego se identifican las métricas e indicadores más apropiadas que se utilizarán en la medición y la evaluación, y finalmente se analiza la información obtenida teniendo en cuenta el propósito inicial de la necesidad de información.

Cabe mencionar que GOCAME además de seguir un enfoque orientado a objetivos, también tiene en cuenta el contexto en el cual se produce la M&E con el fin de proporcionar información más precisa para la toma de decisiones. Otro aspecto importante de GOCAME es que es multi-propósito. Por lo tanto, esta estrategia puede utilizarse para evaluar y analizar diferentes propósitos, como “comprender”, “predecir”, “comparar”, etc. Además, GOCAME se utiliza para evaluar entidades de diversas categorías, como por ejemplo sistemas, sistemas en uso, recursos y procesos. Aun más, el foco de la evaluación puede variar dependiendo de la necesidad, como por ejemplo se puede utilizar GOCAME para evaluar la “calidad externa” de un sistema, la “calidad en uso” de un sistema en uso, o el “costo” de un producto o de un recurso.

GOCAME es considerada una estrategia de M&E integrada [Olsina, y otros, 2011; Papa, 2012] ya que, simultáneamente y de manera integrada, cuenta con las tres capacidades descriptas en la Sección 2.1.2, a saber: *un marco conceptual para el dominio de M&E, una especificación de las vistas del proceso de M&E, y, métodos y*

herramientas específicas que permitan llevar a cabo las actividades del proceso de M&E. En las secciones siguientes se describen cada una de estas capacidades en el contexto de GOCAME.

4.2. Marco conceptual de M&E

El marco conceptual de M&E utilizado en GOCAME se denomina C-INCAMI (*Contextual-Information Need, Concept model, Attribute, Metric and Indicator*) [Molina, 2012; Olsina, y otros, 2008b]. Dicho marco identifica los conceptos claves del dominio de M&E. Esto permite que exista un vocabulario común entre los miembros de la organización al gestionar proyectos de M&E, definir y especificar los requerimientos no funcionales, especificar el contexto en el cual se efectúa la medición, diseñar y seleccionar un conjunto concreto de métricas, interpretar los valores medidos usando indicadores y obtener conclusiones en base a las medidas y valores de indicadores. Asimismo, C-INCAMI permite identificar claramente los datos y metadatos que son fundamentales para obtener resultados repetibles, reproducibles y consistentes para la toma eficaz de decisiones.

Tabla 4-1: Ontología para Medición y Evaluación: glosario abreviado de términos (ver [Olsina, y otros, 2008b] para mas detalles).

Término	Descripción
<i>Términos de Requerimientos No Funcionales</i>	
Attribute (Atributo)	Una propiedad física o abstracta medible de una categoría de entidad
Calculable Concept (Concepto Calculable)	Característica de alto nivel de abstracción.
Concept Model (Modelo de Concepto)	Conjunto de subconceptos y relaciones entre ellos, que provee la base para especificar los conceptos para la evaluación.
Entity (Entidad)	Un objeto concreto que pertenece a una categoría de entidad.
Entity Category (Categoría de Entidad)	Categoría del objeto que es caracterizado por la medición de sus atributos.
Information Need (Necesidad de Información)	Conocimiento necesario para la gestión de objetivos, metas, riesgos y problemas.
<i>Términos de Medición</i>	
Base Measure (Medida Base)	Medida que no depende de otras medidas.
Calculation Procedure (Procedimiento de Cálculo)	Conjunto de instrucciones ordenadas y establecidas de una métrica indirecta o indicador que indica cómo deberían llevarse a cabo los pasos descritos en una medición indirecta o evaluación.
Derived Measure (Medida Derivada)	Medida calculada a partir de otras medidas.
Direct Measurement (Medición Directa)	Medición que produce una medida base.
Direct Metric (Métrica Directa)	Métrica de un atributo que no depende de otras métricas.
Indirect Measurement (Medición Indirecta)	Medición que produce una medida derivada.
Indirect Metric (Métrica Indirecta)	Métrica de un atributo que depende de otras métricas.

Measure (Medida)	Número o categoría asignado a un atributo de una entidad, producto de una medición.
Measurement (Medición)	Tarea que usa la definición de una métrica para producir un valor de una medida.
Measurement Procedure (Procedimiento de Medición)	Conjunto de instrucciones ordenadas y establecidas de una métrica directa que indica cómo deberían llevarse a cabo los pasos descritos en una medición directa.
Metric (Métrica)	El procedimiento de medición o de cálculo definido y la escala.
Scale (Escala)	Conjunto de valores con propiedades definidas.
Unit (Unidad)	Cantidad particular definida y adoptada por convención, con la cual otras cantidades de un mismo tipo son comparadas con el fin de expresar su magnitud relativa a la cantidad.
<i>Términos de Evaluación</i>	
Decision Criterion (Criterio de Decisión)	Umbral, objetivos, o patrones usados para determinar la necesidad de actuar o investigar, o para describir el nivel de confianza en un resultado dado.
Derived Evaluation (Evaluación Derivada)	Evaluación que produce un valor de indicador evaluando un concepto calculable.
Derived Indicator (Indicador Derivado)	Un indicador que es derivado de otros indicadores para evaluar o estimar un concepto calculable.
Elementary Evaluation (Evaluación Elemental)	Evaluación que produce un valor de indicador evaluando un atributo.
Elementary Indicator (Indicador Elemental)	Un indicador que no depende de otros indicadores para evaluar o estimar un concepto calculable.
Elementary Model (Modelo Elemental)	Algoritmo o función con criterios de decisión asociados que modela un indicador elemental.
Evaluation (Evaluación)	Actividad que utiliza la definición de un indicador para producir un valor de indicador.
Global Model (Modelo Global)	Algoritmo o función con criterios de decisión asociados que modela un indicador global.
Indicator (Indicador)	El método de cálculo y la escala definidos, además del modelo y los criterios de decisión para obtener una estimación o evaluación de un concepto calculable con respecto a la necesidad de información.
Indicator Value (Valor de Indicador)	El número o categoría asignado a un concepto calculable.

Cabe mencionar que los diferentes conceptos de C-INCAMI están agrupados en seis componentes, a saber:

1. *Proyecto de M&E* (M&E Project),
2. *Requerimientos No Funcionales* (Requirements),
3. *Contexto* (Context),
4. *Medición* (Measurement),
5. *Evaluación* (Evaluation), y
6. *Análisis y Recomendación* (Analysis).

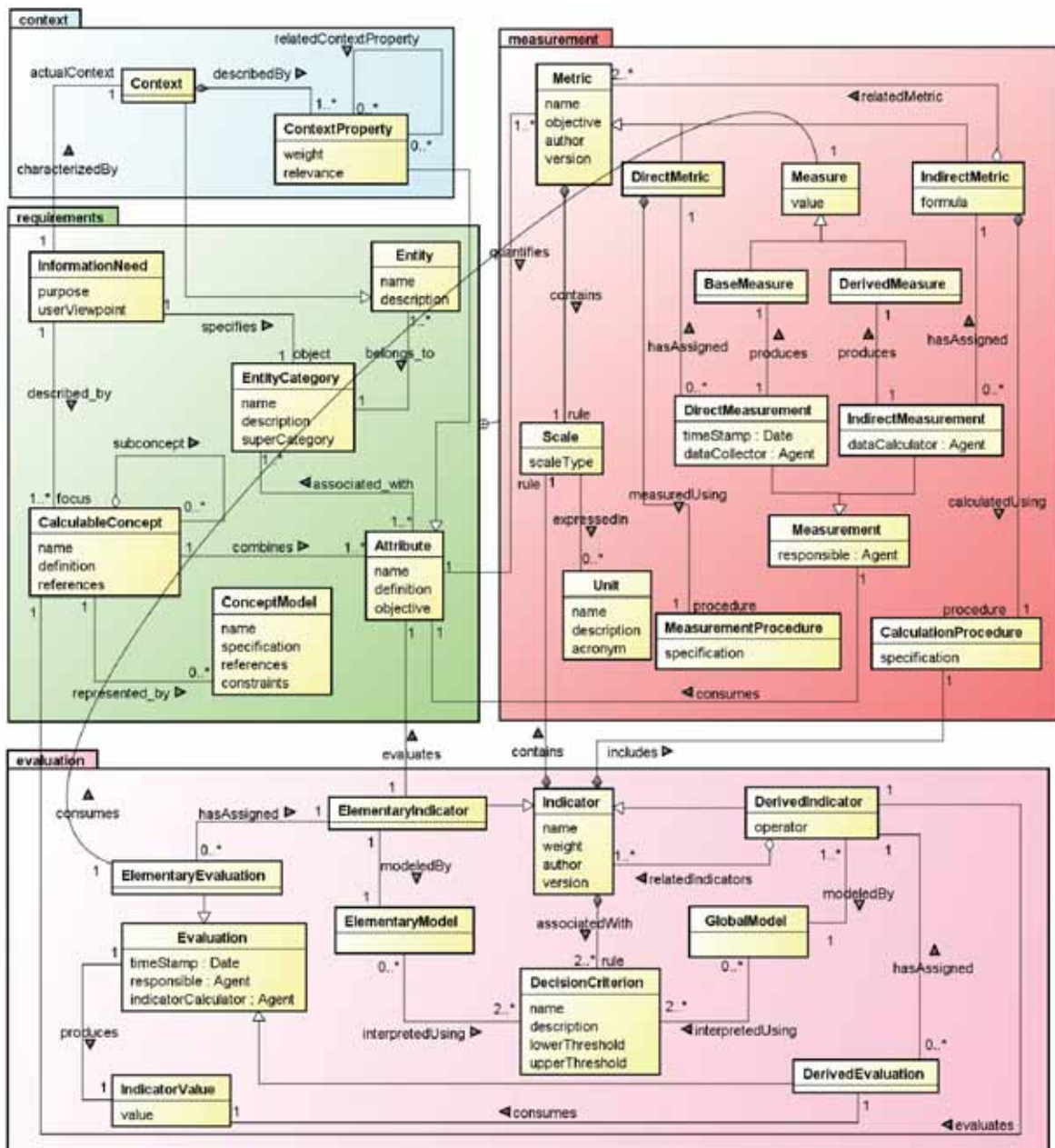


Figura 4-1: Principales términos, atributos y relaciones del marco C-INCAMI.

Dichos componentes no están aislados unos de otros, sino que se relacionan mediante sus conceptos según se muestra en la Figura 4-1 (Nota: con el fin de no sobrecargar la figura, solo se presentan los principales términos de los componentes *Requerimientos No Funcionales*, *Contexto*, *Medición* y *Evaluación*).

La elección de C-INCAMI como marco conceptual para GOCAME se debe a que los términos utilizados, a diferencia de otros marcos conceptuales, se encuentran formalizados en una ontología de M&E [Olsina, y otros, 2004] (en la Tabla 4-1 se definen algunos de los principales términos de esta ontología). Esto permite resolver problemas de terminología entre diferentes fuentes de información y brinda un vocabulario más ricamente estructurado. Por otro lado, cabe notar que algunas bases conceptuales analizadas son ambiguas. Tal es el caso de ISO, que confunde los términos métrica y medida. Algo similar ocurre en CMMI y FMESP. Aun más, en FMESP no existe el concepto de evaluación, aunque sí el de indicador.

Otra clara diferencia entre C-INCAMI y el resto de los marcos conceptuales de M&E revisados, es la inclusión de términos relacionados a especificación de contexto. Para un análisis más completo de las ventajas de C-INCAMI frente a otros marcos conceptuales ver [Olsina, y otros, 2008b; Papa, 2012].

A continuación se comentan brevemente cada uno de los componentes del marco C-INCAMI y se resaltan en cursiva los términos propios de la ontología de M&E (por cuestiones de espacio algunos de estos términos no aparecen en la Figura 4-1).

4.2.1 Componente de Proyecto de M&E

Este componente contiene los conceptos necesarios para permitir al evaluador especificar de forma concisa los datos administrativos del proyecto de M&E (*ME Project*), como puede ser la fecha de inicio, datos del director del proyecto, una descripción, etc.

Un proyecto de M&E se compone de 3 sub-proyectos, a saber: un proyecto de requerimientos (*Requirements Project*), el cual se relaciona con las actividades de especificación de requerimientos no funcionales; un proyecto de medición (*Measurement Project*), que permite llevar a cabo las actividades de especificación de métricas y ejecución de las mediciones; y un proyecto de evaluación (*Evaluation Project*), para especificar indicadores y conocer el nivel de satisfacción alcanzado por los requerimientos no funcionales. La división de un proyecto de M&E en diferentes sub-proyectos responde a necesidades de manejo de la complejidad así como la consideración de una clara división de responsabilidades dentro de un proyecto, permitiendo así delegar los distintos proyectos a diferentes personas.

4.2.2 Componente de Requerimientos No Funcionales

Este componente contiene los conceptos necesarios para definir y especificar los requerimientos no funcionales para un proyecto de M&E, el cual surge a partir de una necesidad de información (*Information Need*) en algún nivel de la organización.

La M&E siempre se realiza sobre una entidad (*Entity*) concreta que pertenece a una categoría de entidad (*Entity Category*). Algunos ejemplos de entidades son "Facebook", "Samsung Galaxy S4 " y "ArgoUML", y pertenecen respectivamente a las siguientes categorías de entidad: "Red Social", "Dispositivo Móvil" y "Programa en Java".

Es importante mencionar que una (categoría de) entidad cuenta con propiedades mensurables llamadas atributos (*Attribute*) o requerimientos elementales. Estos pueden agruparse en categorías o conceptos calculables (*Calculable Concept*), es decir, características o factores de alto nivel de abstracción. A su vez un concepto calculable también puede formar parte de otro concepto calculable de mayor nivel de abstracción. Por ejemplo, la Calidad en Uso es un concepto calculable que agrupa a los subconceptos Eficacia, Eficiencia y Satisfacción, entre otros [ISO/IEC, 2011]. Las relaciones y restricciones entre atributos, subconceptos y conceptos calculables se reflejan mediante un modelo de concepto (*Concept Model*) y se representan comúnmente en la forma de un árbol de requerimientos (*Requirements Tree*), el cual permite realizar la evaluación o estimación como un todo.

4.2.3 Componente de Contexto

Este componente posee aquellos términos necesarios para especificar el contexto o ambiente en el cual se realiza el proyecto de M&E.

Si bien no siempre es necesario especificar información del contexto, hay situaciones en las que conocer el contexto es muy útil, ya que esto permitirá efectuar análisis y comparaciones más robustas con los resultados provenientes de otros proyectos de la organización o de otras organizaciones [Molina, y otros, 2008].

La forma de especificar el contexto es mediante la inclusión de atributos o propiedades (*Context Property*) que sean relevantes al contexto (*Context*) que caracteriza la necesidad de información. Ejemplos de propiedades del contexto pueden ser el ciclo de vida utilizado para el desarrollo de software, el grado de experiencia de los desarrolladores y la cantidad de desarrolladores, por mencionar algunas.

El componente también permite especificar entidades contextuales (*Contextual Entity*), es decir, entidades que, por su definición, sólo pueden ser aplicadas e interpretadas correctamente en determinados contextos. Por ejemplo, la métrica “Cantidad de clases implementadas” es aplicable en un contexto en el cual el software tiene una arquitectura orientada a objetos.

4.2.4 Componente de Medición

Este componente permite especificar, para cada uno de los atributos del árbol de requerimientos, una métrica (*Metric*) que lo cuantifique, ya sea cuantitativamente o cualitativamente.

Una métrica define cómo se llevará a cabo la medición (*Measurement*) así como la escala (*Scale*) en que se expresarán los valores (*Measures*) obtenidos. Se distinguen dos tipos de medición: directa o indirecta. Una medición directa (*Direct Measurement*) es aquella que no depende de la medición previa de otros atributos, a diferencia de la medición indirecta (*Indirect Measurement*). A su vez, en una medición directa se emplea una métrica directa (*Direct Metric*) y el valor obtenido se denomina medida base (*Base Measure*). Por el contrario, en una medición indirecta se hace uso de una métrica indirecta (*Indirect Metric*), la cual se relaciona con otras métricas, y el valor obtenido se llama medida derivada (*Derived Measure*), justamente por ser un valor que se deriva o proviene de valores de otras métricas.

Una métrica directa cuenta con un procedimiento de medición (*Measurement Procedure*), el cual especifica los pasos a seguir para obtener el valor o medida del atributo correspondiente. Por otro lado, una métrica indirecta hace uso de un procedimiento de cálculo (*Calculation Procedure*), el cual indica la forma en que se calculará el valor del atributo, partiendo de valores obtenidos para otros atributos. Adicionalmente, una métrica indirecta especifica una fórmula o función (*Function*) que permitirá efectuar el cálculo.

En cuanto a la escala (*Scale*), esta puede ser: numérica (*Numerical Scale*) o categórica (*Categorical Scale*). En el caso de las escalas numéricas estas poseen una unidad (*Unit*) y su representación puede ser continua o discreta. Por último, en lo

que respecta al tipo de la escala (*Scale Type*) este puede ser: nominal, ordinal, intervalo, proporción o absoluto.

4.2.5 Componente de Evaluación

Identifica el conjunto de conceptos necesarios para definir indicadores (*Indicator*), los cuales permiten obtener un valor (*Indicator Value*) que refleja el nivel de satisfacción alcanzado por cada elemento (atributos y conceptos calculables) del árbol de requerimientos. Este valor se consigue siguiendo un procedimiento de cálculo (*Calculation Procedure*). Un indicador también especifica la escala (*Scale*) utilizada para representar el valor.

Existen dos clases de indicadores: elementales y derivados. Un indicador elemental (*Elementary Indicator*) evalúa a un atributo, y cuenta con un modelo elemental (*Elementary Model*) que define cómo se realizará el mapeo, del valor medido del atributo, al valor (*Indicator Value*) que indique su nivel de satisfacción. En cambio, un indicador derivado (*Derived Indicator*) se asocia a un concepto calculable, y hace uso de un modelo global (*Global Model*) para definir cómo se interpretarán los valores (*Indicator Value*) correspondientes a los conceptos y/o atributos de menor nivel en el árbol de requerimientos. De esta forma se observa que un indicador elemental no depende de otros indicadores para evaluar o estimar un atributo, mientras que un indicador derivado hace uso de otros indicadores para evaluar o estimar un concepto calculable.

Un modelo global (*Global Model*) o elemental (*Elementary Model*) es un algoritmo o función con criterios de decisión (*Decision Criterion*) asociados que modelan un indicador derivado o elemental, respectivamente. Los criterios de decisión son patrones usados para determinar la necesidad de una acción o investigación posterior, o para describir el nivel de confianza o preferencia de un resultado dado. Una diferencia fundamental entre un indicador y una métrica es justamente que el indicador tiene asociado criterios de decisión (o niveles de aceptabilidad) que permiten interpretar el valor del indicador, mientras que una métrica no.

4.2.6 Componente de Análisis y Recomendación

Este componente permite a los evaluadores realizar las actividades de análisis y comparación de los valores o niveles de satisfacción alcanzados por cada elemento del árbol de requerimientos y, asimismo, la justificación de los resultados obtenidos. Para facilitar la interpretación de los datos, su seguimiento y registración se utilizan herramientas y mecanismos de análisis y documentación. A partir de las metas establecidas se generarán las conclusiones y recomendaciones sobre el ente evaluado.

4.3. Relación entre el Marco Conceptual C-INCAMI y el Marco Conceptual de Proceso

Al analizar el marco conceptual C-INCAMI se puede notar que muchos de los términos del dominio de M&E están relacionados a muchos de los términos del dominio de proceso. Por ejemplo, una medición es una tarea, una métrica es un método, una medida es un resultado, etc. Sin embargo, estas relaciones no están explícitamente modeladas en la Figura 4-1. Por lo tanto, con el objetivo de hacer explícitas estas relaciones, se ha vinculado C-INCAMI con el componente de

proceso comentado en la Sección 3.3.1. De esta manera, ahora, los términos de C-INCAMI, particularmente los involucrados en los componentes de Medición y de Evaluación, han sido enriquecidos semánticamente con los términos de proceso [Becker, y otros, 2013; Becker, y otros, 2014].

Para hacer explícita la relación entre ambos dominios se optó por hacer uso de estereotipos de UML [OMG-UML, 2010]. Un estereotipo es un mecanismo de extensibilidad provisto por UML, el cual permite refinar el significado de un elemento del modelo (por ejemplo una clase o una relación). Teniendo esto presente, aquí se utilizarán los estereotipos con el fin de dotar de mayor semántica a los términos de C-INCAMI.

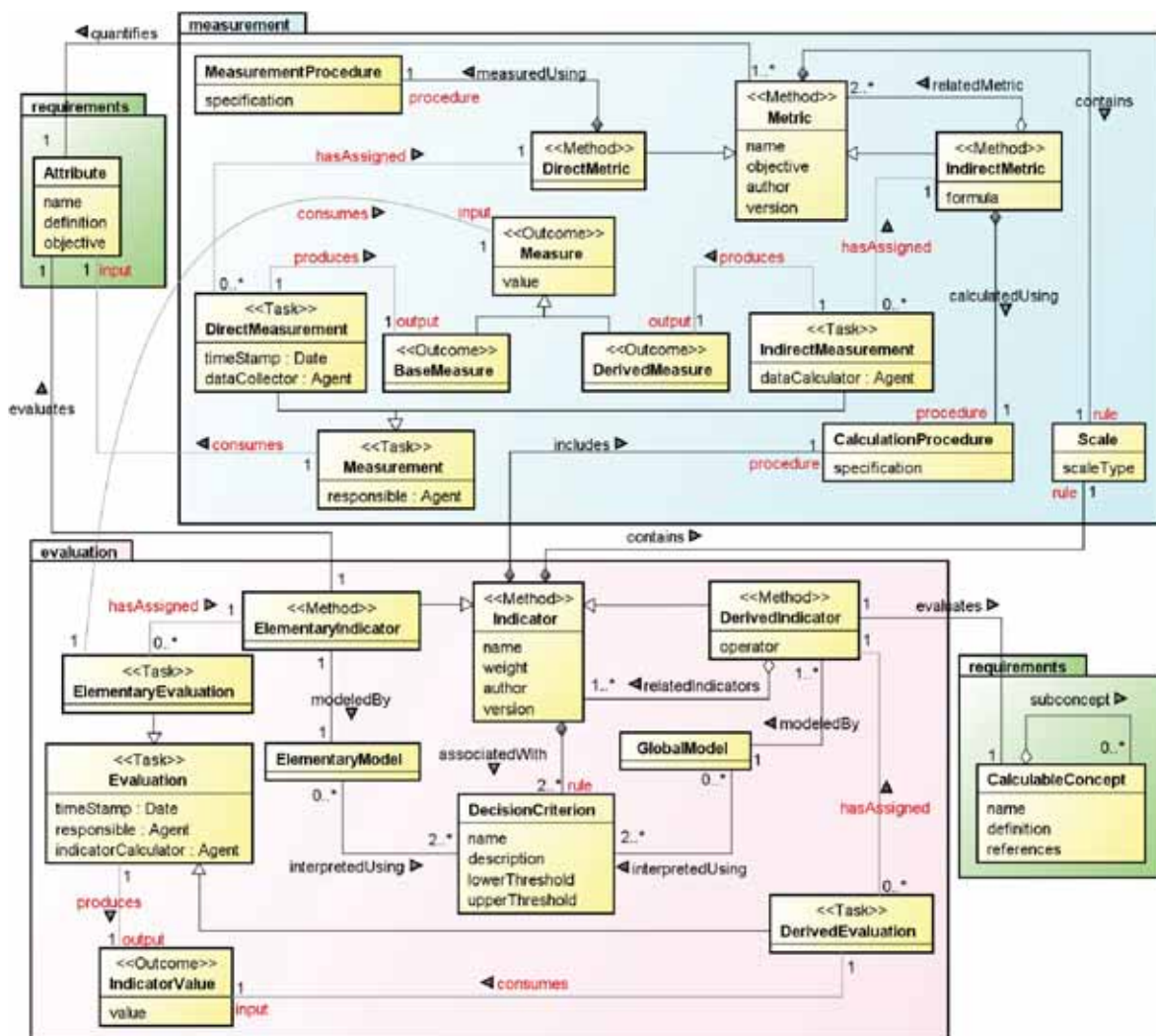


Figura 4-2: Principales términos, atributos y relaciones de los componentes de Medición y de Evaluación enriquecidos con estereotipos del componente de Proceso.

En la Figura 4-2 se pueden observar los componentes de Medición y de Evaluación (recordar Figura 4-1) enriquecidos, mediante estereotipos, con los términos de proceso (recordar Figura 3-23). Por ejemplo, el término *Metric*

(Métrica) tiene el estereotipo <<Method>> (Método)⁶. Mirando la Tabla 4-1, una métrica se define como “el procedimiento de medición o de cálculo definido y la escala”. Ahora, con el estereotipo <<Method>> una métrica también incluye la semántica de un método, el cual se define como el “modo específico y particular de realizar los pasos especificados en la descripción (Work Description) de una definición de trabajo (Work Definition)” (ver Tabla 3-3). Así, una métrica específica cómo deben realizarse los pasos descritos de una tarea de medición. Además, tal como se muestra en la Figura 3-23, un método (Method) tiene como atributos un procedimiento (*procedure*) y reglas (*rules*). De igual manera, una métrica tiene asociados un procedimiento de medición (*Measurement Procedure*) o de cálculo (*Calculation Procedure*) y una escala (*Scale*) como regla.

También es importante remarcar que algunas relaciones entre los términos de M&E han sido renombradas y otras agregadas para mantener consistencia con el componente de proceso. Por ejemplo, la relación *consumes* entre *Elementary Evaluation* y *Measure* antes no existía. Pero al considerar que una tarea (*Task*) consume (*consumes*) un producto de trabajo (*Work Product*) como entrada (*input*) y que ahora *Elementary Evaluation* tiene la semántica de *Task* y *Measure* de *Outcome* (es decir un tipo de *Work Product*) se agregó la relación *consumes* junto con el final de asociación *input*. En la Figura 4-2 se resaltan las relaciones tomadas a partir del componente de proceso usando color rojo en los nombres de las mismas.

Por último es necesario mencionar que un trabajo de características similares es [Dal Moro, y otros, 2008]. En el trabajo citado los autores presentan una ontología de Calidad del Software (luego refinada en [Barcellos, y otros, 2010]) y la relacionan a la Ontología de Proceso de Software o SPO (presentada en [Falbo, y otros, 2005] y refinada en [Guizzardi, y otros, 2008] y luego en [Bringunte, y otros, 2011]). Sin embargo, es importante recordar que, tal como se comentó en la Sección 3.3, SPO utiliza una terminología que difiere de la usada en reconocidos estándares del área de proceso como SPEM, CMMI e ISO 12207. Por otro lado, respecto a la ontología de Calidad de Software observamos una ambigüedad en el uso del término *Measure*, ya que a veces éste se refiere al valor producido por una medición, mientras que otras veces se refiere al instrumento, o procedimiento, para obtener tal valor. En la ontología que subyace en C-INCAMI se hace una clara distinción entre estos dos conceptos mediante los términos *Measure* y *Metric*. Además, los autores utilizan el término *measurable element* para referirse a las propiedades de una entidad, siendo *Attribute* el término ampliamente utilizado en la literatura de M&E [ISO/IEC, 2007b; ISO/IEC, 2005; ISO/IEC, 2001c; García, y otros, 2006b]. Finalmente, C-INCAMI se diferencia en que permite especificar información de contexto [Molina, y otros, 2008].

4.4. Especificación de las Vistas del Proceso de M&E

En un proceso de M&E, como en cualquier otro proceso ingenieril, es importante

⁶ En el contexto del proceso de M&E, una métrica es un método. Sin embargo, en el contexto de revisión y catalogación en repositorios de métricas, una métrica es un artefacto. Por lo tanto dependiendo del contexto, un término podría enriquecerse de diferentes maneras. Lo mismo sucede en otros ámbitos, por ejemplo, en un proceso de fabricación de pc's, una pc es un artefacto, mientras que en el contexto de enseñanza on-line, una pc es un recurso. El contexto que se tiene presente en este trabajo es el de un proceso de M&E.

conocer qué actividades y tareas deben llevarse a cabo para lograr un objetivo. También es necesario identificar claramente los productos de trabajo que sirven de entrada y aquellos que son producidos, definir los roles involucrados, y otros aspectos. Teniendo presente lo mencionado, GOCAME cuenta con la especificación de las vistas del proceso de M&E [Becker, y otros, 2012; Becker, y otros, 2010a; Becker, y otros, 2010b; Becker, y otros, 2010c; Becker, y otros, 2009; Becker, y otros, 2008].

Especificar el proceso desde diferentes vistas o perspectivas provee una visión completa, detallada e integrada del proceso de M&E. Particularmente, la vista *funcional* permite identificar qué actividades y tareas se deben llevar a cabo, así como sus entradas y salidas; la de *comportamiento* indica en qué orden deben ejecutarse las actividades y tareas; la vista *organizacional* muestra los roles involucrados; y la *informativa* da a conocer la estructura de los productos de trabajo consumidos y producidos. Esto permite tener una guía clara y repetible de las actividades y tareas que deben realizarse para definir los requerimientos no funcionales, establecer el contexto, diseñar e implementar la medición, la evaluación, y el análisis y la recomendación, con el fin de soportar un proceso de toma de decisiones.

En el modelado de procesos, un detalle no menos importante a tener presente es el estilo que se utiliza para nombrar las actividades y tareas, así como los productos de trabajo. Por ejemplo en el estándar ISO 15939 (proceso de Medición) los nombres de las actividades/tareas comienzan con verbos en infinitivo (por ejemplo: *Ejecutar* el proceso de medición), mientras que en ISO 14598 (proceso de Evaluación) se emplea la sustantivación de los verbos para nombrar las actividades (por ejemplo: *Ejecución* de la evaluación). Tal inconsistencia en los estilos de los nombres confunde y, a veces, dificulta la identificación de actividades y tareas respecto de productos de trabajo. Para evitar este problema, en GOCAME se ha optado por nombrar las actividades/tareas comenzando con un verbo en infinitivo.

A continuación se especifica el proceso de M&E de GOCAME desde sus diferentes vistas. La notación utilizada para especificar el proceso es SPEM junto con Diagramas de Actividad y de Clase de UML 2.0 (ver la justificación de esta elección en la Sección 2.2.4).

4.4.1 Proceso de GOCAME: Vista Funcional y de Comportamiento

El proceso de M&E de GOCAME integra las siguientes seis actividades principales:

- (A1) *Definir los Requerimientos No Funcionales*
- (A2) *Diseñar la Medición*
- (A3) *Implementar la Medición*
- (A4) *Diseñar la Evaluación*
- (A5) *Implementar la Evaluación*
- (A6) *Analizar y Recomendar*

En la Figura 4-3 se modela la vista funcional y de comportamiento del proceso, la cual incluye las actividades mencionadas, así como secuencias, paralelismos y las principales entradas y salidas. Cabe resaltar que las actividades principales de GOCAME armonizan con las actividades propuestas en los estándares ISO del proceso de medición de software [ISO/IEC, 2007b] y del

proceso para evaluadores [ISO/IEC, 2001b] (recordar Sección 3.2.1), pero con la diferencia que en GOCAME estas actividades se encuentran integradas en un único proceso y además se utiliza un único vocabulario, el provisto por el marco conceptual C-INCAMI.

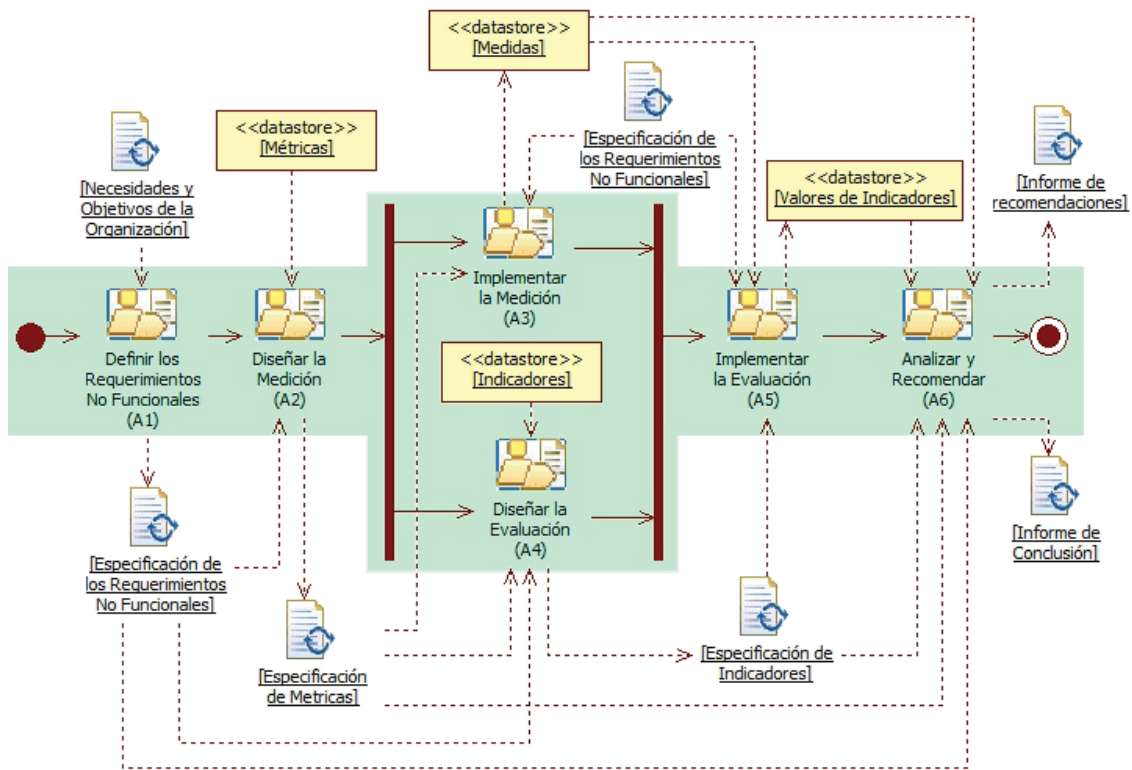


Figura 4-3: Vista funcional y de comportamiento del proceso de M&E de GOCAME.

Vale acotar que en la Figura 4-3, como en varias de las figuras siguientes que contienen modelos de proceso, se utilizan objetos con el estereotipo <<datastore>>. Estos hacen referencia a repositorios o catálogos que permiten almacenar y recuperar definiciones de los distintos elementos de información que son reutilizables en el proceso (por ejemplo atributos, modelos de concepto, métricas, etc). Al describir varias de las actividades del proceso se asume que muchos de estos elementos ya se encuentran debidamente especificados, consensuados por expertos y almacenados en estos catálogos, por lo cual, a menudo, no se describen actividades de creación de los mismos. Por ejemplo, en el proceso de M&E aquí descrito se supone que las métricas que se utilizarán ya han sido previamente diseñadas (creadas), validadas y almacenadas en un repositorio, como parte de un proceso diferente al de M&E en sí [Baffini, y otros, 2006].

Siguiendo la Figura 4-3, se puede notar que, como ya se ha indicado, GOCAME sigue un enfoque *top-down*. A partir de las *Necesidades y Objetivos de la Organización* la primera actividad consiste en *Definir los Requerimientos No Funcionales (A1)*, lo cual incluye especificar el propósito de la evaluación, el foco de la evaluación y atributos, entre otros aspectos. Definir en primer lugar estas cuestiones proporcionará una guía para realizar las actividades siguientes, como por ejemplo elegir métricas adecuadas a la necesidad de información o problema en cuestión.

Una vez producido el artefacto *Especificación de los Requerimientos No*

Funcionales, es posible *Diseñar la Medición (A2)*, es decir, identificar las métricas que se utilizarán para cuantificar los atributos. Las métricas y sus metadatos asociados son recuperadas desde un repositorio. Una vez seleccionadas todas las métricas y creado el artefacto *Especificación de Métricas*, se puede realizar, en forma paralela (o de manera secuencial, sin importar el orden), el diseño de la evaluación o la implementación de la medición.

Implementar la Medición (A3) consiste en utilizar las métricas escogidas anteriormente y así obtener un valor para cada uno de los atributos de las entidades analizadas. Los valores medidos, junto a sus metadatos, son almacenados en un repositorio para su posterior análisis. La actividad *Diseñar la Evaluación (A4)* implica especificar indicadores y criterios de aceptación que nos permitan obtener el grado o nivel de satisfacción que proporcionan cada uno de los requerimientos no funcionales (tanto atributos como conceptos calculables).

Una vez especificados todos los indicadores en el artefacto *Especificación de Indicadores* y realizadas las mediciones correspondientes, se está en condiciones de *Implementar la Evaluación (A5)*, es decir, utilizar los indicadores definidos anteriormente con el fin de conocer el grado de satisfacción brindado por cada uno de los requerimientos. Finalmente, se realiza la actividad *Analizar y Recomendar (A6)*. Teniendo en cuenta la necesidad de información o problema, esta actividad permite realizar un análisis de las medidas y valores de indicadores obtenidos y recomendar acciones a seguir. El análisis y las recomendaciones se plasman en los *Informes de Conclusión y de Recomendaciones*, respectivamente.

A continuación, cada una de estas actividades principales se describen con mayor nivel de detalle. Se utilizará un caso de estudio [Covella, y otros, 2006] durante todo el capítulo, con el fin de ejemplificar las actividades y tareas realizadas en el proceso de M&E aquí modelado, como así también los productos de trabajo producidos o necesarios.

4.4.1.1 Definir los Requerimientos No Funcionales (A1)

Una vez definido el proyecto de M&E, la primera actividad que se debe realizar es Definir los Requerimientos No Funcionales. Básicamente, tiene como entrada las *Necesidades y Objetivos de la Organización* que dan origen al proyecto de M&E. Ejemplos de necesidades y objetivos de la organización son: incrementar la productividad del grupo de desarrollo, conocer la calidad de un sitio web, reducir costos de un producto software, entre otros.

Particularmente, en la Facultad de Ingeniería de la UNLPam, durante los años 2003, 2004 y 2005, se empleó una aplicación Web de e-learning llamada Qplus Campus Virtual destinada a ser utilizada por alumnos ingresantes como soporte para un curso de matemáticas, y se quería “conocer cuál era la percepción de la calidad en uso que tenían los usuarios finales (alumnos), con el fin de saber si ésta superaba cierto umbral de satisfacción aceptable” [Covella, y otros, 2006]. Con el fin de poder dar respuesta a esta necesidad se realizó un proyecto de M&E.

Como salida de la actividad Definir los Requerimientos No Funcionales se produce el artefacto *Especificación de los Requerimientos No Funcionales* (ver sección 4.4.3 para conocer la estructura del mismo). La actividad Definir los Requerimientos No Funcionales agrupa las siguientes tres sub-actividades (ver Figura 4-4), las cuales luego se comentan:

- ✓ Establecer la Necesidad de Información
- ✓ Especificar el Contexto
- ✓ Establecer el Árbol de Requerimientos

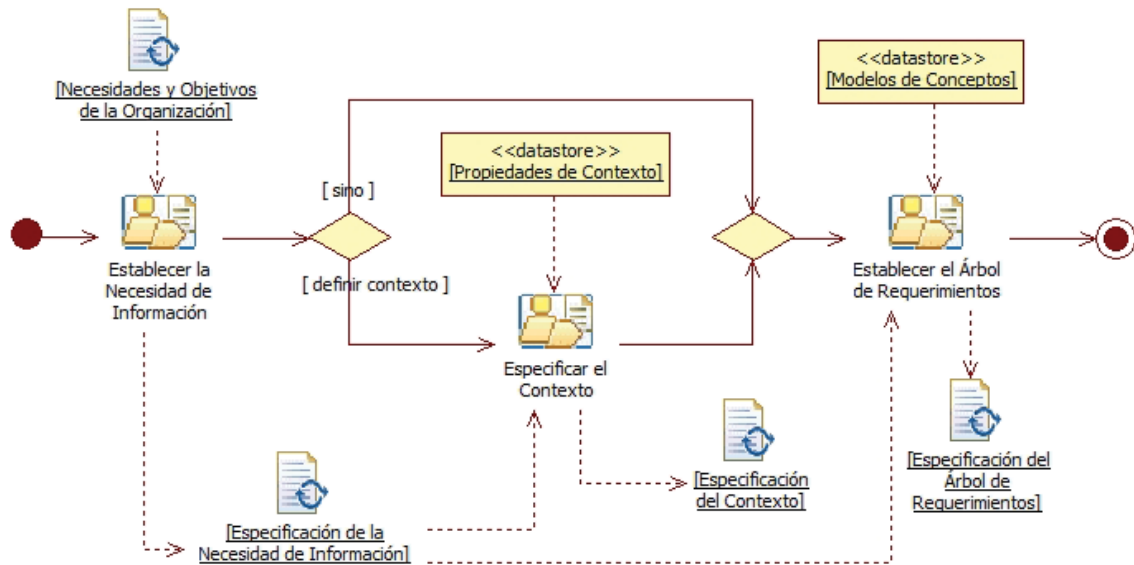


Figura 4-4: Flujo de actividades para Definir los Requerimientos No Funcionales.

Establecer la Necesidad de Información

La actividad *Establecer la Necesidad de Información* implica las siguientes tareas:

- ✓ Definir el propósito
- ✓ Definir el punto de vista del usuario
- ✓ Establecer el objeto
- ✓ Establecer entidad
- ✓ Identificar el foco

Siguiendo el diagrama de la Figura 4-5, podemos ver que en primer lugar, se debe *Definir el propósito*. Esta actividad implica definir cuál es el propósito de la M&E que se llevará a cabo. Ejemplos de posibles propósitos (junto con sinónimos y su significado) son:

- *Comprender / caracterizar / evaluar*: para conocer el valor resultante de una característica foco.
- *Predecir / estimar*: para determinar de antemano el valor resultante de una característica foco.
- *Mejorar*: para incrementar el valor resultante de una característica foco.
- *Reducir*: para disminuir el valor resultante de una característica foco.
- *Controlar / certificar / comparar*: para contrastar el valor resultante de una característica foco contra un valor planificado, estándar o de referencia.

Siguiendo el caso de estudio comentado, y teniendo en cuenta que la necesidad planteada es “conocer la percepción de calidad en uso que tienen los alumnos ingresantes a la Facultad de Ingeniería de la UNLPam al usar una aplicación de e-learning denominada ‘Qplus Campus Virtual’ que da soporte al curso nivelatorio denominado *Introducción a la Matemática*”, el propósito de realizar el proceso de M&E es: “comprender”.

Luego, sabiendo el propósito, se debe *Definir el punto de vista de usuario* que se tendrá en cuenta al realizar la evaluación. Algunos ejemplos de puntos de vista de usuario son: usuario casual, usuario experto, desarrollador, por citar algunos. En nuestro caso, se fija como punto de vista el de un “*alumno ingresante*”, dado que la aplicación fue utilizada sólo por alumnos ingresantes y son ellos los destinatarios de la aplicación. Una vez definidos el propósito y el punto de vista, se procede a *Establecer el objeto*. Esta tarea involucra especificar cuál será la categoría de entidad (por ejemplo: un producto, un recurso, etc.) a ser caracterizada a través de la medición de sus atributos. De esta forma, en nuestro caso de estudio, la categoría de entidad es “*aplicación de e-learning*”, y corresponde a la super-categoría *sistema en uso*⁷.

Seguidamente, se puede llevar a cabo la tarea *Establecer entidad*. Para ello, se debe tener en cuenta la categoría de entidad previamente establecida. Como se observa en la Figura 4-5, esta tarea es opcional en esta etapa del proceso, ya que podría ocurrir que no se sepa aún cuál será la entidad concreta a evaluar, por ejemplo, al establecer un programa de M&E genérico. Asimismo, también se da la posibilidad de establecer varias entidades, por ejemplo, para los casos en los cuales se quiera comparar dos o más entidades. Es importante mencionar que establecer la entidad en este momento del proceso puede favorecer la realización de la próxima tarea, denominada *Identificar el foco* de la evaluación. Volviendo al caso de estudio planteado, sólo tenemos una entidad para evaluar, la cual es el sistema “*Qplus Campus Virtual*”.

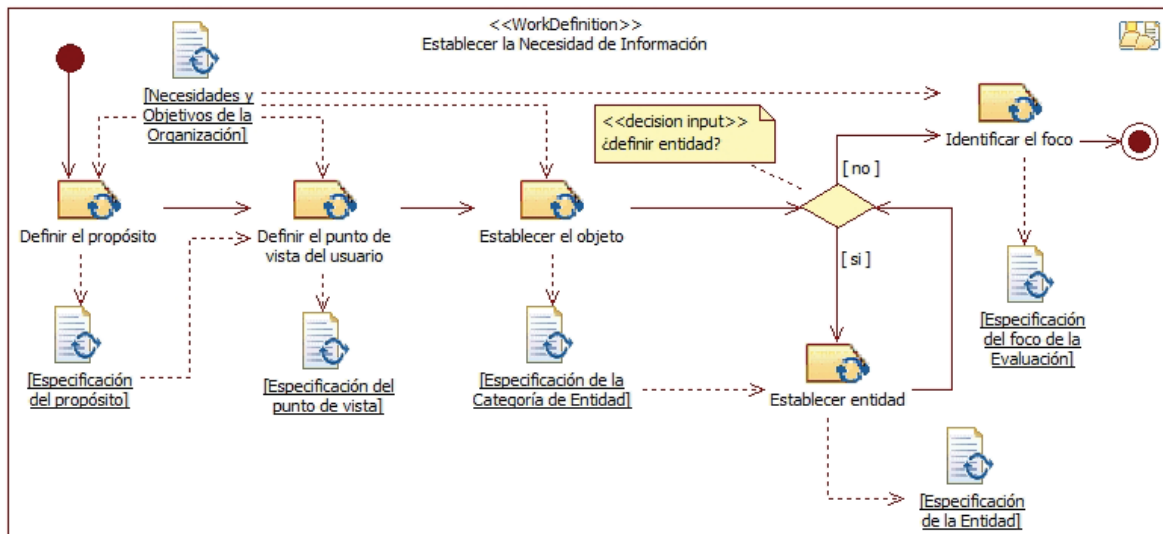


Figura 4-5: Vista funcional y de comportamiento de la actividad *Establecer la Necesidad de Información*.

Identificar el foco consiste en determinar cuál será el concepto calculable (recordar este término en la sección 4.2.2.) de más alto nivel que se desea evaluar (por ejemplo calidad externa, productividad, etc.). En esta tarea, también se puede llegar a identificar cuáles son los subconceptos asociados al foco, tal como pueden ser usabilidad, eficiencia, portabilidad, etc. En nuestro caso de estudio analizado, el

⁷ El término *sistema en uso* se refiere al producto o sistema siendo utilizado en un contexto real por usuarios reales.

foco a considerar en la evaluación es la “*Calidad en uso*”, la cual está relacionada a aspectos de “*Eficacia*”, “*Productividad*”⁸, y “*Satisfacción*” [ISO/IEC, 2001a].

De esta manera se completa la sub-actividad *Establecer la Necesidad de Información*, teniendo como resultado el documento *Especificación de la Necesidad de Información*, el cual contiene la especificación concisa del propósito, punto de vista de usuario, foco para la necesidad de información acordada y la categoría de la entidad que será objeto de estudio.

Especificar el Contexto

Retomando la Figura 4-4, notamos que una vez establecida la necesidad de información se debe optar por *Especificar el Contexto* en el cual se realizarán las mediciones. Contar con información del contexto permitirá realizar análisis más robustos y objetivos de los resultados, principalmente si estos provienen de distintos proyectos y organizaciones [Molina, y otros, 2008]. La importancia del contexto radica en que no es siempre factible obtener conclusiones válidas al evaluar un mismo producto o sistema en contextos distintos o bien en un contexto que no está claramente especificado. Especificar el contexto permitirá que se pueda realizar recomendaciones más ajustadas a la realidad, ya que no sólo se contará con datos de las mediciones y evaluaciones, sino también con información relevante al contexto, la cual influye en la interpretación de los resultados y las decisiones.

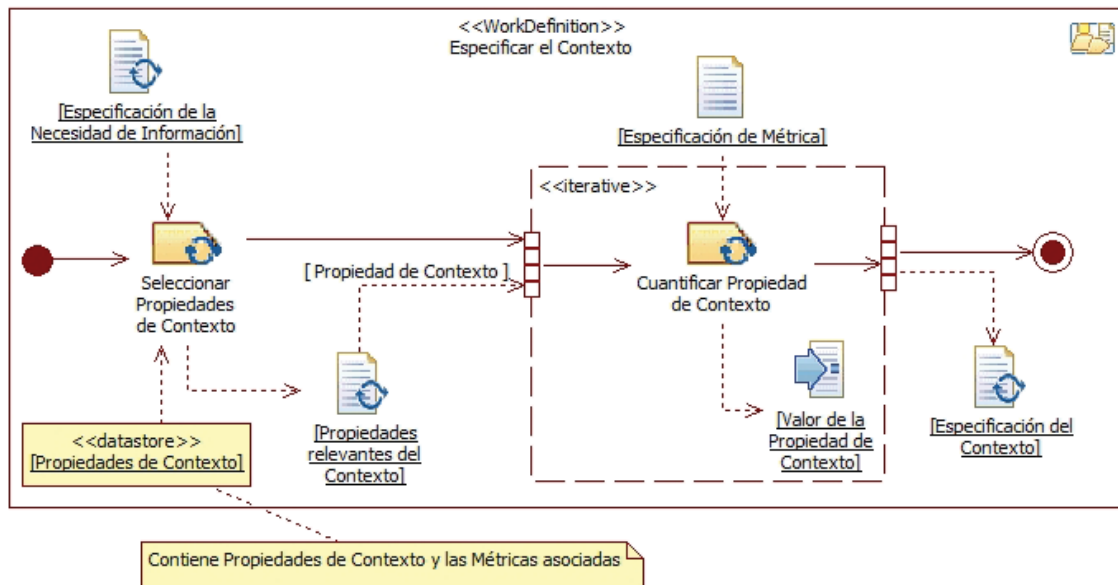


Figura 4-6: Tareas para *Especificar el Contexto*.

La especificación del contexto se efectúa registrando las propiedades del contexto y sus valores correspondientes. En la Figura 4-6 se puede observar que Especificar el Contexto involucra dos tareas:

- ✓ *Seleccionar Propiedades de Contexto*
- ✓ *Cuantificar Propiedad de Contexto*

La primera tarea, implica identificar aquellas propiedades que son relevantes

⁸ Llamada “Eficiencia” en [ISO/IEC, 2011].

al contexto del proyecto en cuestión, tal como pueden ser el tipo de ciclo de vida de desarrollo utilizado, el grado de experiencia de los desarrolladores, el sistema operativo, el procesador utilizado, entre otras. Cabe aclarar que la relevancia de una propiedad depende de la necesidad de información especificada, por lo cual se puede observar en la Figura 4-6 que la *Especificación de la Necesidad de Información* sirve como entrada a esta tarea. Una vez definidas las propiedades que caracterizan el contexto, se debe proceder a cuantificar cada una de ellas haciendo uso de alguna métrica. Como se muestra en la Figura 4-6, esto se realiza iterativamente mediante la tarea *Cuantificar Propiedad de Contexto* en base a la métrica asociada (definida en el artefacto *Especificación de Métrica*). Al finalizar estas dos tareas, obtendremos como salida una *Especificación del Contexto*, la cual incluye las propiedades más relevantes y sus valores.

Cabe aclarar que, para la realización de la sub-actividad *Especificar el Contexto* sería deseable, por cuestión de reutilización, contar con un repositorio de propiedades de contexto con sus métricas asociadas [Rivera, y otros, 2007]. Este repositorio no sólo ayuda a agilizar el proceso debido a que las propiedades y métricas están apropiadamente almacenadas, sino que brinda además un mayor grado de objetividad, ya que permite utilizar únicamente propiedades (y sus métricas asociadas) que hayan sido debidamente consensuadas por expertos en un proceso de discusión. Por este último motivo, en el presente trabajo no se considera la posibilidad de crear propiedades de contexto, o asociar nuevas métricas a estas, sino que sólo se limita a la selección de las mismas; además de ser actividades que escapan al proceso de M&E en sí.

En el caso de estudio utilizado como ejemplo, se consideró si era necesario *Especificar el Contexto*. Teniendo en cuenta que en [ISO/IEC, 2001a] la Calidad en Uso se define como “la capacidad de un producto de software de facilitar a usuarios específicos alcanzar metas específicas con eficacia, productividad, seguridad y satisfacción *en un contexto específico de uso*” (cursivas agregadas), entonces se puede concluir que es de importancia especificar el contexto. Las propiedades relevantes respecto de la necesidad de información especificada son:

- a) el “*ancho de banda de la red local*”,
- b) el “*navegador web utilizado*” y
- c) la “*experiencia del usuario utilizando aplicaciones Web*”.

Decimos que estas propiedades son relevantes debido a que:

- a) dependiendo del ancho de banda de la red local, la productividad del usuario al interactuar con la aplicación puede variar,
- b) la utilización de determinados navegadores puede afectar la eficacia y productividad dependiendo de los controles y *scripts* que permita ejecutar,
- c) la experiencia del usuario afecta a la forma en que interactúa y percibe a la aplicación Web.

Una vez seleccionadas las propiedades que caracterizan el contexto, se debe proceder a cuantificar cada una de ellas utilizando la métrica asociada a la propiedad, según se especifique en el catálogo de *Propiedades de Contexto*. Al finalizar esta actividad, en el caso analizado se obtuvo que, para la propiedad a) el valor es “*15.3 Kbits/seg*”⁹, para la b) es “*Internet Explorer*” y, finalmente, para c) es

⁹ Tener presente que este valor fue obtenido en el año 2006.

“pocas veces”.

Establecer el Árbol de Requerimientos

Recordar que, en la tarea *Identificar el foco*, se debía especificar cuál sería el concepto de mayor nivel de abstracción que se quería evaluar (en nuestro ejemplo es: *Calidad en Uso*). Además, se podía llegar a definir una lista de subconceptos relacionados a este (en nuestro caso: *Eficacia*, *Productividad* y *Satisfacción*). Si bien en esa tarea se definieron conceptos y subconceptos, no obstante no se especificó la forma en la cual se relacionan los mismos, es decir, no se especificó un modelo de concepto (recordar este término en la sección 4.2.2). Además, tampoco se especificaron los atributos que se van a medir. Para tal fin se ha diseñado la sub-actividad *Establecer el Árbol de Requerimientos*, la cual comprende (ver Figura 4-7):

- ✓ *Seleccionar un Modelo de Concepto*
- ✓ *Definir el Árbol de Requerimientos*

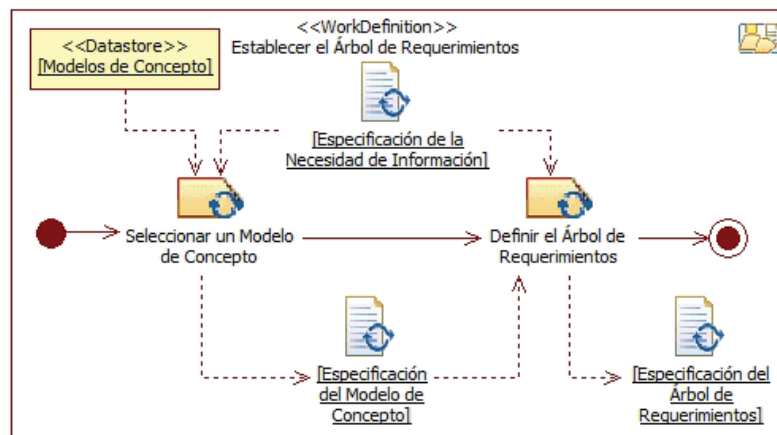


Figura 4-7: Tareas que intervienen para *Establecer el Árbol de Requerimientos*.

La tarea *Seleccionar un Modelo de Concepto* consiste en escoger, desde un repositorio, un modelo de concepto que relacione, de alguna forma, los conceptos y subconceptos asociados al foco establecido en la *Especificación de la Necesidad de Información*. A modo de ejemplo, en la Figura 4-8 se muestra el modelo de concepto definido por la ISO 9126-1 para la característica *Calidad en Uso*¹⁰.

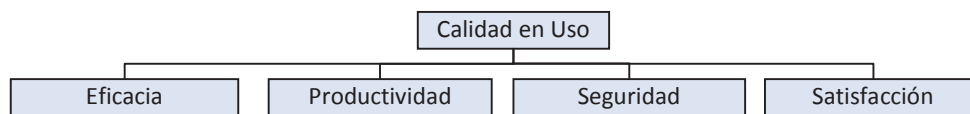


Figura 4-8: Modelo de Calidad en Uso definido en ISO 9126-1.

1. Calidad en Uso
 - 1.1. Eficacia
 - 1.1.1. *Complejidad de Tareas*
 - 1.1.2. *Eficacia de Tareas*
 - 1.2. Productividad
 - 1.2.1. *Eficiencia en relación a Complejidad de Tareas*
 - 1.2.2. *Eficiencia en relación a Eficacia*
 - 1.3. Satisfacción

Figura 4-9: Árbol de Requerimientos del caso de estudio analizado [Covella, y otros, 2006].

¹⁰ Actualmente el estándar ISO 9126 fue reemplazado por ISO 25010, redefiniendo sus modelos.

Luego de haber seleccionado un modelo de concepto, se debe *Definir el Árbol de Requerimientos*. Esta tarea consiste en agregar o quitar conceptos, subconceptos, atributos y relaciones entre ellos, teniendo presente las restricciones del modelo de concepto, además de la categoría de entidad definida en el documento Especificación de la Necesidad de Información. Una vez finalizada esta tarea, obtendremos una *Especificación del Árbol de Requerimientos*, la cual tendrá como nodo raíz el concepto foco, como hojas los atributos y como nodos intermedios los subconceptos (la Figura 4-9 muestra el Árbol de Requerimientos).

Especificación de Requerimientos No Funcionales

Especificación de la Necesidad de Información

Propósito: *comprender*

Punto de vista de usuario: alumno ingresante

Categoría de entidad (Objeto) bajo análisis: Aplicación de e-learning / Sistema en uso

Entidad(es): *Qplus Campus Virtual*

Foco de la Evaluación: Calidad en Uso

Conceptos (Características) Relacionados: Eficacia, Productividad y Satisfacción

Especificación del Contexto

Propiedades relevantes del Contexto	Valor
Ancho de banda de la red local	15.3 Kbits/seg
Navegador web utilizado	Internet Explorer
Experiencia del usuario utilizando aplicaciones Web	baja (pocas veces)

Especificación del Árbol de Requerimientos

Modelo base utilizado: Modelo de Calidad en Uso

Tipo: propio – estándar – mixto

Referencias: ISO 9126-1: Software Engineering Product Quality - Part 1: Quality Model (2001)

Especificación: ...

Árbol de Requerimientos:

1. **Calidad en Uso**
 - 1.1. **Eficacia**
 - 1.1.1. Completitud de Tareas
 - 1.1.2. Eficacia de Tareas
 - 1.2. **Productividad**
 - 1.2.1. Eficiencia en relación a Completitud de Tareas
 - 1.2.2. Eficiencia en relación a Eficacia
 - 1.3. Satisfacción

Atributos

Completitud de Tareas

Definición: proporción de las tareas completadas por los usuarios respecto de las tareas propuestas.

Objetivo: permite conocer en qué medida las tareas propuestas han sido completadas por los usuarios. A diferencia del atributo Eficacia de Tarea, aquí se toman en cuenta sólo las tareas que han sido completadas correctamente en su totalidad.

Eficacia de Tareas

Definición: correctitud en la realización total o parcial de las tareas, para todos los usuarios.

Objetivo: permite conocer en qué medida el producto puede ser utilizado eficazmente por los usuarios en la realización de tareas típicas. Se considera que una tarea completada parcialmente otorga igualmente un cierto grado de eficacia, en la medida en que las metas parciales se cumplen correctamente.

Eficiencia en Relación a Completitud de Tareas

⋮

Eficiencia en relación a Eficacia

⋮

Figura 4-10: Ejemplo de una *Especificación de Requerimientos No Funcionales*.

Volviendo una vez más al caso analizado, dado que el foco de la evaluación identificado es *Calidad en Uso* y los conceptos asociados a este son *Eficacia*, *Productividad* y *Satisfacción*, al momento de *Seleccionar un Modelo* se puede elegir el modelo de *Calidad en Uso* propuesto por la ISO ya que, como se aprecia en la Figura 4-8, incluye y relaciona estos conceptos, además de *Seguridad*.

Ahora se debe proceder a *Definir el Árbol de Requerimientos*. En nuestro caso no es necesario agregar más subconceptos, pero sí quitar el subconcepto Seguridad, debido a que se consideró poco relevante para el producto y el contexto en que se realizó el estudio. Por otro lado, se asociaron atributos a los conceptos ya establecidos, dando lugar a la formación del *Árbol de Requerimientos* ilustrado en la Figura 4-9.

Cabe aclarar que *Satisfacción*, una característica para evaluar la percepción subjetiva de los usuarios, en un principio se la identificó como un concepto calculable. Pero al momento de *Definir el Árbol de Requerimientos* se optó por considerarla como un atributo, ya que, en este caso de estudio, se decidió emplear directamente un cuestionario de satisfacción de índole general (ver [Covella, y otros, 2006]). No obstante, de ser necesario, es posible identificar atributos para evaluar aspectos específicos de Satisfacción a un nivel de granularidad más fino.

Al finalizar todas las actividades modeladas en la Figura 4-4, se obtiene como salida un documento estructurado con la especificación de los requerimientos no funcionales, conteniendo un modelo de calidad instanciado (o árbol de requerimientos), el propósito, el punto de vista de usuario, la especificación del contexto, la categoría de la entidad a evaluar (y opcionalmente, la entidad) y el foco de la evaluación. A modo de ejemplo, en la Figura 4-10 se presenta el artefacto *Especificación de Requerimientos No Funcionales* obtenido como resultado de haber ejecutado la actividad A1.

4.4.1.2 Diseñar la Medición (A2)

Una vez que se han definido todos los requerimientos no funcionales, ya es posible identificar, de una manera objetiva, cuáles serán las métricas que se utilizarán para obtener las medidas que cuantifican a los atributos de la entidad (o entidades). Básicamente, *Diseñar la Medición* consiste en (ver Figura 4-11):

- ✓ *Establecer Entidad*
- ✓ *Asignar Métricas*

Recordar que esta actividad no consiste en diseñar las métricas en sí, sino sólo en identificar y asignar las métricas más apropiadas para cuantificar cada uno de los atributos del árbol de requerimientos, seleccionándolas desde un catálogo de métricas (identificado con el estereotipo *datastore* en la Figura 4-11). Estas métricas deberían haber sido diseñadas y acordadas previamente por expertos, por ejemplo, haciendo uso de un sistema colaborativo de revisión de métricas, como el descrito en [Baffini, y otros, 2006], donde se define el proceso deseable para la creación y revisión de métricas por medio del acuerdo o consenso de un grupo de expertos y/o interesados de la organización.

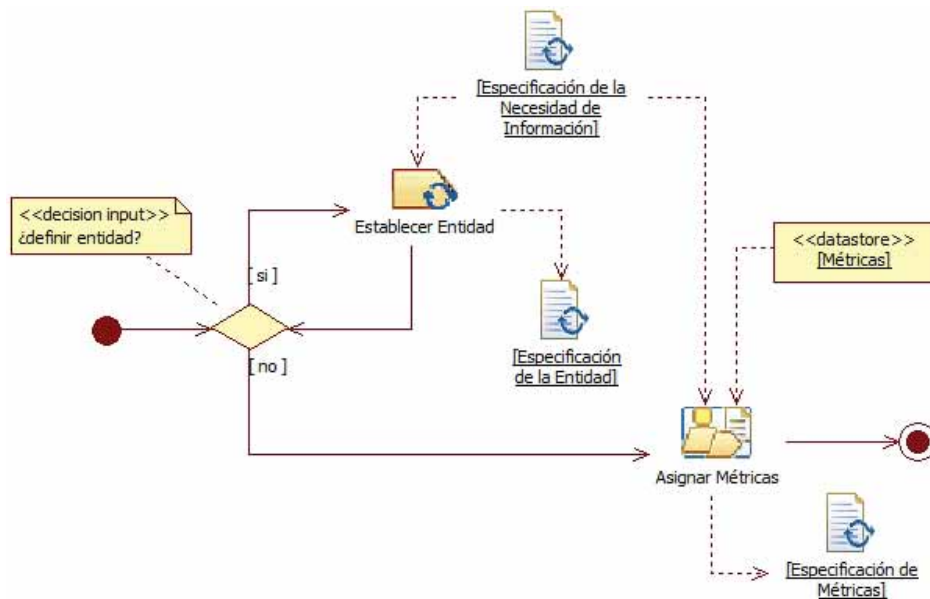


Figura 4-11: Diseñar la Medición.

Establecer Entidad

Debido a que la medición se realiza sobre atributos de una entidad concreta, al *Diseñar la Medición* se debe permitir identificar las entidades concretas que se van a evaluar. Como se observa en la Figura 4-11, la identificación de las entidades se realiza mediante la tarea *Establecer Entidad*. La entidad escogida debe pertenecer a la categoría de entidad especificada en el documento *Especificación de la Necesidad de Información*. Notar también que esta tarea es opcional, ya que podría ocurrir que, al momento de Diseñar la Medición, aun no se tengan claramente definidas cuál o cuáles son las entidades concretas, por lo cual, esta tarea puede ser postergada a lo sumo hasta la actividad *Implementar la Medición* (ver Sección 4.4.1.3).

Asignar Métricas

La siguiente actividad es *Asignar Métricas* a los atributos. Recordar que los atributos fueron identificados en la actividad A1, más precisamente durante la tarea *Definir el Árbol de Requerimientos* (sección 4.3). Para *Asignar Métricas* a los atributos, se deben llevar a cabo las siguientes tareas:

- ✓ *Seleccionar una Métrica*
- ✓ *Identificar las Métricas relacionadas*
- ✓ *Seleccionar una Herramienta*

Como se observa en la Figura 4-12, estas tareas deben realizarse de forma iterativa por cada uno de los atributos especificados en el árbol de requerimientos. La primera tarea a realizar es *Identificar una Métrica* que cuantifique el atributo. Como ya se mencionó, las métricas no son diseñadas en este proceso, sino que son obtenidas desde un catálogo de métricas. El uso de catálogos, favorece la reutilización de métricas, especificadas de manera consistente, en diferentes proyectos de M&E. Es importante aclarar que en los casos en que no exista una métrica adecuada, es necesario comenzar un proceso colaborativo de revisión de

métricas, mediante el cual se ponga en discusión una nueva métrica para ser agregada al catálogo y así poder seleccionarla luego.

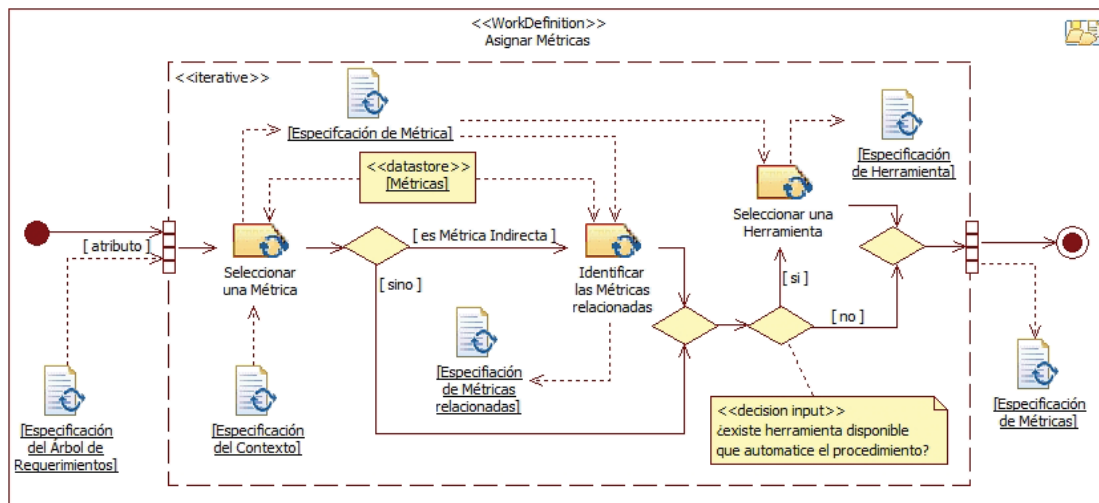


Figura 4-12: Tareas de la actividad *Asignar Métricas*.

Al momento de seleccionar las métricas, también es necesario tener en cuenta la Especificación del Contexto, ya que pueden existir métricas no adecuadas o aplicables en determinados contextos. Por ejemplo, si se desea medir el tamaño del código fuente de un módulo escrito en un lenguaje orientado a objetos (como JAVA), una métrica aplicable pudiera ser “cantidad de clases codificadas”. Sin embargo, si el lenguaje no es orientado a objetos (por ejemplo C), la métrica mencionada no sería adecuada, pudiendo ser conveniente utilizar una métrica como “líneas de código”.

Aparte de las métricas utilizadas para cuantificar los atributos presentes en el árbol de requerimientos, también se deben considerar otras. Se trata de aquellas métricas relacionadas a las métricas indirectas. Las métricas indirectas se basan en el cálculo de una fórmula cuyos valores de entrada se obtienen a partir de otras métricas (que cuantifican otros atributos de la entidad, que muchas veces no pertenecen al árbol de requerimientos), por ejemplo, en el caso del cálculo de promedios o proporciones. Por tal razón, luego de seleccionar una métrica, en el caso que esta sea indirecta (recordar componente de Medición de la Figura 4-1), es necesario *Identificar las Métricas relacionadas*, lo cual incluye también identificar los atributos que son cuantificados por estas métricas relacionadas.

Se debe tener presente que estos atributos relacionados no serán incluidos en el árbol de requerimientos diseñado, pero deberán ser tenidos en cuenta al momento de llevar a cabo la medición ya que son el punto de partida obligado para el cálculo de las métricas indirectas que cuantifiquen atributos del árbol de requerimientos. Como ejemplo, supongamos que tenemos el atributo *Enlaces Rotos* y a este atributo le asignamos la métrica indirecta *Cantidad de Enlaces Rotos* que está definida como la suma de las medidas obtenidas por las métricas directas *Cantidad de Enlaces Internos Rotos* y *Cantidad de Enlaces Externos Rotos*. En este caso, debemos recuperar también las últimas dos métricas mencionadas, así como también los atributos que ellas cuantifican, que son *Enlaces Internos Rotos* y *Enlaces Externos Rotos* respectivamente.

Dependiendo del tipo de métrica, directa o indirecta, ésta posee un procedimiento de medición o de cálculo. Muchas veces el procedimiento puede estar automatizado por una o varias herramientas. En los casos en que exista alguna herramienta disponible, que automatice el procedimiento, se realiza la tarea *Seleccionar una Herramienta*. La elección de la herramienta podría realizarse al momento de *Implementar la Medición*, por ser una tarea íntimamente relacionada con el acto de la medición en sí. Sin embargo, podría ocurrir que quien diseñe la medición sea un agente diferente a quien ejecute la medición, por lo que se podría asumir que aquel que se encarga del diseño de la medición sería más apto para saber cuál es la herramienta que debe emplearse acorde al objetivo establecido (para conocer las incumbencias de los agentes según el rol que jueguen, ver la Sección 4.4.2).

Para ejemplificar la actividad *Diseñar la Medición* tengamos en cuenta los atributos especificados en el árbol de requerimientos de la Figura 4-9, a saber:

- *Compleitud de Tareas*
- *Eficacia de Tareas*
- *Eficiencia en relación a Compleitud de Tareas*
- *Eficiencia en relación a Eficacia*
- *Satisfacción*

Para el atributo *Compleitud de Tareas*, en el caso de estudio citado se identificó una métrica denominada *Promedio de la Proporción de Tareas Completadas sobre Tareas Propuestas para Todos los Usuarios (P_CT)*, cuya fórmula asociada es:

$$P_CT = \frac{\sum_{j=1}^{j=n} CT(j)}{n}$$

Debido a que esta es una métrica indirecta, es necesario identificar las métricas relacionadas, en este caso: *Proporción de Tareas Completadas sobre Tareas Propuestas (CT)*. Cabe aclarar que, en la fórmula, *n* no es un valor obtenido mediante una métrica sino que es una constante (parámetro) que representa la “Cantidad de usuarios”. A su vez se debió identificar el atributo que es cuantificado por esta métrica relacionada, el cual es *Tareas Completadas en relación a Tareas Propuestas*.

La métrica *Proporción de Tareas Completadas sobre Tareas Propuestas* también es una métrica indirecta y su función asociada es:

$$CT = \frac{T_c}{T_p}$$

Debido a que, nuevamente, se tiene una métrica indirecta, también se deben identificar las métricas y atributos relacionados a ésta. En este caso, *T_c* corresponde a la métrica directa *Número de tareas completadas correctamente* que cuantifica el atributo *Tareas Completadas* y *T_p* es la métrica directa *Número de tareas propuestas* que cuantifica a *Tareas Propuestas*.

Ahora que se ha identificado una métrica para el atributo *Compleitud de Tareas*, y también las métricas y atributos relacionados, se procede a *Seleccionar una Herramienta* que permita automatizar el procedimiento de cálculo. Sin embargo, en el caso analizado no se identificó ninguna herramienta en particular.

Especificación de Métrica

Nombre de la Métrica: Promedio del grado de Satisfacción, P_GS.

Atributo que cuantifica: Satisfacción.

Objetivo: Calcular el promedio del grado de satisfacción para todos los usuarios.

Procedimiento de Cálculo (Fórmula):
$$P_{GS} = \frac{\left(\sum_{j=1}^{j=n} GS(j) \right)}{n}$$

n: número de usuarios que intervienen en el test. Interpretación del resultado: $0 \leq P_{GS} \leq 40$, cuanto más próximo a 40, mejor.

Escala: numérica

Tipo de Escala: proporción

Unidad: grado de satisfacción

Herramienta: Ninguna

Métricas Relacionadas

Nombre de la Métrica: Satisfacción para un Usuario, GS.

Atributo que cuantifica: Satisfacción para un Usuario.

Objetivo: Conocer el grado de satisfacción de un usuario, usando el producto evaluado.

Procedimiento de Cálculo (Fórmula):
$$GS = \left(\sum_{k=1}^{\frac{m}{2}} \text{item}_{2k} + \sum_{k=1}^{\frac{m}{2}} \text{item}_{2k-1} \right)$$

ítem_k: k-esimo ítem del cuestionario, m: cantidad de ítems del cuestionario.

Cada ítem puede ser respondido seleccionando una de 5 (cinco) posibles respuestas, oscilando entre “Totalmente en Desacuerdo” y “Totalmente de Acuerdo” de una escala tipo Likert.

El puntaje para los ítems impares se calcula como la posición de la respuesta menos 1 (uno) y el puntaje para los ítems pares como 5 (cinco) menos la posición de la respuesta. De este modo, el puntaje máximo que puede obtenerse para cada ítem es 4 y el mínimo 0, y para cada cuestionario el máximo puntaje cuarenta (40). Interpretación del resultado: $0 \leq GS \leq 40$, cuanto más próximo a 40, mejor. El 40 como valor máximo del cuestionario no es arbitrario sino que responde a características del diseño de la encuesta.

Escala: numérica

Tipo de Escala: absoluto

Unidad: grado de satisfacción

Herramienta: Ninguna

Figura 4-13: Especificación de la métrica utilizada para el atributo Satisfacción.

A continuación, se vuelven a realizar las mismas tareas para los restantes atributos del árbol de requerimientos. Finalmente, como artefacto de salida se obtiene una *Especificación de Métricas*. Una métrica específica, además del nombre y el objetivo, la escala, el tipo de escala, el procedimiento de medición o de cálculo, la herramienta que automatiza el procedimiento, entre otros metadatos. En la Figura 4-13 se muestra un ejemplo de una especificación de una métrica relacionada al caso de estudio.

4.4.1.3 Implementar la Medición (A3)

Según [Fenton, y otros, 1997] la medición es “el proceso por el cual se asignan números o símbolos a atributos de entidades del mundo real de tal forma que los describa de acuerdo con reglas claramente definidas”. Precisamente, la actividad *Implementar la Medición* consiste en obtener un valor o medida, ya sea numérica o categórica, para cada uno de los atributos de una o más entidades, utilizando como entrada el conjunto de métricas obtenidas en A2 y las herramientas (si las hubiera) que automatizan los procedimientos de medición y/o cálculo. Para llevar a cabo la actividad *Implementar la Medición* se debe:

- ✓ *Establecer Entidad*
- ✓ *Medir los Atributos*

Establecer Entidad

La medición se realiza sobre atributos pertenecientes a una entidad concreta. Por lo tanto, si aún no se ha establecido una entidad (tal cual se discutió en la sección 4.4.1.1) se debe realizar obligatoriamente la tarea *Establecer Entidad*. Una vez establecida una o varias entidades, y tal como se observa en la Figura 4-14, se deben *Medir los atributos* para cada una de las entidades especificadas.

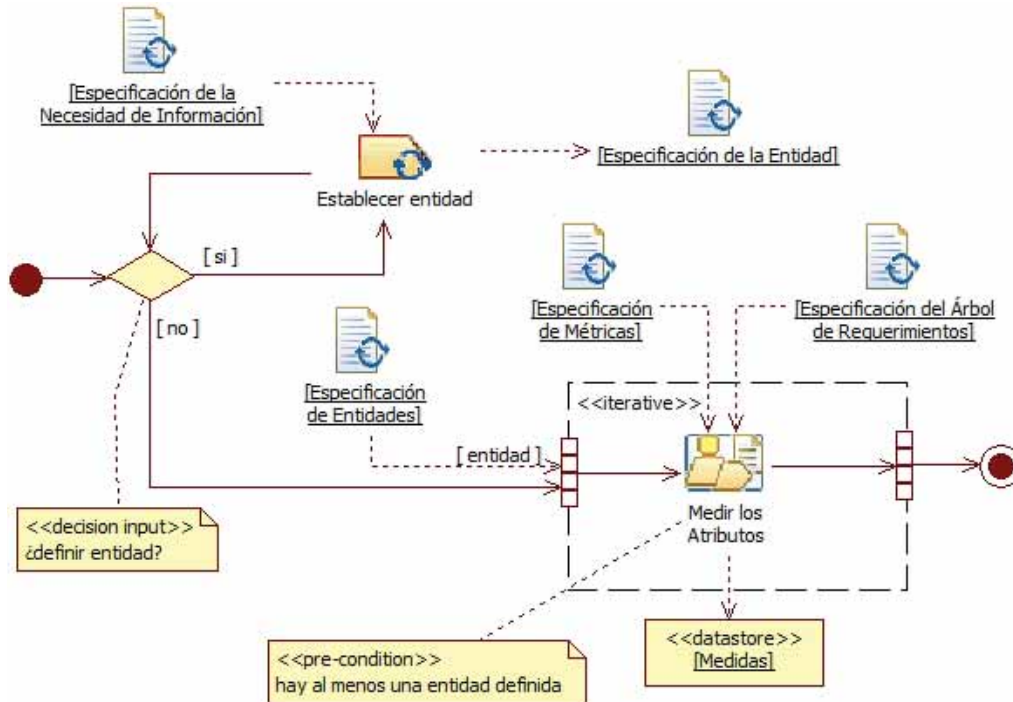


Figura 4-14: Actividad *Implementar la Medición*.

Si tenemos en cuenta el caso de estudio que venimos desarrollando, no fue necesario establecer una entidad en esta parte del proceso porque ya se había establecido en A1. Sin embargo, podría haberse definido aquí, e incluso agregar otras entidades a ser evaluadas, si fuera el caso.

Medir los Atributos

Según la Figura 4-15, se puede apreciar que *Medir los Atributos* implica cuantificar cada uno de los atributos definidos en el árbol de requerimientos. La tarea que permite obtener los valores de los atributos del árbol se denomina *Cuantificar Atributo*. Debemos recordar que es muy probable que existan atributos del árbol que son medidos utilizando métricas indirectas, por lo cual, es necesario, previamente, cuantificar los atributos relacionados, identificados en la actividad *Diseñar la Medición*. La medición (tarea *Cuantificar Atributo*) se realiza utilizando la herramienta que automatiza el procedimiento de medición o cálculo, o se realiza manualmente por un agente humano, siguiendo la especificación dada en el procedimiento de la métrica.

Como resultado, se producen las medidas asociadas a cada uno de los atributos. Notar que estas medidas deben ser adecuadamente almacenadas en un repositorio con el fin de servir como entrada a la actividad *Implementar la Evaluación* (ver Sección 4.4.1.5), como así también serán útiles al momento de

analizar la información obtenida en la evaluación y brindar recomendaciones (ver Sección 4.4.1.6).

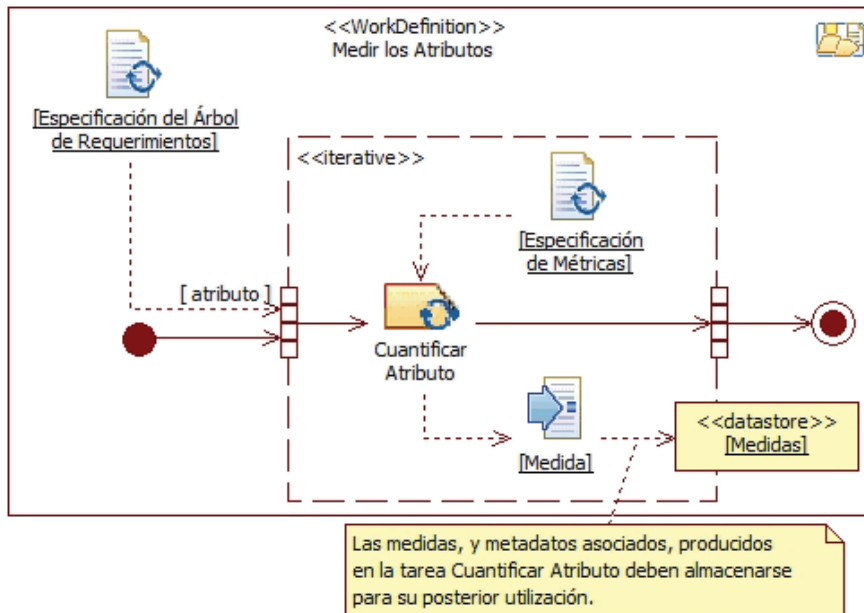


Figura 4-15: Vista funcional y de comportamiento para *Medir los Atributos*.

A modo de ejemplo nos concentraremos en la tarea *Cuantificar Atributo* para medir el atributo *Complejidad de Tareas* de la entidad *Qplus Campus Virtual*. Teniendo presente que este atributo es medido mediante la métrica indirecta *Promedio de la Proporción de Tareas Completadas sobre Tareas Propuestas para Todos los Usuarios*, en primer lugar se abordó la medición de los atributos relacionados. En este caso, posee un atributo directamente relacionado: *Tareas Completadas en relación a Tareas Propuestas*, cuya métrica asociada es: *Proporción de Tareas Completadas sobre Tareas Propuestas*. A su vez, para la medición de este atributo, antes se deben medir los atributos: *Tareas Completadas* y *Tareas Propuestas*, mediante las métricas: *Número de Tareas Completadas Correctamente* y *Número de Tareas Propuestas*, respectivamente, ya que su métrica también es indirecta.

Para la métrica *Número de Tareas Completadas Correctamente*, de acuerdo a su procedimiento de medición, se contaron para cada usuario las tareas completadas correctamente. Es decir, cada tarea en la que el usuario completó correctamente todas y cada una de las sub-metas definidas. En la Tabla 4-2 se presenta un resumen de los resultados obtenidos. En cuanto al atributo *Tareas Propuestas*, se obtuvo el valor “4”.

Tabla 4-2: Resultados obtenidos por los usuarios para la métrica *Número de Tareas Completadas Correctamente*.

Usuario	Resultados
Usuario 1	2
Usuario 2	1
Usuario 3	1
Usuario 4	1
Usuario 5	4
Usuario 6	4

Luego, fue posible calcular la métrica *Proporción de Tareas Completadas sobre Tareas Propuestas*, cuyo procedimiento de cálculo es el cociente entre las *Tareas Completadas* y las *Tareas Propuestas*. Los resultados obtenidos son los que se visualizan en la Tabla 4-3.

Tabla 4-3: Resultados obtenidos por los usuarios para la métrica *Proporción de Tareas Completadas sobre Tareas Propuestas*.

Usuario	Resultados
Usuario 1	0,50
Usuario 2	0,25
Usuario 3	0,25
Usuario 4	0,25
Usuario 5	1,00
Usuario 6	1,00

Evidentemente, los mejores resultados correspondieron a los usuarios que habían completado correctamente todas las tareas. La interpretación de la métrica indica que el resultado debe ser entre 0 (cero) y 1 (uno), y que es mejor cuanto más próximo a 1 se encuentre.

Tabla 4-4: Medidas obtenidas para cada uno de los atributos al *Implementar la Medición*.

	Medida
1. Calidad en Uso	
1.1. Eficacia	
1.1.1. <i>Compleitud de Tareas</i>	0,542
1.1.2. <i>Eficacia de Tarea</i>	0,656
1.2. Productividad	
1.2.1. <i>Eficiencia en relación completitud de tareas</i>	0,498
1.2.2. <i>Eficiencia en relación a Eficacia</i>	1,306
1.3. Satisfacción	34,83

Ahora ya es posible calcular la métrica *Promedio de la Proporción de Tareas Completadas sobre Tareas Propuestas para Todos los Usuarios*, realizando el cociente entre la sumatoria de los resultados obtenidos para la métrica *Proporción de Tareas Completadas sobre Tareas Propuestas* y el número de usuarios, en este caso 6. El resultado obtenido fue 0,542. De manera similar, se midieron todos los atributos, obteniendo finalmente los valores (medidas) mostrados en la Tabla 4-4.

4.4.1.4 Diseñar la Evaluación (A4)

El valor de una métrica no representa por sí mismo el nivel de satisfacción de un requerimiento elemental (atributo). Por lo tanto es necesario definir una transformación a partir del valor medido que produzca un nuevo valor numérico o categórico. Esta transformación está dada por un Indicador Elemental, tal como se comentó al describir el componente de Evaluación presentado en la sección 4.2.5. Asimismo, existen Indicadores Derivados, los cuales permiten interpretar los requerimientos de mayor nivel de abstracción, es decir, los conceptos calculables.

La actividad *Diseñar la Evaluación* permite definir, para cada atributo y concepto calculable del árbol de requerimientos, un indicador que lo evaluará. Es decir, definir cómo se van a interpretar los valores de los atributos, para luego obtener el grado de satisfacción brindado por cada uno de los requerimientos de mayor nivel de abstracción.

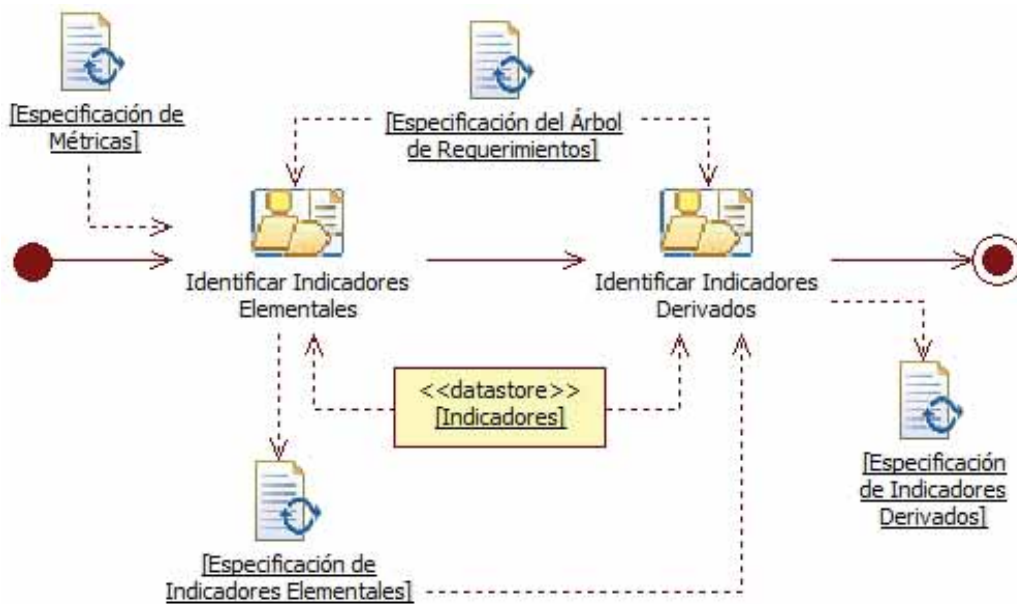


Figura 4-16: Actividades para *Diseñar la Evaluación*.

Como entradas a esta actividad principal, se debe contar con la *Especificación del Árbol de Requerimientos* (la cual forma parte de la *Especificación de los Requerimientos No Funcionales*) y la *Especificación de Métricas*, producidas en las actividades A1 y A2 respectivamente (recordar Figura 4-3). En la Figura 4-16 se pueden observar dichas entradas y las siguientes sub-actividades:

- ✓ *Identificar Indicadores Elementales*
- ✓ *Identificar Indicadores Derivados*

Identificar Indicadores Elementales

Esta sub-actividad consiste en especificar un indicador elemental para cada uno de los atributos de la *Especificación del Árbol de Requerimientos*. El artefacto *Especificación de Métricas* sirve como entrada a esta actividad ya que en este se encuentran los metadatos de las métricas (como por ejemplo la escala y el tipo de escala) que permitirán especificar cómo se transformará el valor medido del atributo en un valor de indicador.

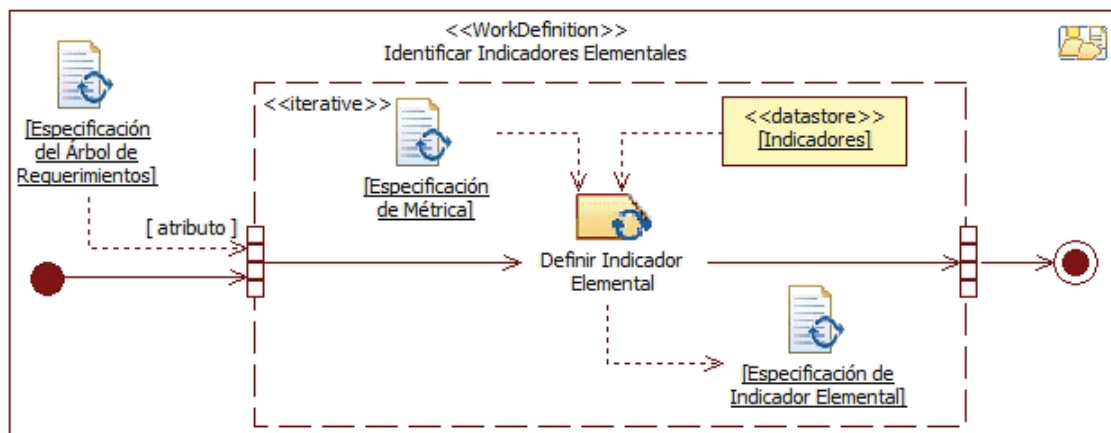


Figura 4-17: Actividades involucradas para *Identificar Indicadores Elementales*.

En la Figura 4-17 se puede notar que para *Identificar Indicadores Elementales* se debe llevar a cabo la tarea *Definir Indicador Elemental* de manera iterativa para cada uno de los atributos del árbol de requerimientos. Esta tarea consiste en establecer el modelo elemental para el indicador, es decir una función que transforme un valor medido en un valor de indicador. Esta transformación o mapeo puede ser de diferentes tipos, por ejemplo: exacto, por rangos o por umbrales, por mencionar algunos. A su vez, es necesario identificar los criterios de decisión (ver componente de Evaluación de la Figura 4-1) que permitan interpretar el valor de indicador. Además, se debe identificar la escala, el tipo de escala y la unidad para el indicador que se está especificando. Cabe notar que la escala, tipo de escala y unidad son dependientes de la fórmula o modelo del indicador. Muchos de los metadatos asociados al indicador pueden ser recuperados desde un repositorio de indicadores, lo cual agiliza el diseño.

Para ejemplificar la tarea *Definir Indicador Elemental*, volvemos a centrarnos en el caso de estudio analizado. Teniendo presente el primer atributo del árbol de requerimientos (*Compleitud de Tareas*) definimos su indicador elemental. La métrica asociada a este atributo en la actividad *Diseñar la Medición* es *Promedio de la Proporción de Tareas Completadas sobre Tareas Propuestas para Todos los Usuarios* y el indicador que se define es denominado *Grado de Cumplimiento Respecto a Proporción de Tareas Completadas sobre Tareas Propuestas (%P_CT_P)*. El modelo elemental o función de transformación se definió como sigue:

$$\%P_CT_P = P_CT * 100$$

donde *P_CT* es el *Promedio de la Proporción de Tareas Completadas sobre Tareas Propuestas para Todos los Usuarios*.

Una vez definido el modelo elemental se identificaron los siguientes criterios de decisión:

- $0 \leq \%P_CT_P \leq 45$: no satisfactorio
- $45 < \%P_CT_P \leq 70$: marginal (regular)
- $70 < \%P_CT_P \leq 100$: satisfactorio

donde *no satisfactorio* significa “No cumple los requisitos mínimos de satisfacción. Se requieren cambios urgentes”, *marginal (regular)* significa “Cumple con los requisitos mínimos de satisfacción. Se requiere una planificación de cambios para la mejora” y *satisfactorio* significa “Cumple con los requisitos de forma satisfactoria. No se requieren cambios”. Por último, los metadatos asociados a la escala fueron los siguientes:

- Escala: numérica
- Tipo de Escala: proporción
- Unidad: % (porcentaje)

En la Figura 4-18 se muestra la Especificación de Indicador Elemental para el atributo *Compleitud de Tareas*.

Especificación de Indicador Elemental

Nombre del Indicador Elemental: Grado de Cumplimiento Respecto a Proporción de Tareas Completadas sobre Tareas Propuestas, %P_CT_P.

Atributo que evalúa: Completitud de Tareas

Modelo Elemental: $\%P_CT_P = P_CT * 100$

Donde P_CT: Promedio de la Proporción de Tareas Completadas sobre Tareas Propuestas para Todos los Usuarios

Criterios de Decisión:

$0 \leq \%P_CT_P \leq 45$: no satisfactorio

$45 < \%P_CT_P \leq 70$: marginal (regular)

$70 < \%P_CT_P \leq 100$: satisfactorio

Donde:

no satisfactorio significa: No cumple los requisitos mínimos de satisfacción. Se requieren cambios urgentes.

marginal (regular) significa: Cumple con los requisitos mínimos de satisfacción. Se requiere una planificación de cambios para la mejora.

satisfactorio significa: Cumple con los requisitos de forma satisfactoria. No se requieren cambios.

Procedimiento de Cálculo: Obtener el valor de la métrica Promedio de la Proporción de Tareas Completadas sobre Tareas Propuestas para Todos los Usuarios (P_CT) y multiplicarlo por 100.

Escala: numérica

Tipo de Escala: proporción

Unidad: % (porcentaje)

Figura 4-18: Especificación del Indicador Elemental *Grado de Cumplimiento Respecto a Proporción de Tareas Completadas sobre Tareas Propuestas*.

Identificar Indicadores Derivados

Una vez definido cómo se evaluarán los atributos, se debe proceder, tal como se muestra en la Figura 4-19, a definir los indicadores derivados que se utilizarán para evaluar cada uno de los conceptos y sub-conceptos definidos en la actividad A1.

Al igual que para definir un indicador elemental, al definir un indicador derivado se debe indicar el modelo global (función o algoritmo) mediante el cual se calculará el nivel de satisfacción de los diferentes conceptos calculables del árbol de requerimientos. Un modelo global, por ejemplo, puede ser un modelo LSP -*Logic Scoring of Preference*- o un modelo de redes neuronales, entre otros. A menudo, estos modelos deben ser parametrizados o configurados. Tal es el caso de un modelo aditivo basado en pesos, donde se deben asociar pesos a los atributos o subconceptos de un mismo nivel, como una forma de reflejar mayor importancia de unos respecto de otros. Otros parámetros, por ejemplo al utilizar redes neuronales, pueden ser los pesos asociados a cada neurona o el factor de aprendizaje empleado para corregir los pesos de forma dinámica al calcular el valor del indicador.

También se deben fijar los criterios de decisión asociados al modelo global para indicar la forma en que cada valor de indicador debe ser interpretado. Por último, se debe identificar la escala, el tipo de escala y la unidad que tendrán los indicadores.

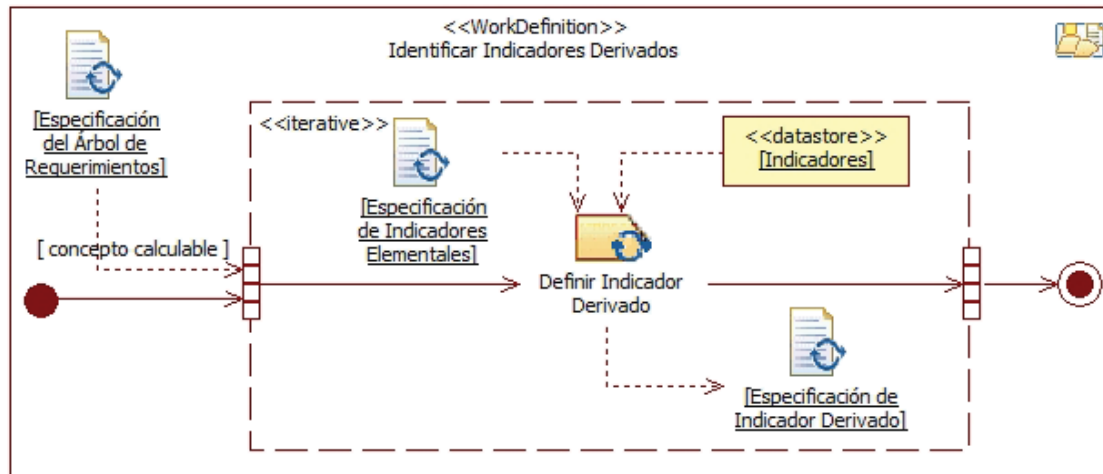


Figura 4-19: Identificar Indicadores Derivados.

Para el caso de estudio de *Qplus Campus Virtual*, se optó por utilizar el mismo modelo global para todos los indicadores. Particularmente, el modelo escogido fue el modelo LSP [Dujmovic, 1996]. Este es un modelo de agregación lógica de preferencias, basado en pesos, que permite utilizar operadores de preferencia lógica para modelar distintos niveles de intensidad de conjunción/disyunción (“y/o”), entre otros aspectos. Este modelo fue elegido en el caso de estudio debido a que se consideró más apropiado un modelo de agregación lógico que uno simplemente lineal y aditivo. Esta decisión está justificada en el hecho de que un modelo de agregación lógico multi-criterio permite manejar consistentemente relaciones de simultaneidad, reemplazabilidad y neutralidad, además de superar el problema de modelar requerimientos obligatorios que carece un modelo lineal y aditivo. El modelo LSP se basa en la siguiente fórmula o función:

$$ID(r) = (P_1 * I_1^r + P_2 * I_2^r + \dots + P_m * I_m^r)^{1/r}$$

donde:

ID representa el valor del Indicador Derivado a calcular

I_i son los valores de los Indicadores del nivel inferior, $0 \leq I_i \leq 1$

P_i representan pesos o ponderación, $(P_1 + P_2 + \dots + P_m) = 1$; $P_i > 0$; $i = 1 \dots m$

r es un coeficiente conjuntivo/disyuntivo, $-\infty \leq r \leq +\infty$

Según la función anterior se observa que existen dos parámetros a ser fijados: los pesos (P_i) y el coeficiente del bloque de agregación (r). Para los pesos se asignó igual importancia a cada atributo dentro de cada subconcepto y, del mismo modo, a cada subconcepto (y al atributo Satisfacción) que conforman el concepto calculable *Calidad en Uso*. Este criterio se empleó teniendo en cuenta que se trataba de un estudio exploratorio. En la Tabla 4-5 se puede observar la distribución equitativa de los pesos.

Con respecto al parámetro r , la elección de su valor¹¹ permite seleccionar un operador que va desde la pura conjunción y la cuasi-conjunción hasta la cuasi-disyunción y la pura disyunción. El punto medio representa a la aditividad (para $r=1$) la cual no está ni conjuntiva ni disyuntivamente polarizada, es decir, representa una función de relaciones de neutralidad o independencia entre las

¹¹ Los distintos valores de r , dependiente del tipo de operador y la cantidad de entradas, ya se encuentran tabulados en el modelo LSP [Dujmovic, 1996].

entradas. En el caso de estudio, a nivel de subconceptos, el operador lógico empleado fue C-, un operador para modelar una relación de cuasi-conjunción débil. Esto significa que un valor 0 (cero) en alguno de los atributos de entrada no producirá un 0 (cero) en la salida, aunque “castigará” a la misma. A nivel de concepto calculable (Calidad en Uso), se empleó el operador lógico C++ para modelar una relación de cuasi-conjunción fuerte, lo que implica que para obtener valores distintos de 0 (cero) de salida es obligatoria la existencia de valores distintos de 0 (cero) en las subcaracterísticas de entrada. Además, cuanto mayor sean esos valores de entrada, mayor será el grado de satisfacción global (concepto calculable foco). En la Tabla 4-5 se indican los operadores lógicos asignados en cada caso.

Tabla 4-5: Asignación de operadores lógicos y de pesos.

	Operador	Peso
1. Calidad en Uso	C++	
1.1. Eficacia	C-	0,33
1.1.1. <i>Compleitud de Tareas</i>		0,50
1.1.2. <i>Eficacia de Tarea</i>		0,50
1.2. Productividad	C-	0,33
1.2.1. <i>Eficiencia en relación completitud de tareas</i>		0,50
1.2.2. <i>Eficiencia en relación a Eficacia</i>		0,50
1.3. <i>Satisfacción</i>		0,33

En cuanto a los criterios de decisión y la escala, se ha optado por escoger los mismos que para el ejemplo del indicador elemental. De hecho, lo más apropiado es que todos los indicadores, tanto elementales como derivados, posean los mismos criterios de decisión y escala para conseguir un conjunto de valores normalizados y así facilitar la interpretación de los resultados.

Al finalizar la actividad *Diseñar la Evaluación*, como artefacto de salida, se obtiene un documento con la especificación de los distintos indicadores elementales y derivados (llamado *Especificación de Indicadores*), el cual contiene la escala y los niveles (o grados) de aceptabilidad, entre otros metadatos, necesarios para calcular el árbol de requerimientos y determinar finalmente el nivel de satisfacción global para la necesidad de información establecida. A modo de ejemplo, en la Figura 4-20 se esboza el artefacto *Especificación de Indicadores* producido en el caso de estudio.

Especificación de Indicadores

Especificación de Indicadores Elementales

Nombre del Indicador Elemental: Grado de Cumplimiento Respecto a Proporción de Tareas Completadas sobre Tareas Propuestas, %P_CT_P.

Atributo que evalúa: Completitud de Tareas

Modelo Elemental: $\%P_CT_P = P_CT * 100$

Donde P_CT: Promedio de la Proporción de Tareas Completadas sobre Tareas Propuestas para Todos los Usuarios

Criterios de Decisión:

$0 \leq \%P_CT_P \leq 45$: no satisfactorio

$45 < \%P_CT_P \leq 70$: marginal (regular)

$70 < \%P_CT_P \leq 100$: satisfactorio

Donde:

no satisfactorio significa: No cumple los requisitos mínimos de satisfacción.

Se requieren cambios urgentes.

marginal (regular) significa: Cumple con los requisitos mínimos de satisfacción. Se requiere una planificación de cambios para la mejora.

satisfactorio significa: Cumple con los requisitos de forma satisfactoria. No se requieren cambios.

Procedimiento de Cálculo: Obtener el valor de la métrica Promedio de la Proporción de Tareas Completadas sobre Tareas Propuestas para Todos los Usuarios (P_CT) y multiplicarlo por 100.

Escala: numérica

Tipo de Escala: proporción

Unidad: % (porcentaje)

Nombre del Indicador Elemental: ...

⋮

Especificación de Indicadores Derivados

Conceptos de Nivel 1

Nombre del Indicador Derivado: Nivel de Eficacia, N_Eficacia

Concepto Calculable que evalúa: Eficacia

Modelo Global (instanciado):

$$N_Eficacia = (0.5 * \%P_CT_P^{0,261} + 0.5 * \%P_ETT_u_P^{0,261})^{1/0,261}$$

Criterios de Decisión:

$0 \leq N_Eficacia \leq 45$: no satisfactorio

$45 < N_Eficacia \leq 70$: marginal (regular)

$70 < N_Eficacia \leq 100$: satisfactorio

Donde:

no satisfactorio significa: ...

Procedimiento de Cálculo: ...

Escala: numérica

Tipo de Escala: proporción

Unidad: % (porcentaje)

Nombre del Indicador Derivado: Nivel de Productividad, N_Productividad

Concepto Calculable que evalúa: Productividad

⋮

Conceptos de Nivel 0

Nombre del Indicador Derivado: Nivel de Calidad en Uso, N_CalidadEnUso

Concepto Calculable que evalúa: Calidad en Uso

Modelo Global (instanciado):

$$N_CalidadEnUso = (0.3 * N_Eficacia^{-7,639} + 0.3 * N_Productividad^{-7,639} + 0.3 * \%P_GS_P^{-7,639})^{1/-7,639}$$

Criterios de Decisión:

⋮

Figura 4-20: Ejemplo de una *Especificación de Indicadores*.

4.4.1.5 Implementar la Evaluación (A5)

La ejecución de esta actividad permite obtener los valores de los distintos indicadores asociados a cada uno de los elementos del árbol de requerimientos, utilizando la *Especificación de Indicadores* obtenida en la actividad A4 y las medidas obtenidas en A3 (recordar Figura 4-3). Al finalizar esta actividad, se podrá conocer el grado de satisfacción proporcionado por los diferentes conceptos calculables y atributos de la entidad o entidades analizadas. *Implementar la Evaluación* consiste en llevar a cabo, para cada una de las entidades, las siguientes sub-actividades (ver la Figura 4-21):

- ✓ *Calcular Indicadores Elementales*
- ✓ *Calcular Indicadores Derivados*

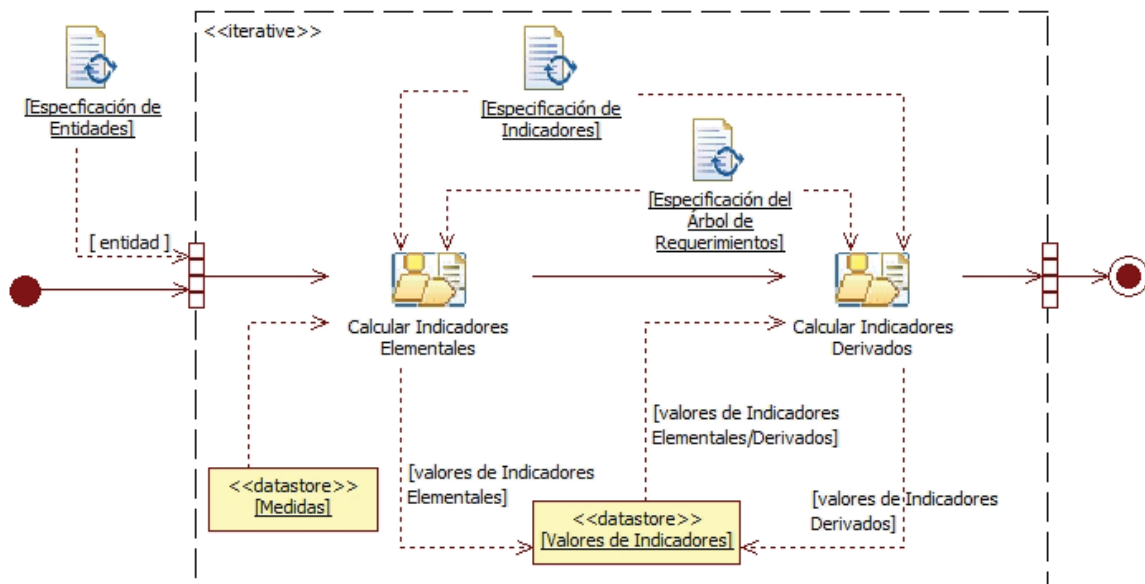


Figura 4-21: Actividad *Implementar la Evaluación*.

Cabe mencionar que, si bien en la Figura 4-21 se modela como un proceso iterativo, el mismo puede realizarse de forma paralela. Es decir, en lugar de calcular los indicadores para una de las entidades, luego para otra, y así hasta culminar con todas las entidades, se puede realizar, al mismo tiempo, el cálculo de los indicadores de cada entidad. Esto es posible debido a que el cálculo de los indicadores para cada entidad es independiente de los valores de los indicadores de entidades distintas. Lo mismo puede decirse en el caso de la medición (ver Figura 4-14).

Calcular Indicadores Elementales

Esta sub-actividad permite realizar una transformación de los valores obtenidos en la medición, normalmente heterogéneos en cuanto a su unidad, a un conjunto de valores normalizados. El cálculo de los indicadores elementales se logra ejecutando la tarea *Calcular Indicador Elemental* para cada atributo del árbol de requerimientos, siguiendo la descripción del procedimiento dada en la especificación del indicador. En la Figura 4-22 se muestra la sub-actividad *Calcular Indicadores Elementales*.

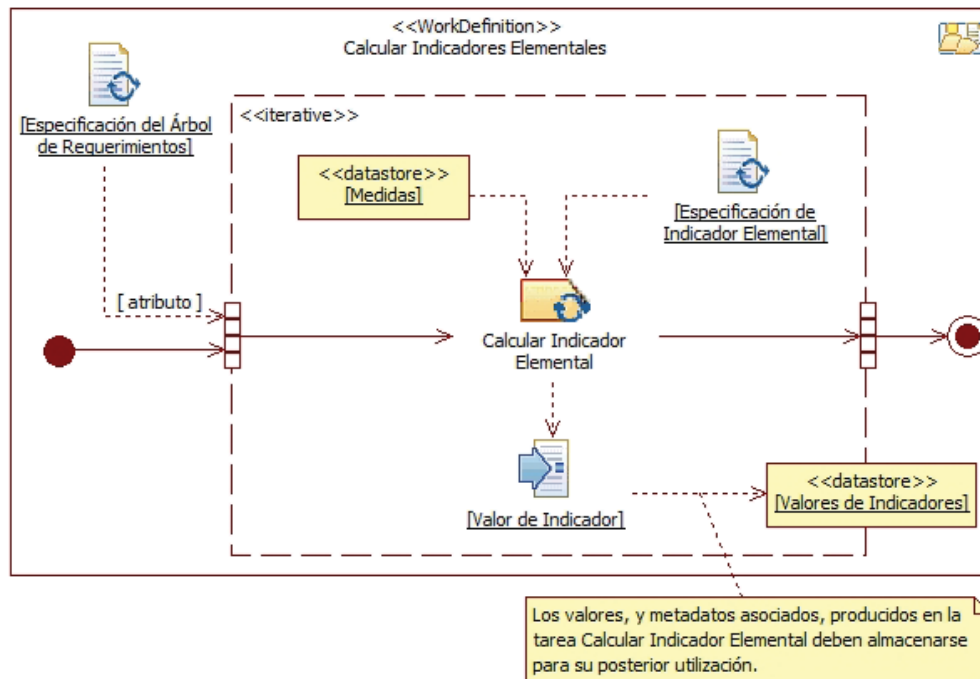


Figura 4-22: Flujo de actividades para *Calcular Indicadores Elementales*.

Para ejemplificar la actividad *Calcular Indicadores Elementales* vamos a calcular el indicador *Grado de Cumplimiento Respecto a Proporción de Tareas Completadas sobre Tareas Propuestas (%P_CT_P)* para el atributo *Compleitud de Tareas*. Teniendo en cuenta el procedimiento de cálculo asociado al modelo elemental del indicador, obtenemos como valor: *54,20*. Este valor, según los criterios de decisión establecidos, corresponde a un nivel de satisfacción marginal (regular), lo que significa que *“cumple con los requisitos mínimos de satisfacción, pero se requiere una planificación de cambios para la mejora”*.

Tabla 4-6: Valores obtenidos al *Calcular Indicadores Elementales*.

	Nivel de Satisfacción (%)
1. Calidad en Uso	
1.1. Eficacia	
1.1.1. <i>Compleitud de Tareas</i>	54,20
1.1.2. <i>Eficacia de Tarea</i>	65,60
1.2. Productividad	
1.2.1. <i>Eficiencia en relación completitud de tareas</i>	49,80
1.2.2. <i>Eficiencia en relación a Eficacia</i>	54,04
1.3. Satisfacción	87,08

Procediendo de manera similar, se obtuvieron los valores del resto de los indicadores elementales, logrando los valores que se muestran en la Tabla 4-6.

Calcular Indicadores Derivados

Una vez obtenidos todos los valores de los indicadores elementales, sólo resta *Calcular Indicadores Derivados*. Esta actividad produce todos los valores de los indicadores derivados, teniendo como entrada los valores de los indicadores elementales, como así también la *Especificación de Indicadores* (recordar Figura 4-21).

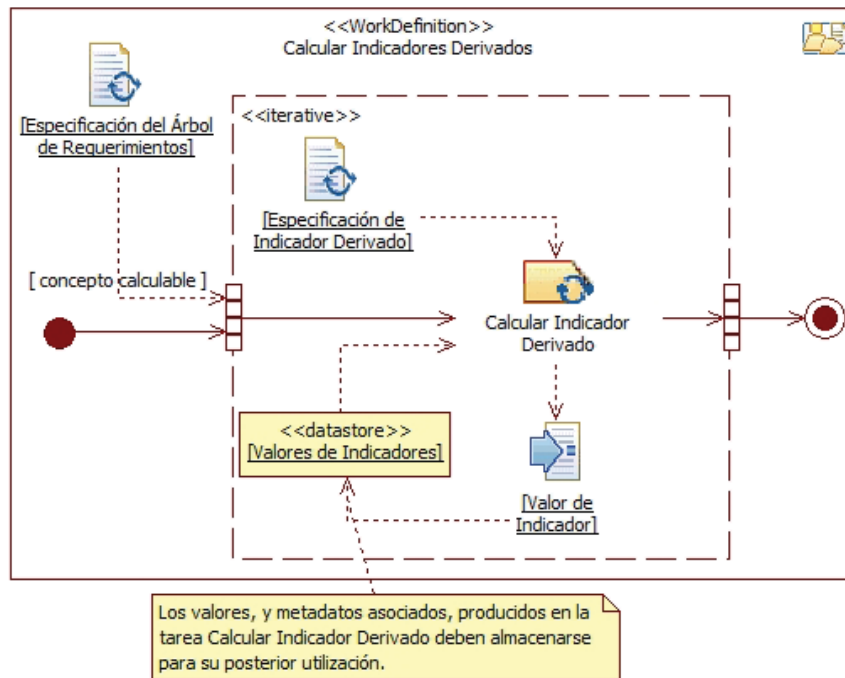


Figura 4-23: Flujo de actividades para *Calcular Indicadores Derivados*.

Como se puede apreciar en la Figura 4-23, el cálculo de los indicadores derivados se realiza de manera similar al de los indicadores elementales. La diferencia reside en que el cálculo de los indicadores derivados depende de los valores de los indicadores elementales (o de otros indicadores derivados si hubiere más de un nivel de subconceptos) y no, al menos en forma directa, de las medidas en sí. Al igual que todos los cálculos anteriormente mencionados, éste puede realizarse por medio de un agente automático o humano, ya sea utilizando una herramienta que automatice el procedimiento de cálculo o siguiendo la especificación del procedimiento.

Debido a que los conceptos calculables están jerarquizados en el árbol de requerimientos, los indicadores asociados deben calcularse siguiendo este orden jerárquico hacia arriba. A modo de ejemplo, teniendo presente el árbol de requerimientos de la Tabla 4-6, en primer lugar se deben calcular los indicadores correspondientes a Eficacia y Productividad (por ser los conceptos de más bajo nivel en el árbol), y luego se calcula el indicador asociado a *Calidad en Uso*.

Al finalizar la actividad *Implementar la Evaluación* se obtienen los valores de los diferentes indicadores, tanto elementales como derivados, que servirán de entrada a la actividad de análisis y recomendación (ver sección 4.4.1.6).

En el caso de estudio presentado, al *Calcular Indicadores Derivados* se utilizó la herramienta WebQEMTool [Olsina L. , y otros, 2001], que facilita el cálculo y la interpretación de los datos, el seguimiento de la evaluación y que, además, tiene mecanismos de documentación. Esta herramienta permite seleccionar entre dos clases de modelo de agregación (aditivo y LSP) y una vez ingresados los pesos de cada atributo, los operadores lógicos y los puntajes de los indicadores elementales, realiza automáticamente el cálculo de los indicadores derivados. Sin embargo, a modo de ejemplo, aquí haremos el cálculo del indicador derivado *Nivel de Eficacia (N_Eficacia)*, utilizando la especificación dada en la Figura 4-20. Por lo tanto, para obtener el valor del indicador se sigue la fórmula del modelo global:

$$N_Eficacia = (0.5 * \%P_CT_P^{0,261} + 0.5 * \%P_ETT_u_P^{0,261})^{1/0,261}$$

donde: $\%P_CT_P$ corresponde a *Grado de Cumplimiento Respecto a Proporción de Tareas Completadas sobre Tareas Propuestas* (asociado al atributo *Compleitud de Tareas*), y $\%P_ETT_u_P$ corresponde a *Grado de Cumplimiento Respecto a Proporción de Tareas Completadas por Todos los Usuarios* (asociado al atributo *Eficacia de Tarea*).

Por lo tanto, si ahora tenemos en cuenta los valores de la Tabla 4-6, el valor del indicador resultante es:

$$N_Eficacia = (0.5 * 54,20^{0,261} + 0.5 * 65,60^{0,261})^{1/0,261} = 59,70$$

De igual forma, se obtiene el indicador asociado a *Productividad* y, finalmente, el indicador asociado a *Calidad en Uso*. En la Tabla 4-7 se pueden observar los valores obtenidos para cada indicador al ejecutar la actividad principal A5 para el caso de estudio.

Tabla 4-7: Resultados de los indicadores para el caso *Qplus Campus Virtual*.

	Nivel de Satisfacción [%]
1. Calidad en Uso	57,45
1.1. Eficacia	59,70
1.1.1. <i>Compleitud de Tareas</i>	54,20
1.1.2. <i>Eficacia de Tarea</i>	65,60
1.2. Productividad	51,89
1.2.1. <i>Eficiencia en relación completitud de tareas</i>	49,80
1.2.2. <i>Eficiencia en relación a Eficacia</i>	54,04
1.3. <i>Satisfacción</i>	87,08

4.4.1.6 Analizar y Recomendar (A6)

Según [Software Engineering Laboratory, 1995], cualquier actividad de recolección de datos (como por ejemplo, medición) sin análisis es una pérdida de esfuerzo. Teniendo esto presente, la actividad *Analizar y Recomendar* consiste en la realización de tareas orientadas al análisis de los datos e información provista por las métricas e indicadores y la justificación de los resultados obtenidos. Las principales tareas a seguir son (ver Figura 4-24):

- ✓ *Diseñar el análisis*
- ✓ *Implementar el análisis*
- ✓ *Elaborar el informe de conclusión*
- ✓ *Realizar recomendaciones*

Primeramente se debe *Diseñar el análisis* teniendo en cuenta los tipos de los datos (conocidos a través de las especificaciones de métricas e indicadores), conjuntamente con las *Necesidades y Objetivos de la Organización* y otros documentos como las especificaciones del *Contexto* y la *Necesidad de Información* (incluidos en la *Especificación de los Requerimientos No Funcionales*). Esta tarea implica definir qué métodos y técnicas matemáticos y estadísticos de análisis se utilizarán, ya sean de tipo cuantitativo y/o cualitativo, y los objetivos que se persiguen al utilizarlos.

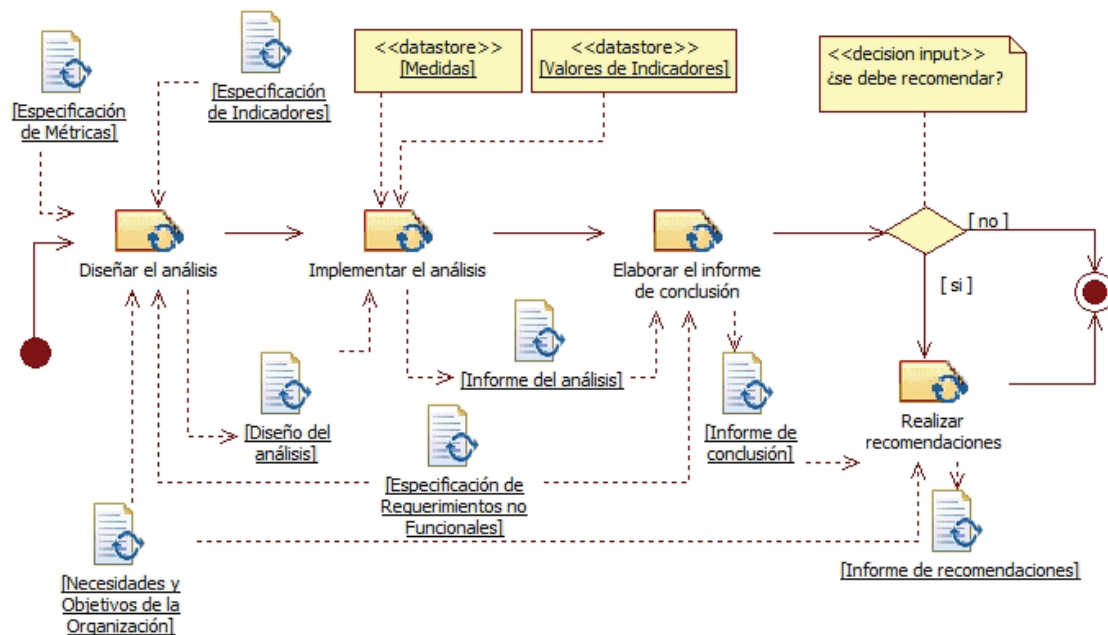


Figura 4-24: Tareas intervinientes durante la actividad *Analizar y Recomendar*.

También es importante especificar, durante esta tarea, cuales serán las herramientas que se utilizarán para realizar el análisis, y también los medios a través de los cuales luego se mostrará la información producida, tal como pueden ser tablas, cuadros comparativos y gráficos, entre otros. Algunas herramientas y mecanismos de documentación para facilitar la interpretación de los datos y su seguimiento son: WebQEM_Tool [Olsina L. , y otros, 2001], TAW (Test de Accesibilidad Web - <http://www.tawdis.net>), MINITAB (herramienta de estadística - <http://www.minitab.com/es-mx/products/minitab/>), por citar sólo unas pocas.

Luego de haber producido el *Diseño del análisis*, se procede a *Implementar el análisis* para así obtener resultados más claros y concisos. Los datos numéricos e información generada a partir de las medidas y los valores de indicadores deben ser volcados en un *Informe del análisis* mediante tablas, gráficos y otros documentos que permitan visualizar apropiadamente los resultados conseguidos y así facilitar, luego, la toma de decisiones.

Una vez diseñado e implementado el análisis, el proceso culmina con el *Informe de conclusión* del caso. Para ello se analizan los resultados del análisis considerando el foco de la evaluación, el propósito y el perfil de usuario establecidos en la *Especificación de la Necesidad de Información*. Esta tarea se denomina *Elaborar el informe de conclusión*. En muchos casos, no sólo se debe realizar un informe de conclusión, sino que también se debe entregar a los interesados un *Informe de recomendaciones*, el cual contenga, por ejemplo, qué cursos de acción (quizás incluyendo debilidades y fortalezas de cada uno) se deben seguir para hacer frente a los problemas identificados como producto del análisis. Para ello es necesario llevar a cabo la tarea *Realizar recomendaciones*. En la Figura 4-25, se muestra un extracto de las conclusiones obtenidas para el caso de estudio analizado.

Informe de Conclusión

Considerando los resultados obtenidos, se observa que el cálculo del puntaje del indicador de Calidad en Uso resultó por debajo del valor mínimo de aceptabilidad del 70,00%, arrojando un 57,45%. Además, teniendo en cuenta el puntaje a nivel características, esto es: Eficacia (59,70), Productividad (51,89) y Satisfacción: (87,08) no se cumple con la hipótesis de que “al menos, a nivel de características, el sitio está por encima de un umbral de satisfacción aceptable”, ya que solo Satisfacción está por encima del nivel de aceptabilidad esperado del 70%.

Dado que la Eficiencia está asociada al consumo de recursos para llevar adelante una tarea y, en este caso el recurso mensurado fue tiempo, se puede concluir que, o bien hay ciertos mecanismos que no son lo suficientemente fáciles de aprender, o bien no son sencillos de utilizar para un usuario intencional pero no experto. Parece lógico, entonces, que sea necesaria una gran cantidad de tiempo para poder finalizar una tarea correctamente, si el usuario no tiene el entrenamiento necesario.

Figura 4-25: Extracto de las conclusiones obtenidas para el caso de estudio analizado.

4.4.2 Roles de GOCAME: Vista Organizacional

Las diferentes tareas de un proceso son ejecutadas por agentes en cumplimiento de determinados roles. Un rol representa un “conjunto de habilidades (capacidades, competencias y responsabilidades) que debe poseer un agente para realizar una definición de trabajo”, según se definió en la Tabla 3-3. En la Figura 4-26 se pueden apreciar los roles involucrados en un Proyecto de M&E de GOCAME¹².



Figura 4-26: Roles involucrados en GOCAME.

¹² Los nombres de los roles se presentan en inglés tal cual aparecen en la publicación original [Becker, y otros, 2012].

Tabla 4-8: Definición de los roles de GOCAME.

Rol	Responsabilidades / Capacidades	Actividades (ver Figura 4-27)
Quality Assurance (QA) Manager	Responsable de liderar un proyecto de M&E (<i>ME Project</i>).	➤ Supervisa todas las actividades de M&E
Non Functional Requirements (NFR) Manager	Responsable de liderar un proyecto de requerimientos no funcionales (<i>Requirements Project</i>). Este rol debiera ser jugado por un ingeniero en requerimientos.	➤ Definir los Requerimientos No Funcionales
Measurement Manager	Responsable de liderar un proyecto de medición (<i>Measurement Project</i>).	➤ Diseñar la Medición ➤ Implementar la Medición
Evaluation Manager	Responsable de liderar un proyecto de evaluación (<i>Evaluation Project</i>).	➤ Diseñar la Evaluación ➤ Implementar la Evaluación
Evaluation Requester	Responsable de solicitar la evaluación. Colabora aportando información valiosa relacionada a las necesidades y objetivos de la organización. Notar que este rol puede ser cumplido por un humano o una organización.	➤ Definir los Requerimientos No Funcionales ➤ Diseñar la Medición ➤ Diseñar la Evaluación
Metrics Expert	Responsable de identificar las métricas (<i>Metric</i>) y herramientas (<i>Tool</i>) apropiadas para cada atributo (<i>Attribute</i>) del árbol de requerimientos (<i>Requirements tree</i>). Este rol debe tener permisos de acceso al catálogo de métricas.	➤ Diseñar la Medición
Data Calculator	Responsable de calcular las medidas derivadas (<i>Derived Measure</i>) usando la <i>Especificación de Métricas</i> . Este rol debe tener conocimientos en el uso de herramientas de medición.	➤ Implementar la Medición
Data Collector	Responsable de obtener las medidas base (<i>Base Measure</i>) usando la <i>Especificación de Métricas</i> . Este rol debe tener conocimientos en el uso de herramientas de medición.	➤ Implementar la Medición
Indicators Expert	Responsable de identificar los indicadores (<i>Indicator</i>) más apropiados y para definir los criterios de decisión (<i>Decision Criterion</i>) para cada atributo (<i>Attribute</i>) y concepto calculable (<i>Calculable Concept</i>) del árbol de requerimientos (<i>Requirements tree</i>). Este rol debe tener permisos de acceso al catálogo de indicadores.	➤ Diseñar la Evaluación
Indicators Calculator	Responsable de calcular los valores de indicadores (<i>Indicator Value</i>) usando la <i>Especificación de Indicadores</i> . Notar que este rol usualmente es cumplido por un agente automático (software).	➤ Implementar la Evaluación
Analysis Designer	Responsable de identificar los métodos y técnicas de análisis de datos más apropiados a ser usados teniendo en cuenta las escalas (<i>Scale</i>) y tipos de escala (<i>Scale Type</i>) así como las <i>Necesidades y Objetivos de la Organización</i> . Adicionalmente, este rol debe conocer técnicas de documentación y visualización.	➤ Analizar y Recomendar
Data Analyzer	Responsable de conducir el análisis de datos basado en el <i>Diseño del análisis</i> . Notar que este rol puede ser	➤ Analizar y Recomendar

	cumplido por un agente humano o automático, o por ambos.	
Recommender	Responsable de conducir las recomendaciones basadas en el <i>Informe de Conclusión</i> .	➤ Analizar y Recomendar
Results Communicator	Responsable de comunicar los resultados de la evaluación y las recomendaciones al Evaluation Requester.	➤ Analizar y Recomendar

Según [Heidrich, y otros, 2006], un rol debe definirse en base a tres aspectos: *i)* responsabilidades: es decir obligaciones y permisos del rol con respecto al proceso, como por ejemplo crear un informe financiero, acceder a los registros de los empleados, etc.; *ii)* actividades: esto es, la participación del rol en las actividades; y *iii)* capacidades: requisitos de experiencia y habilidades de los agentes que ocupan el rol, como por ejemplo estar capacitado en el uso de PHP, análisis de datos con técnicas de estadísticas paramétricas, etc.

En la Tabla 4-8 se definen los roles de GOCAME en base a sus responsabilidades y las actividades en las cuales están involucrados. Notar que en las definiciones se han resaltado en *itálica* ciertos elementos del proceso de GOCAME y también los términos de C-INCAMI para mostrar la inclusión de la terminología de M&E en la definición de los roles.

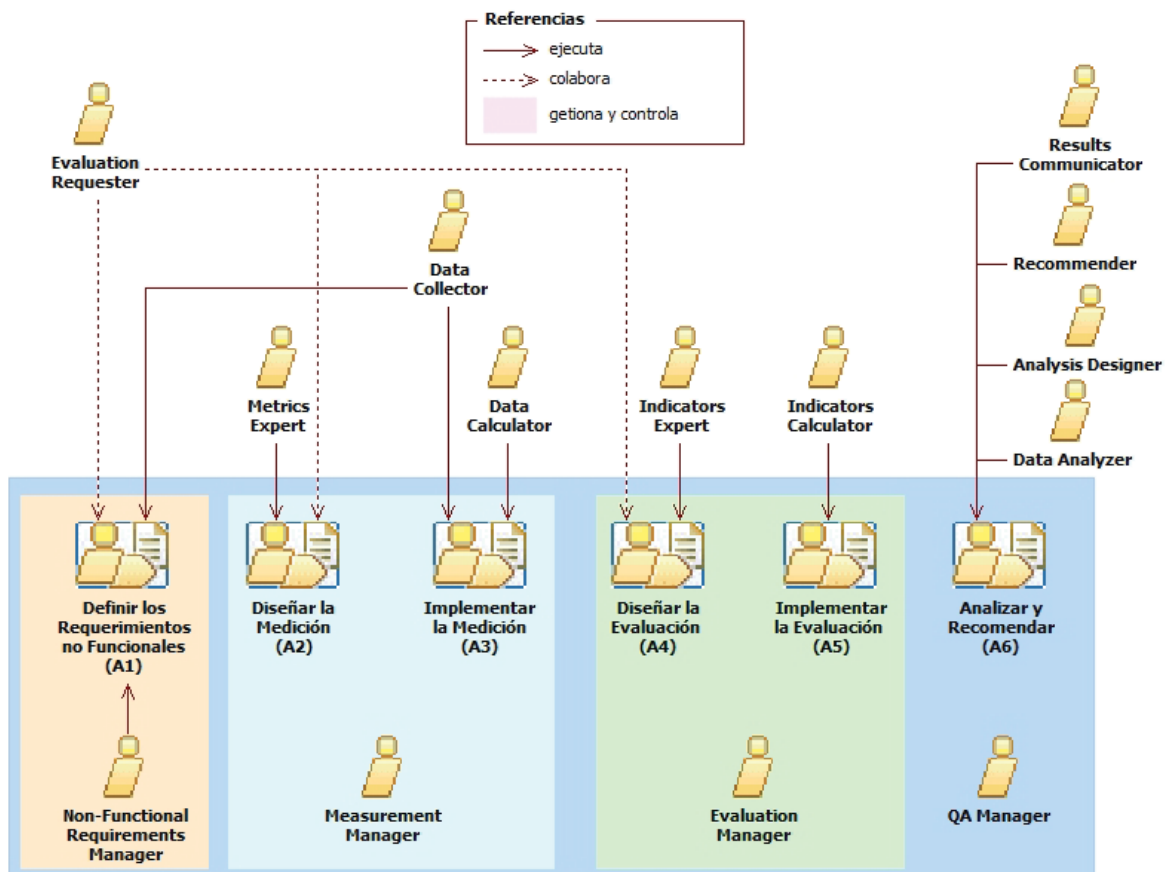


Figura 4-27: Vista organizacional del proceso de M&E de GOCAME.

En la Figura 4-27 se presenta la vista organizacional del proceso de M&E de GOCAME, asociando los diferentes roles a cada una de las actividades en las que tiene competencia. Notar que en dicha figura se utilizan flechas de línea entera, punteada y recuadros de colores. La flecha de línea entera significa que el rol

asociado lleva a cabo las tareas de la actividad; la flecha punteada representa que el rol involucrado colabora en las tareas de la actividad; y los recuadros de colores muestran las actividades sobre las cuales el rol tiene funciones de gestión y control. Por otro lado, es importante recordar que un rol puede ser cumplido por agentes humanos o automatizados. Por ejemplo, la medición de un atributo puede ser realizada por un agente automatizado en cumplimiento del rol *Data Collector*. Además, un rol pudiera ser cubierto por un grupo de personas. Por ejemplo, el rol *Evaluation Requester* pudiera ser una persona o una organización.

El rol *Quality Assurance (QA) Manager* es quien se encarga de crear el Proyecto de M&E, asignar los diferentes recursos necesarios para llevar adelante la M&E y de supervisar las diferentes actividades que se realizan durante el proceso de M&E. Sin embargo este rol no participa directamente en las actividades descritas en la Sección 4.4.1.

Por otro lado, como se mencionó en la sección 4.2.1, un Proyecto de M&E se divide en tres sub-proyectos. El primer sub-proyecto es el de Requerimientos No Funcionales, el cual está bajo la responsabilidad del *Non Functional Requirements Manager*. Este rol se encarga de llevar a cabo las diferentes tareas de la actividad A1. Es importante mencionar que, en la Figura 4-27, el rol *Evaluation Requester* también aparece asociado a la actividad A1 (así como a A2 y A4). Sin embargo, este rol no es quien ejecuta las tareas de la actividad en cuestión, sino más bien es quien colabora con los roles ejecutores. Por ejemplo, el *Evaluation Requester* es quien conoce las *Necesidades y Objetivos Organizacionales*, y por lo tanto colabora proveyendo la información necesaria para que el *NFR Manager* pueda producir la *Especificación de los Requerimientos No Funcionales* en A1. La colaboración del *Evaluation Requester* en las diferentes actividades se ve reflejado en la Figura 4-27 mediante una relación de línea punteada.

Las actividades A2 y A3 se realizan en el marco del sub-proyecto de Medición, cuyo responsable es el *Measurement Manager*. Particularmente, la actividad A2 la lleva a cabo el *Metrics Expert*. Quien posea este rol debe tener vasto conocimiento en el área de medición y contar con experiencia en el diseño y selección de métricas, lo cual le permitirá escoger las métricas y las herramientas de medición más adecuadas para cuantificar los atributos del árbol de requerimientos. Es importante la participación del *Evaluation Requester* en la actividad A2, ya que éste puede aportar información relevante para que el *Metrics Expert* seleccione las métricas más acordes a las necesidades del proyecto.

En cuanto a la actividad A3, se distinguen dos roles principales: el *Data Collector* y el *Data Calculator*. El primero se encarga de producir las medidas base haciendo uso de las métricas directas, mientras que el segundo produce las medidas derivadas mediante las métricas indirectas. Estos dos roles deben contar con conocimiento en el uso de herramientas de medición y pueden ser cumplidos por un mismo agente. Además, el *Data Collector* también se encarga de cuantificar las propiedades de contexto en la actividad A1.

Respecto de las actividades A4 y A5, estas se llevan a cabo bajo el sub-proyecto de Evaluación, cuyo responsable es el *Evaluation Manager*. La actividad A4 es ejecutada por el *Indicators Expert*, cuya responsabilidad es diseñar adecuadamente los indicadores elementales y derivados que permitan conocer el grado de satisfacción alcanzado por cada uno de los elementos del árbol de

requerimientos. En esta actividad, el *Evaluation Requester* también debe participar, por ejemplo, ayudando a definir los criterios de aceptación asociados a los indicadores. Por otro lado, la actividad A5 la ejecuta el *Indicators Calculator*, quien se encarga de producir los valores de indicadores.

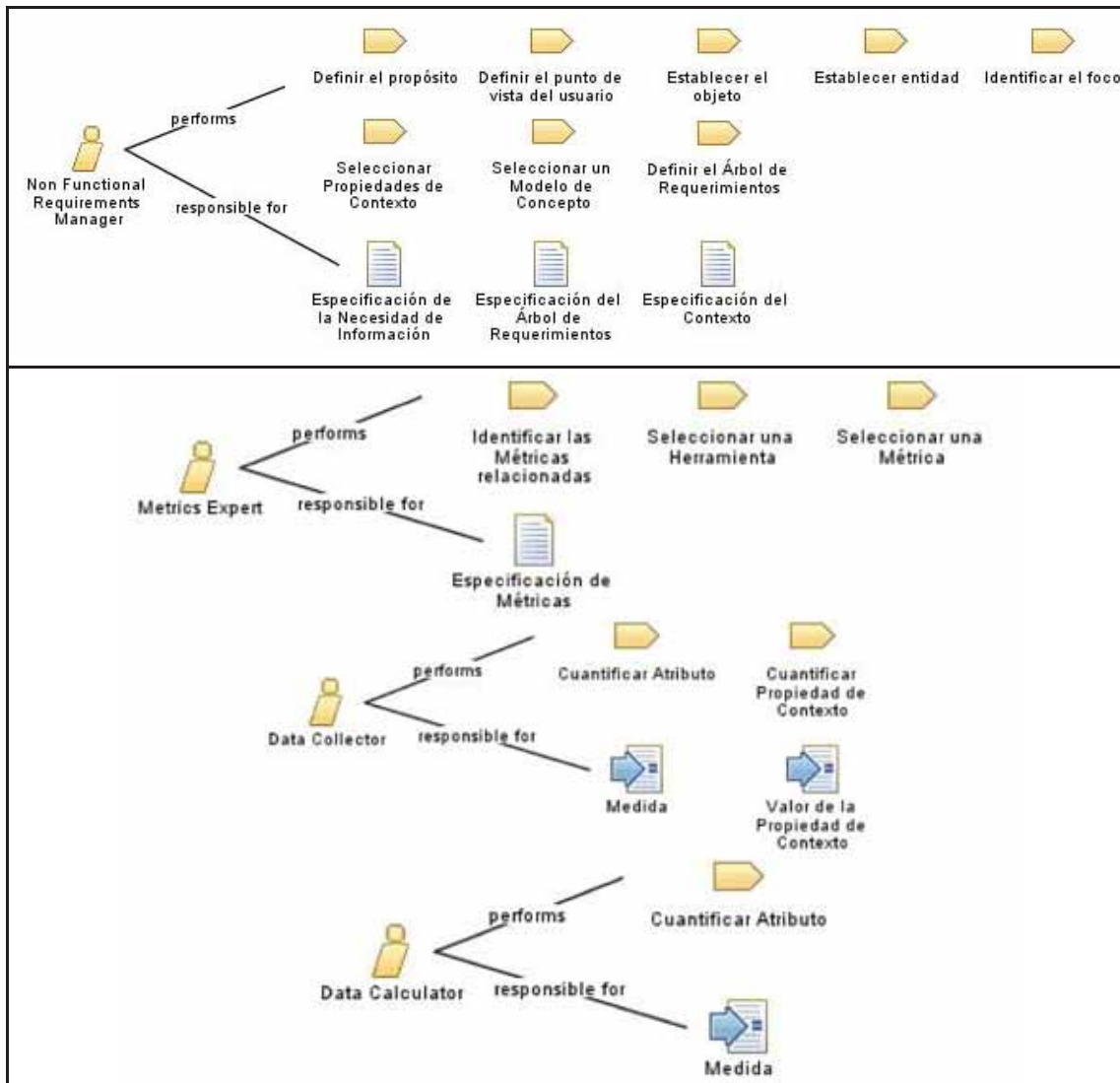


Figura 4-28: Roles de las actividades A1, A2 y A3, asociados a las tareas que ejecutan y los productos de trabajo por los cuales son responsables.

Por último, la actividad A6 involucra los siguientes roles: *Analysis Designer*, *Data Analyzer*, *Recommender* y *Results Communicator*. El *Analysis Designer* se encarga de identificar los métodos y técnicas apropiadas para luego analizar los datos que fueron producidos a través de las métricas y de los indicadores. Este rol debe tener conocimientos de métodos matemáticos y estadísticos, como así también de herramientas de documentación y visualización. En cuanto al rol *Data Analyzer*, este es el responsable de llevar a cabo el análisis de los datos producidos en la medición y en la evaluación. El *Data Analyzer*, también debe poseer conocimientos matemáticos y estadísticos a fin de poder seguir el documento *Diseño del análisis* provisto por el *Analysis Designer*.

Teniendo en cuenta que en la actividad A6 pueden existir tareas de recomendación, se ha incluido un rol denominado *Recommender*. Quien posea este rol debe contar con experiencia en el área de Aseguramiento de Calidad para así

proveer recomendaciones oportunas para aumentar la calidad de las entidades evaluadas. Por último, el *Results Communicator* se encarga de producir el *Informe de conclusión* y comunicar los resultados y recomendaciones al *Evaluation Requester*.

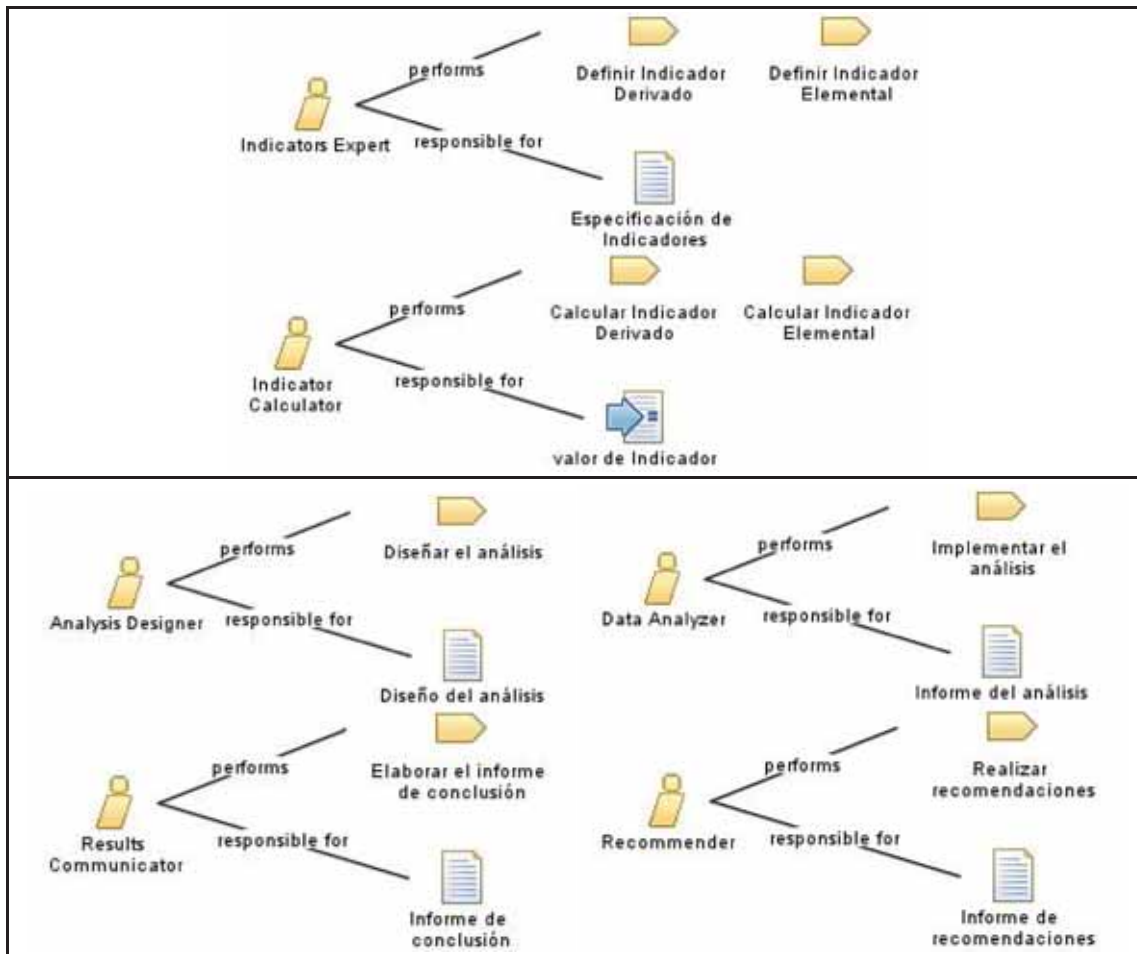


Figura 4-29: Roles de las actividades A4, A5 y A6, asociados a las tareas que ejecutan y los productos de trabajo por los cuales son responsables.

Finalmente, si bien la Figura 4-27 exhibe los roles involucrados en cada actividad, esta no permite identificar claramente qué tareas debe ejecutar cada rol. Para cubrir esta cuestión se presentan las Figuras 4-28 y 4-29¹³, las cuales muestran específicamente las tareas que debe ejecutar cada rol (ver relación “performs” en las figuras). A su vez, las figuras mencionadas también permiten observar los productos de trabajo que debe producir cada rol (relación “responsible for”). Cabe mencionar que los roles *QA Manager*, *Measurement Manager* y *Evaluation Manager* no aparecen en estas figuras. La razón es que no intervienen directamente en la ejecución de las tareas del proceso de M&E. Más bien, sus tareas están relacionadas a supervisar el trabajo de otros roles, y se encargan de tareas de gestión de sus respectivos proyectos, como puede ser la asignación de recursos. Estas tareas no forman parte del alcance de esta tesis.

4.4.3 Productos de Trabajo de GOCAME: Vista Informativa

Si bien en la sección 4.4.1 se mostraron ejemplos de los diferentes productos de

¹³ Los diagramas de estas figuras son generados automáticamente por la herramienta EPF Composer [Eclipse] a partir de la definición de las tareas, roles y productos de trabajo.

trabajo producidos a lo largo del proceso de M&E de GOCAME, en esta sección se modela la estructura de los mismos, proveyendo así una vista informacional.

Observando la Figura 4-3 (presentada en la Sección 4.4.1) se puede observar que los artefactos producidos son la *Especificación de los Requerimientos No Funcionales* (en A1), la *Especificación de Métricas* (en A2), la *Especificación de Indicadores* (en A4) y los *Informes de Conclusión y de Recomendaciones* (en A6). Por otro lado, las actividades A3 y A5 no producen artefactos sino datos o resultados (recordar el concepto *Outcome* en sección 3.3.1), a saber: medidas y valores de indicadores. Estas salidas, junto con sus metadatos asociados, son almacenados en repositorios (claramente identificados con el estereotipo *datastore* en la Figura 4-3 y otras). En esta tesis el alcance de la vista informacional está limitado a la estructura de los artefactos producidos y no de los datos (para este detalle se puede consultar el modelo C-INCAMI en la Sección 4.2). Teniendo esto presente, a continuación se comentan los artefactos producidos en A1, A2, A4 y A6.

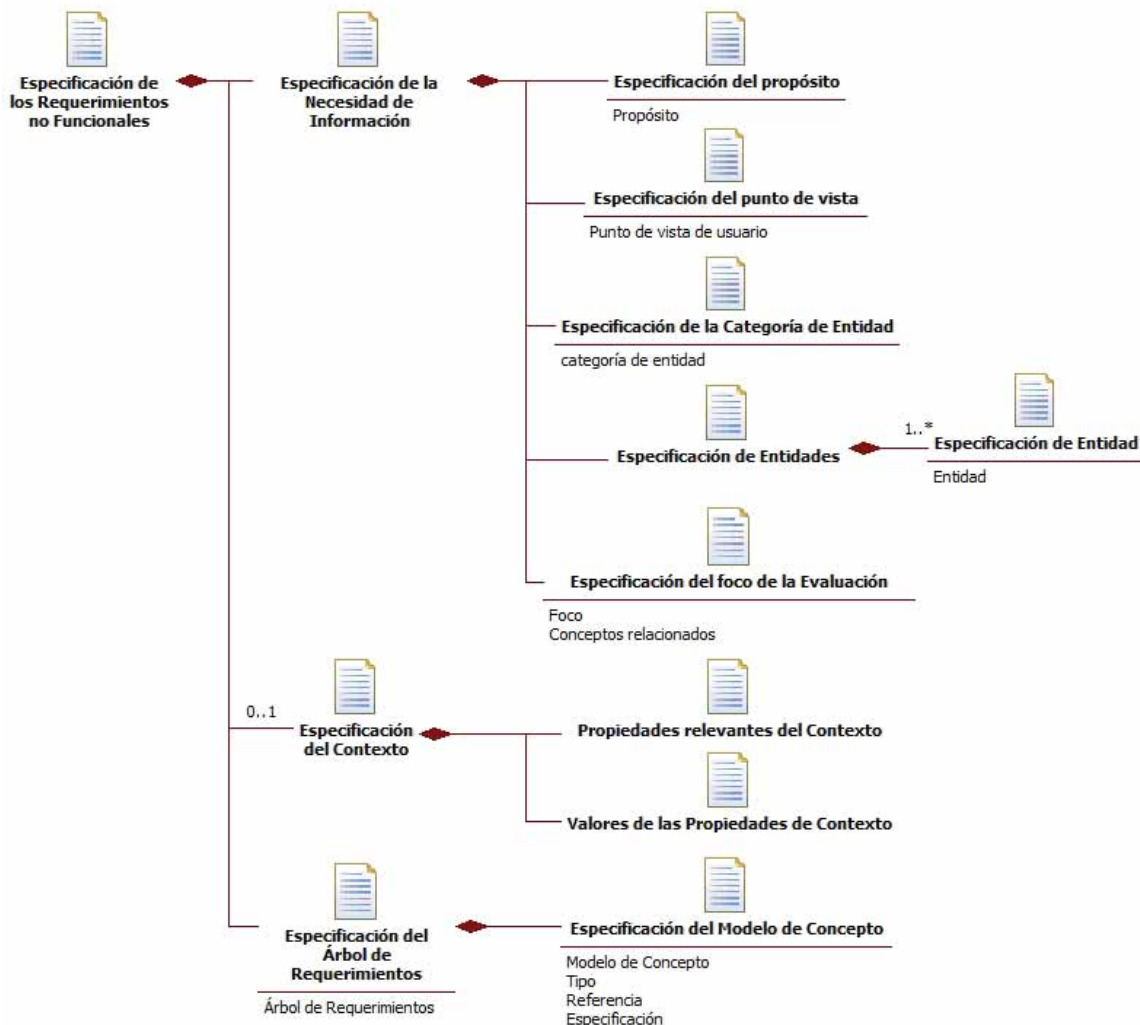


Figura 4-30: Composición del artefacto *Especificación de los Requerimientos No Funcionales*.

En la Figura 4-30 se presenta la estructura de la *Especificación de los Requerimientos No Funcionales*. En la misma se puede apreciar que este artefacto se compone de otros artefactos, a saber: una *Especificación de la Necesidad de Información*, una o ninguna *Especificación del Contexto* y una *Especificación del*

Árbol de Requerimientos. La cardinalidad (0..1) en el extremo de la relación entre la *Especificación de los Requerimientos No Funcionales* y la *Especificación del Contexto* se debe a que la actividad Definir el Contexto es opcional (ver la Figura 4-4). De manera similar, el artefacto *Especificación de Entidades* se compone de una o más *Especificaciones de Entidad*, ya que el proceso de M&E de GOCAME permite definir más de una entidad a evaluar, por ejemplo para propósitos de comparación.

Para un ejemplo de una *Especificación de los Requerimientos No Funcionales* ver la Figura 4-10. Notar que en dicha figura, los artefactos *Especificación de la Necesidad de Información*, *Especificación del Contexto* y *Especificación del Árbol de Requerimientos* aparecen como secciones del documento *Especificación de los Requerimientos No Funcionales*.

La *Especificación de Métricas* producida en A2 se compone de un conjunto de *Especificaciones de Métrica*, tal como se desprende de la Figura 4-31. En el caso de las métricas indirectas (ver sección 4.2.4) la *Especificación de Métrica* poseerá una *Especificación de Métricas relacionadas*, mientras que cuando se trate de métricas directas, no. A su vez, una *Especificación de Métricas relacionadas* se compone de un conjunto de *Especificaciones de Métrica*. En la Figura 4-13 se presenta un ejemplo de una *Especificación de Métrica*.

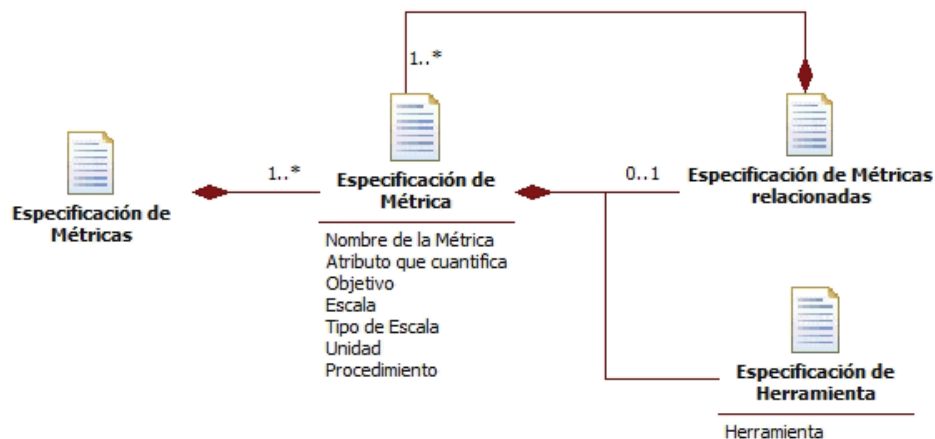


Figura 4-31: Composición del artefacto *Especificación de Métricas*.

Respecto de la *Especificación de Indicadores*, esta se compone de la *Especificación de Indicadores Elementales* y de la *Especificación de Indicadores Derivados*. A su vez, la *Especificación de Indicadores Elementales* se compone de un conjunto de *Especificaciones de Indicador Elemental* y la *Especificación de Indicadores Derivados* se conforma de *Especificaciones de Indicador Derivado*, como se observa en la Figura 4-32. Un ejemplo de una *Especificación de Indicadores* se observa en la Figura 4-20.

Por último, los *Informes de Conclusiones y de Recomendaciones* no tienen una estructura particular. Los mismos pueden contener secciones de texto, como así también gráficos, diagramas y tablas, entre otros documentos asociados, que permitan comunicar eficazmente las conclusiones y recomendaciones de una manera resumida y entendible para los interesados.

Finalmente, cabe mencionar que los atributos que en las Figuras 4-30, 4-31 y 4-32 representan la información básica que deben contener los documentos. Notar también que los términos utilizados en la definición de los artefactos respetan la terminología proporcionada por el marco conceptual de M&E C-INCAMI.

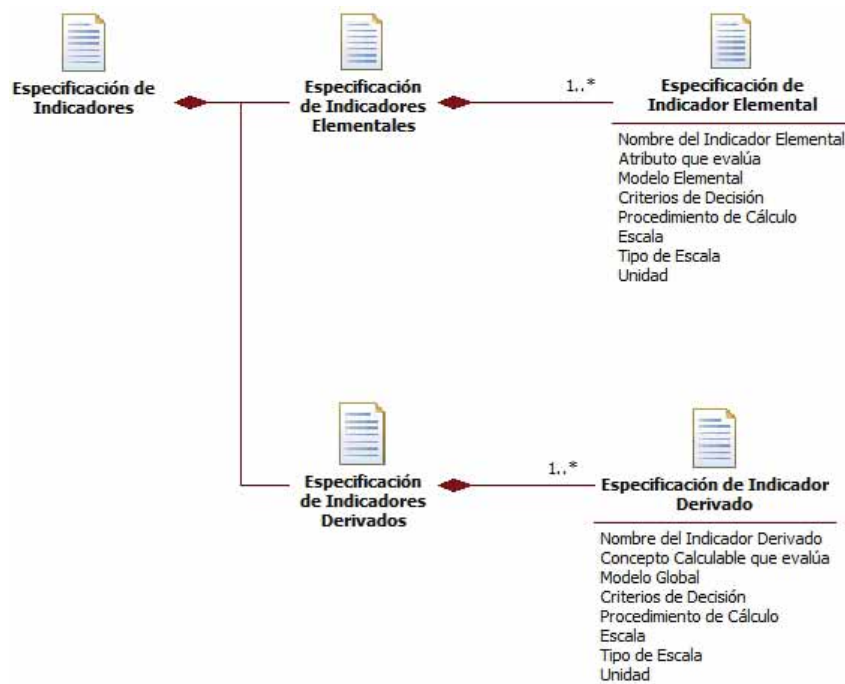


Figura 4-32: Composición del artefacto *Especificación de Indicadores*.

4.5. Métodos y Herramientas

En la Sección 4.4.1 se observa que existen tareas muy diversas que permiten especificar, reunir, registrar y usar datos y metadatos de propiedades de contexto, métricas e indicadores. Para llevar a cabo todas estas tareas es necesario contar con una metodología, es decir, un conjunto bien definido y cooperante de métodos, técnicas y herramientas que sean aplicados sistemáticamente a las distintas tareas del proceso. Con este propósito surgió la metodología WebQEM (*Web Quality Evaluation Method*) [Olsina, y otros, 2002], la cual da soporte al tercer pilar de la estrategia integrada de M&E GOCAME: los métodos y herramientas.

La metodología WebQEM aporta una estrategia eficaz, centrada en expertos, para evaluar y analizar la calidad de sitios y aplicaciones Web, o de productos software en general. Se basa en el uso de un modelo jerárquico de requerimientos de calidad (o árbol de requerimientos), donde las características de más alto nivel pueden ser por ejemplo aquellas prescriptas en la norma ISO 25000 u otras; y a partir de ellas se derivan subcaracterísticas de menor nivel de abstracción, para finalmente a estas últimas asociar atributos. La metodología permite evaluar y determinar el grado de satisfacción, o nivel de cumplimiento, de las características del modelo para un determinado producto software o Web.

Para llevar a cabo la medición (ver tarea *Cuantificar Atributo* de la actividad A5 en la Sección 4.4.1.3), los métodos utilizados son las métricas. Existen dos tipos de métricas: directas e indirectas (recordar componente de Medición en la Sección 4.2.4). La métrica a aplicar depende del atributo a medir, y la misma es escogida en la actividad A2. Los pasos concretos a seguir para realizar la medición del atributo se encuentran especificados en el procedimiento de medición o de cálculo de la métrica directa o indirecta, respectivamente. En la Figura 4-13 se observa una plantilla de especificación de una métrica, en la cual se define claramente el nombre de la métrica, el atributo que cuantifica, el procedimiento de cálculo que debe seguirse para obtener un valor para el atributo en cuestión, la escala y la

herramienta que automatiza el procedimiento, entre otros metadatos.

Por otro lado, la evaluación (ver tareas *Calcular Indicador Elemental* y *Calcular Indicador Derivado* de la actividad A5 en la Sección 4.4.1.5) se realiza mediante métodos llamados indicadores, los cuales pueden ser: elementales o derivados (recordar componente de Evaluación en la Sección 4.2.5). Los indicadores elementales especifican cómo transformar las medidas en valores que representen el nivel de satisfacción alcanzado por un atributo. En contraposición, los indicadores derivados indican cómo calcular el nivel de satisfacción alcanzado por los conceptos calculables haciendo uso de modelos de agregación de características, subcaracterísticas y atributos (como Modelo de Puntaje Lineal Aditivo, Modelo de Puntaje de Preferencia Lógica, entre otros).

Al igual que las métricas indirectas, los indicadores cuentan con un procedimiento de cálculo en el cual se especifican los pasos que se deben seguir para obtener un valor de indicador. Asimismo, los indicadores tienen asociados criterios de decisión que indican cómo debe interpretarse el valor de indicador. Los indicadores a utilizar son escogidos en la actividad A4. En la Figura 4-18 se presenta una plantilla con la especificación de un indicador elemental, donde se define el nombre del indicador, el atributo que evalúa, el modelo elemental y el procedimiento de cálculo que debe seguirse para obtener un valor de indicador, la escala y los criterios de decisión que permiten interpretar el valor obtenido, entre otros metadatos. Notar que los metadatos especificados para los indicadores, como así también los relacionados a las métricas, provienen del marco conceptual C-INCAMI. Esto permite diseñar, almacenar y utilizar de una manera consistente los métodos, particularmente las métricas e indicadores, a lo largo del tiempo.

Además, con el objetivo de brindar soporte tecnológico al proceso de M&E, en el año 2005 se desarrolló la aplicación Web denominada INCAMI_Tool [Molina, 2005; Papa, 2005], la cual a su vez extendía a una herramienta previa llamada WebQEM_Tool [Olsina L. , y otros, 2001]. Para que INCAMI_Tool siga estrictamente el proceso de M&E aquí presentado la misma ha sido adaptada. Actualmente la aplicación es llamada C-INCAMI_Tool [Molina, 2012; Becker, y otros, 2012], ya que ahora también provee la posibilidad de guardar información referente al contexto en el cual se enmarca el proyecto de M&E.

Por un lado, la aplicación sirve de guía al usuario para llevar a cabo aquellas tareas involucradas en la creación de proyectos, la definición y especificación de requerimientos no funcionales, información de contexto y diseño e implementación de la medición y la evaluación, y la generación de informes de los resultados. La aplicación tiene en cuenta el flujo de actividades del proceso, por lo tanto sólo se presentan al usuario aquellas actividades que son posibles realizar en base a las tareas completadas, haciendo cumplir así las interdependencias. A lo largo del proceso los diferentes artefactos producidos son almacenados, para ser consultados en las actividades que los requieran como entrada. Por otro lado, C-INCAMI_Tool también permite especificar un conjunto de datos y metadatos necesarios para asegurar que los valores sean repetibles y comparables, de forma consistente, entre proyectos, basándose en el marco C-INCAMI. En la Figura 4-33 se muestran algunas capturas de pantalla de la herramienta web C-INCAMI_Tool.

La aplicación web interactúa con un repositorio semántico que contiene las instancias de los conceptos básicos, como por ejemplo atributos, métricas y el

conjunto de objetos que la definen (escala, función, procedimiento de medición o de cálculo, etc.), categorías de entidad, entre otros, que serán compartidos por todos los proyectos de M&E de la organización. De esta forma se establece una base homogénea para todos los proyectos, permitiendo efectuar comparaciones de forma consistente, y la reutilización de instancias de conceptos. Este repositorio es mantenido por un sistema de revisión [Baffini, y otros, 2006; Rivera, y otros, 2007].

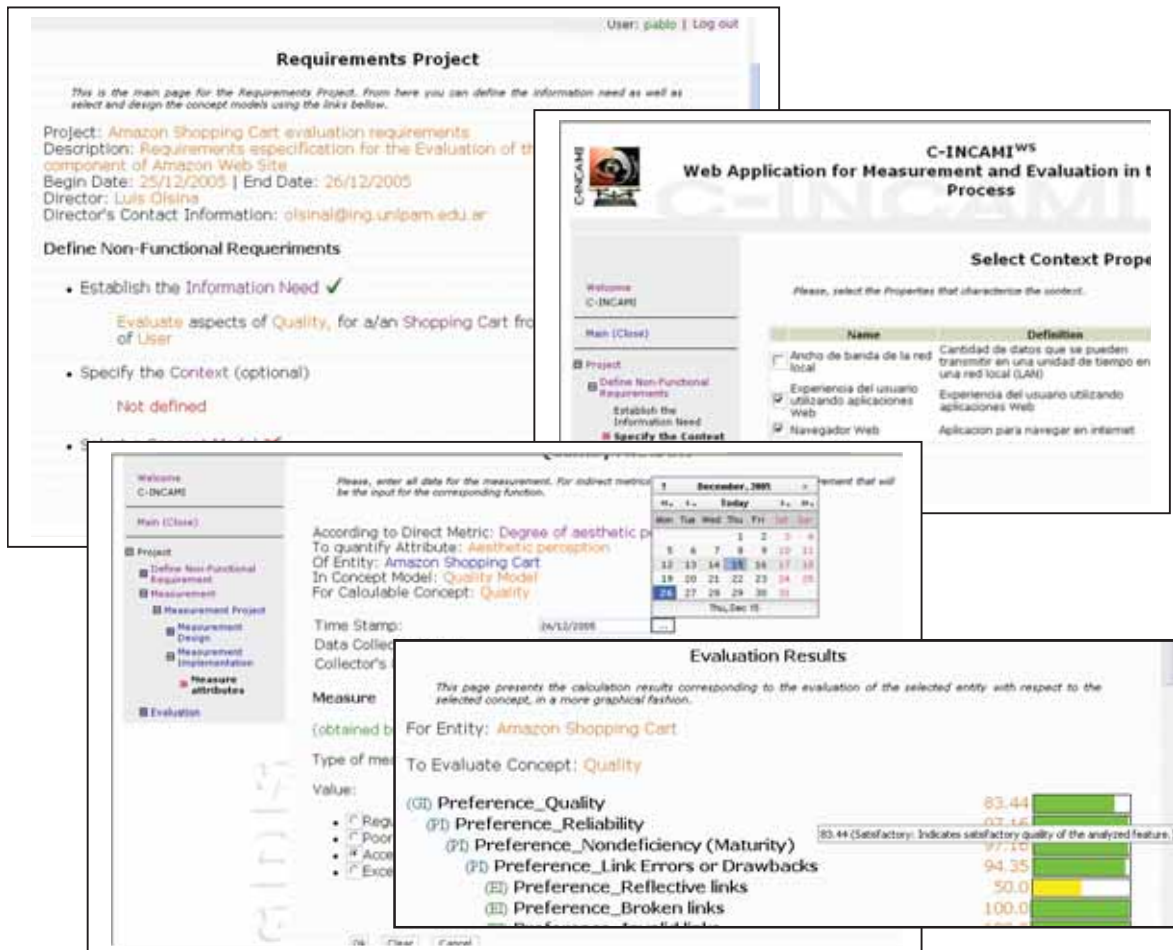


Figura 4-33: Capturas de pantalla de la herramienta C-INCAMI_Tool.

La aplicación web se basa en una arquitectura multicapa, tal como se indica en la Figura 4-34. La capa de presentación consiste de una interfaz web compuesta por un conjunto de páginas dinámicas, que guían al usuario a lo largo del proceso. En la capa de lógica de negocio se encuentran aquellos componentes que permiten llevar a cabo las tareas. Están incluidas tanto las clases de negocio (correspondientes al marco C-INCAMI) que implementan la lógica inherente a cada concepto, así como las clases que implementan la lógica relacionada a la definición y especificación de requerimientos no funcionales, y de diseño e implementación de la medición y la evaluación. Los componentes de esta capa hacen uso de la capa de acceso a datos para ofrecer servicios de persistencia.

En la capa de acceso a datos se encuentran los componentes que ofrecen el servicio de escribir y leer, hacia y desde un medio de almacenamiento, datos relevantes para la aplicación. El objetivo de esta capa es ocultar la complejidad de la manipulación de tecnologías para almacenamiento persistente de información, necesaria para recuperar y almacenar los objetos de datos manipulados por el

sistema. La capa de persistencia se compone de las tecnologías necesarias para almacenar de forma persistente los datos necesarios para la aplicación.

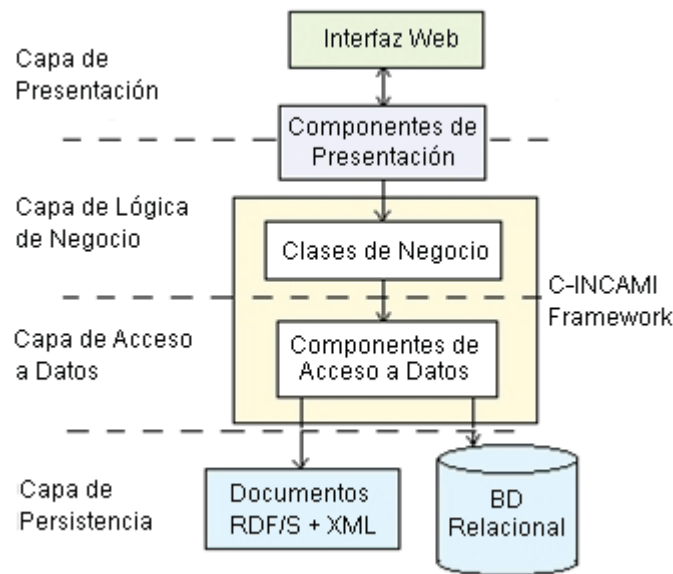


Figura 4-34: Arquitectura de la aplicación C-INCAMI_Tool.

4.6. Conclusiones

En este capítulo se ha presentado GOCAME, una estrategia integrada de M&E multi-propósito y se han comentado sus tres capacidades fundamentales, a saber: i) el marco conceptual de M&E, con base ontológica, denominado C-INCAMI (Sección 4.2), ii) la especificación del proceso de M&E teniendo en cuenta diferentes vistas (Sección 4.4), y iii) la metodología WebQEM junto a la herramienta C-INCAMI_Tool que permiten llevar a cabo las actividades del proceso (Sección 4.5).

Si bien se han tenido en cuenta las tres capacidades y la manera en cómo estas se relacionan, principalmente se ha puesto énfasis en la especificación de las vistas del proceso. Contar con la especificación de distintas vistas del proceso de M&E de GOCAME es uno de los aportes de este trabajo. En la literatura revisada se ha observado que los enfoques de M&E examinados no tienen en cuenta las diferentes vistas del proceso (recordar Sección 3.2). Por ejemplo, en los estándares ISO solo se hace uso de la vista funcional a nivel global y en el estándar *de facto* CMMI directamente no se presenta ningún modelo de proceso. Teniendo presente la importancia de contar con procesos bien definidos para que los agentes puedan lograr los objetivos perseguidos, esto es, que se pueda identificar claramente las actividades/tareas que deben llevarse a cabo, así como la secuencia de las mismas y entradas y salidas, entre otros aspectos, se ha modelado el proceso de M&E de GOCAME desde diferentes vistas o perspectivas.

Particularmente, se han tenido en cuenta las vistas funcional y de comportamiento, que indican las actividades/tareas que deben llevarse a cabo, así como el orden en que deben ejecutarse y los productos de trabajo que sirven de entrada y/o son producidos a lo largo del proceso. También se ha incluido la vista organizacional con el fin de mostrar los roles que están involucrados en el proceso y se ha presentado la vista informacional, mostrando la estructura de cada uno de los diferentes artefactos, así como ejemplos que sirven de plantillas de los mismos.

Modelar el proceso de M&E de GOCAME desde estas vistas proporciona una visión completa, detallada e integrada del proceso. Esto permite contar con una guía clara y repetible de las actividades/tareas que deben llevarse a cabo para definir los requerimientos no funcionales, establecer el contexto y diseñar e implementar la medición, la evaluación, y el análisis y la recomendación, con el fin de soportar un proceso de toma de decisiones más robusto. También facilita la comunicación y la realización de las tareas de las partes intervinientes, como así también la selección del personal asociado a las actividades en función del rol que deben cumplir, entre otros beneficios. Otro aspecto a tener presente es que la terminología empleada en las vistas (modelos) del proceso de GOCAME es consistente gracias al uso de un marco conceptual de M&E. El marco empleado es C-INCAMI, y fue vinculado a una ontología de procesos para proveerle mayor riqueza semántica (Sección 4.3).

Finalmente, cabe mencionar también que la estrategia GOCAME ha sido utilizada como una estrategia base que permite definir nuevas estrategias de M&E. Por ejemplo, a partir de GOCAME se ha desarrollado SIQinU (*Strategy to Improve Quality in Use*). Esta estrategia se describe en el siguiente capítulo.

Capítulo 5: Estrategia SIQinU

En este capítulo se presenta la estrategia de M&E de propósito específico llamada SIQinU. Se abordan sus tres capacidades y se resalta la especificación del proceso. Además se muestra la alineación existente entre SIQinU y GOCAME.

Por otra parte, se propone una plantilla basada en el principio W5H y el marco conceptual de proceso que permite especificar de manera completa las actividades.

5.1. Introducción

Un objetivo constante de cualquier organización de Ingeniería de Software debe ser la mejora continua de la calidad de sus productos y servicios [Software Engineering Laboratory, 1995]. Las organizaciones de desarrollo de software/web reconocen que la M&E es un pre-requisito para llevar adelante procesos sistemáticos de mejora [Lott, y otros, 1995]. La estrategia de M&E GOCAME provee los mecanismos que permiten conocer la situación actual de las entidades evaluadas. Es decir, es como si se tratara de una instantánea que muestra, por ejemplo, el nivel de satisfacción alcanzado por una entidad respecto de su Calidad Interna, Externa¹⁴ o en Uso. Sin embargo, GOCAME no incluye actividades/tareas y métodos para incrementar la calidad de las entidades evaluadas. Con el fin de dar soporte a esta cuestión, se ha desarrollado una nueva estrategia de M&E llamada SIQinU (*Strategy to Improve Quality in Use*) [Lew, y otros, 2012].

La estrategia SIQinU se fundamenta en las relaciones existentes entre la Calidad Externa y la Calidad en Uso. Específicamente, la Calidad Externa “influye” a la Calidad en Uso y, a su vez, la Calidad en Uso “depende de” o está determinada por la Calidad Externa [ISO/IEC, 2011]. Teniendo presente estas relaciones, SIQinU busca mejorar, de manera incremental y continua, la Calidad en Uso de una aplicación web (de ahora en adelante denominada WebApp) a partir de la mejora de la Calidad Externa. Cabe mencionar que, si bien SIQinU fue pensada para aplicarse sobre una WebApp, esta estrategia también puede extenderse a aplicaciones software en general.

Básicamente, SIQinU evalúa la Calidad en Uso para detectar problemas de uso reales, mapearlos a atributos de Calidad Externa (teniendo presente la relación “depende de”) y, luego, aplicar acciones de mejora sobre estos atributos que, posiblemente, impactarán positivamente sobre la Calidad en Uso (relación “influye”), reduciendo así los problemas de uso detectados. Finalmente, se vuelve a evaluar la Calidad en Uso de la WebApp para observar la ganancia lograda.

¹⁴Actualmente, ISO 25010 no utiliza de manera explícita el término Calidad Externa, como así tampoco Calidad Interna. En su lugar hace uso del término Calidad de Producto [ISO/IEC, 2011]. Sin embargo, el término Calidad Externa aun sigue siendo empleado en trabajos recientes. Ejemplos de ello son [Lew, y otros, 2013] y [METI, 2011]. Por lo tanto, en el presente trabajo, se seguirá usando el término Calidad Externa por ser el término más difundido y aceptado.

La estrategia SIQinU se deriva de GOCAME, por lo tanto comparte similitudes con esta última en cuanto a sus tres capacidades. Por ejemplo, SIQinU reutiliza muchos de los elementos del proceso, el marco conceptual de M&E y algunos métodos y herramientas de GOCAME (ver detalles en la Sección 5.3). Sin embargo, SIQinU también posee diferencias respecto de GOCAME. Por ejemplo, es una estrategia de propósito específico porque únicamente está pensada para “conocer” y “mejorar” la Calidad en Uso. Además cuenta con actividades/tareas de mejora, y métodos específicos para llevarlas a cabo.

Otra diferencia notable es que SIQinU propone un proceso con ciclos de mejora, así como de re-evaluación (tanto de la Calidad en Uso como de la Calidad Externa) para observar la ganancia lograda en cada iteración (ver Figura 5-1).

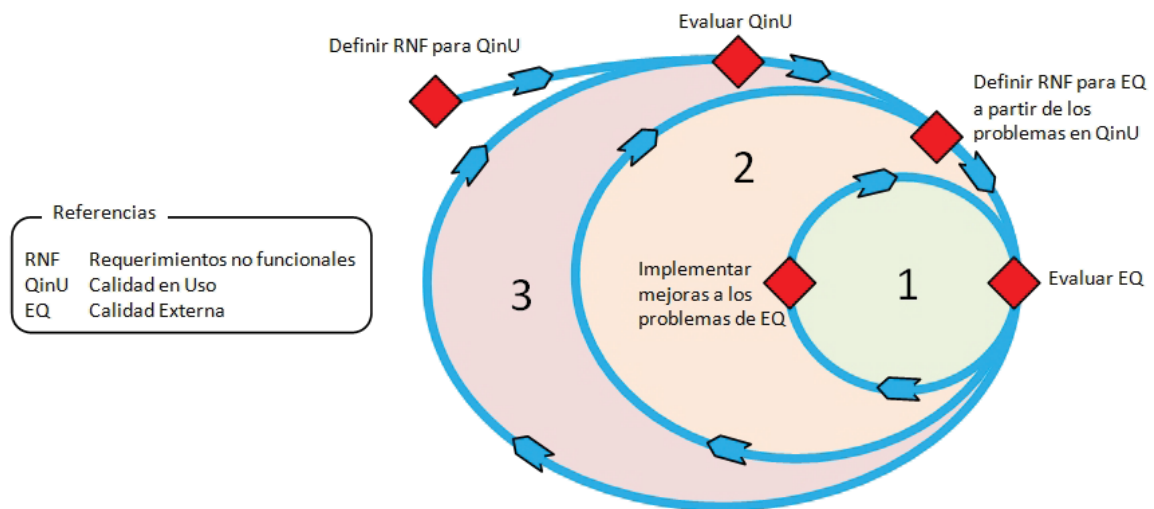


Figura 5-1: Ciclos del proceso de SIQinU: (1) mejora del sistema, (2) re-evaluación de Calidad Externa, y (3) re-evaluación de Calidad en Uso.

Según el proceso de SIQinU, en primer lugar se definen los requerimientos no funcionales para Calidad en Uso y se evalúa el sistema en uso. Considerando los problemas detectados en Calidad en Uso, a continuación se definen los requerimientos no funcionales para Calidad Externa y se evalúan. A partir de esta evaluación se detectan problemas de Calidad Externa y se recomiendan acciones de mejora. El equipo de desarrollo implementa en el sistema las mejoras propuestas y se vuelve a evaluar la Calidad Externa (ciclo 1 según Figura 5-1) para conocer la ganancia obtenida a partir de los cambios realizados. En este punto, se puede volver a proponer nuevas mejoras a ser implementadas o volver a definir nuevos requerimientos no funcionales para Calidad Externa (ciclo 2) y así descubrir otros problemas de Calidad Externa a solucionar que permitan incrementar la Calidad en Uso. Una vez solucionado los problemas de Calidad Externa encontrados, se evalúa nuevamente la Calidad en Uso (ciclo 3) para observar la ganancia obtenida en la Calidad en Uso gracias a las mejoras realizadas en la Calidad Externa. En el caso de encontrar nuevos problemas o de persistir los previamente encontrados en el sistema en uso, se puede volver a derivar y definir nuevos requerimientos no funcionales para Calidad Externa, y seguir con el proceso de mejora y re-evaluación.

La mayoría de los estudios típicos de *testing* de usabilidad suelen ser subjetivos, no repetibles, consumen tiempo y a veces no son confiables debido a que los participantes cambian su comportamiento al saber que están siendo

observados [Nielsen, y otros, 1994]. Una característica distintiva de la estrategia SIQinU es que puede ser implementada de una manera económica y sistemática, reduciendo así los problemas mencionados, al aprovechar las capacidades del lado del servidor para registrar datos de uso (comportamiento) reales del usuario – agregando, por ejemplo, trozos de código en la WebApp y conseguir los datos necesarios para obtener las medidas y calcular los indicadores para Calidad en Uso de una manera no intrusiva.

A continuación se presenta en detalle el proceso explícito de SIQinU, el cual fue intuitivamente esquematizado en la Figura 5-1. Luego se comenta la alineación, desde la perspectiva de sus capacidades, entre las estrategias GOCAME y SIQinU (Sección 5.3). Finalmente, se presenta una plantilla de especificación de procesos (Sección 5.4).

5.2. Proceso de SIQinU

El proceso que subyace en SIQinU consta de seis fases, las cuales se denominan¹⁵:

(Ph I) *Fase I: Specify Requirements and Evaluation Criteria for QinU*

(Ph II) *Fase II: Perform QinU Evaluation and Analysis*

(Ph III) *Fase III: Derive/Specify Requirements and Evaluation Criteria for EQ*

(Ph IV) *Fase IV: Perform EQ Evaluation and Analysis*

(Ph V) *Fase V: Recommend, Perform Improvement Actions and Re-evaluate EQ*

(Ph VI) *Fase VI: Re-evaluate Quality in Use and Analyze Improvement Actions*

En la Figura 5-2 se presenta la vista de comportamiento en la cual pueden observarse las fases mencionadas, así como las actividades que éstas involucran. Además, se aprecia el orden en que deben llevarse a cabo las actividades y las iteraciones posibles.

La Fase I (*Specify Requirements and Evaluation Criteria for QinU*) implica definir los requerimientos no funcionales (características y atributos) relacionados a Calidad en Uso que serán evaluados. Además, esta fase involucra diseñar las tareas que llevan a cabo los usuarios a través de la WebApp, definir el tipo de usuario, especificar el contexto de uso e identificar las métricas e indicadores a utilizar, entre otras actividades. A continuación, en la Fase II (*Perform QinU Evaluation and Conduct Preliminary Analysis*), se obtienen las medidas y valores de indicadores que permitirán conocer el nivel de satisfacción alcanzado por la WebApp respecto a su Calidad en Uso. Es importante destacar que es deseable que la WebApp cuente con capacidades para registrar ciertos datos (como cantidad de errores, completitud y precisión en la realización de las tareas y sub-tareas) durante su uso, lo cual permitirá realizar mediciones de una manera no intrusiva para el usuario. A partir de las medidas y de los valores de indicadores se deben detectar los problemas de Calidad en Uso que deben ser resueltos con mayor urgencia.

Teniendo presente los problemas de Calidad en Uso, durante la Fase III (*Derive/ Specify Requirements and Evaluation Criteria for EQ*) se deben derivar requerimientos no funcionales relacionados a Calidad Externa. Estos

¹⁵ Los nombres de las fases, actividades y productos de trabajo asociados, así como muchos de los modelos de proceso de SIQinU, se presentan en inglés tal cual aparecen en las publicaciones originales [Becker, y otros, 2011], [Lew, y otros, 2012] y [Becker, y otros, 2012].

requerimientos deben surgir teniendo en cuenta la relación “depende de” que existe entre los atributos y características de Calidad en Uso y los de Calidad Externa. También, durante esta fase se deben identificar las métricas e indicadores a utilizar en la Fase IV (*Perform EQ Evaluation and Analysis*). Esta última fase involucra cuantificar los atributos usando las métricas y calcular los valores de indicadores, además de identificar las partes de la WebApp que necesitan ser mejoradas desde un enfoque de Calidad Externa.

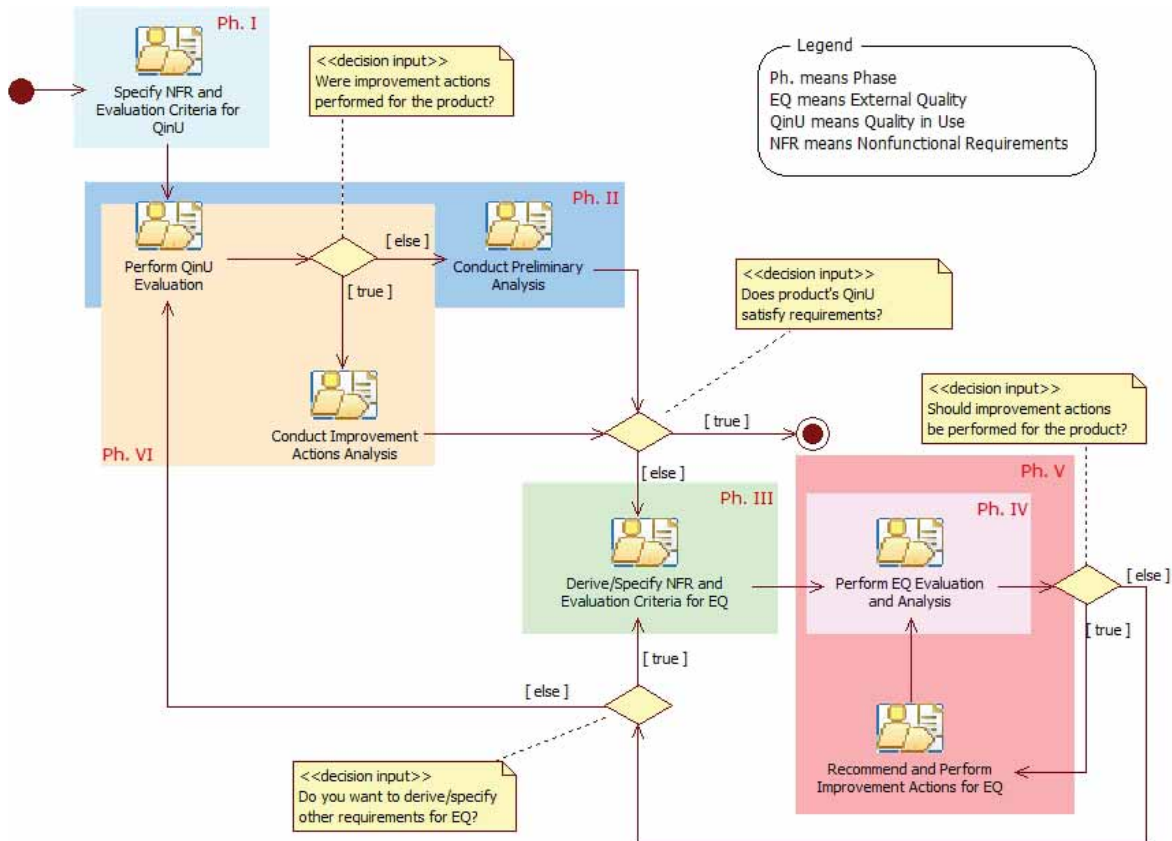


Figura 5-2: Vista de comportamiento del proceso de SIQinU.

A continuación, en base a los problemas detectados en Calidad Externa, durante la Fase V (*Recommend, Perform Improvement Actions, and Re-evaluate EQ*) se deben realizar acciones de mejora, es decir modificar la WebApp. Luego de introducir los cambios recomendados, se procede a re-evaluar la Calidad Externa con el objetivo de conocer la ganancia lograda (respecto al nivel alcanzado en la evaluación de la Fase IV) gracias a las acciones de mejora. Luego de la Fase V, puede suceder que sea necesario implementar nuevas mejoras en la WebApp (recordar ciclo 1 de la Figura 5-1) o que los evaluadores detecten que otros atributos de Calidad Externa deberían derivarse (recordar ciclo 2 de la Figura 5-1), bajo la premisa de que, si se mejoran estos nuevos atributos, se logrará mejorar también la Calidad en Uso (teniendo presente la relación “influye” que existe entre la Calidad Externa y la Calidad en Uso). Estos aspectos son tenidos en cuenta en el proceso de SIQinU, tal como se observa en la Figura 5-2.

Finalmente, y luego de que la nueva versión de la WebApp fue utilizada por usuarios reales, en un contexto real de uso, se lleva a cabo la Fase VI (*Re-evaluate QinU and Analyze Improvement Actions*). Ésta implica nuevamente evaluar la Calidad en Uso con el fin de conocer la ganancia lograda respecto de la obtenida en

la Fase II. En el caso de que el nivel de satisfacción logrado no alcance cierto nivel de aceptabilidad, nuevamente se podría continuar por la Fase III, realizar nuevas mejoras en Calidad Externa y así sucesivamente (recordar ciclo 3 de la Figura 5-1), hasta lograr el nivel de aceptabilidad requerido para Calidad en Uso.

A continuación se detallan las diferentes fases, teniendo en cuenta, principalmente, las vistas funcional y de comportamiento. Las vistas informacional y organizacional serán tratadas con menor detalle debido a que la mayoría de los productos de trabajo y roles son los mismos que en GOCAME.

5.2.1 Fase I: Specify Requirements and Evaluation Criteria for QinU (Ph I)

Las actividades de la primera fase de SIQinU están orientadas a la especificación y diseño de aspectos relacionados a la evaluación de la Calidad en Uso. Particularmente, las actividades a llevar a cabo durante la Fase (Ph) I son:

- ✓ *Establish Information Need*
- ✓ *Specify Project Context*
- ✓ *Design Tasks*
- ✓ *Establish QinU Requirements Tree*
- ✓ *Design QinU Measurement*
- ✓ *Design QinU Evaluation*
- ✓ *Design Preliminary Analysis*

Las vistas funcional y de comportamiento de Ph I se presentan en la Figura 5-3. Vale mencionar que las actividades de Ph I comprenden total o parcialmente a las actividades A1, A2, A4 y A6 de GOCAME. Por ejemplo, *Establish Information Need*, *Specify Project Context* y *Establish Requirements Tree* forman parte de A1 (recordar Figura 4-4) y *Design Preliminary Analysis* abarca algunas de las tareas de A6. Estas actividades se describieron en el Capítulo 4.

Si bien la mayoría de las actividades de Ph I pertenecen a GOCAME, se ha modificado el flujo entre las mismas. Por ejemplo, teniendo en cuenta la vista de comportamiento de la actividad A1 de GOCAME y la de Ph I de SIQinU, se puede apreciar que en GOCAME especificar información del contexto es opcional (ver actividad *Especificar el Contexto* en Figura 4-4) mientras que en Ph I es obligatoria (ver actividad *Specify Project Context* en Figura 5-3). Esto se debe a que GOCAME puede utilizarse para evaluar diferentes enfoques de calidad para los cuales la información del contexto puede no ser necesaria, sin embargo, teniendo presente que Ph I está enfocada únicamente en aspectos de Calidad en Uso, y que, según [ISO/IEC, 2011], la Calidad en Uso se define como “el grado en que un producto o sistema puede ser utilizado por usuarios específicos para satisfacer sus necesidades para lograr objetivos específicos con efectividad, eficiencia, libre de riesgo y satisfacción en *contextos de uso específicos*”, la actividad *Specify Project Context* es necesaria.

Además, considerando que la Calidad en Uso se evalúa en relación a ciertas tareas que debe realizar el usuario utilizando la aplicación, se ha agregado la actividad *Design Tasks*, no considerada en GOCAME. A continuación se detallan las 7 actividades principales de Ph I.

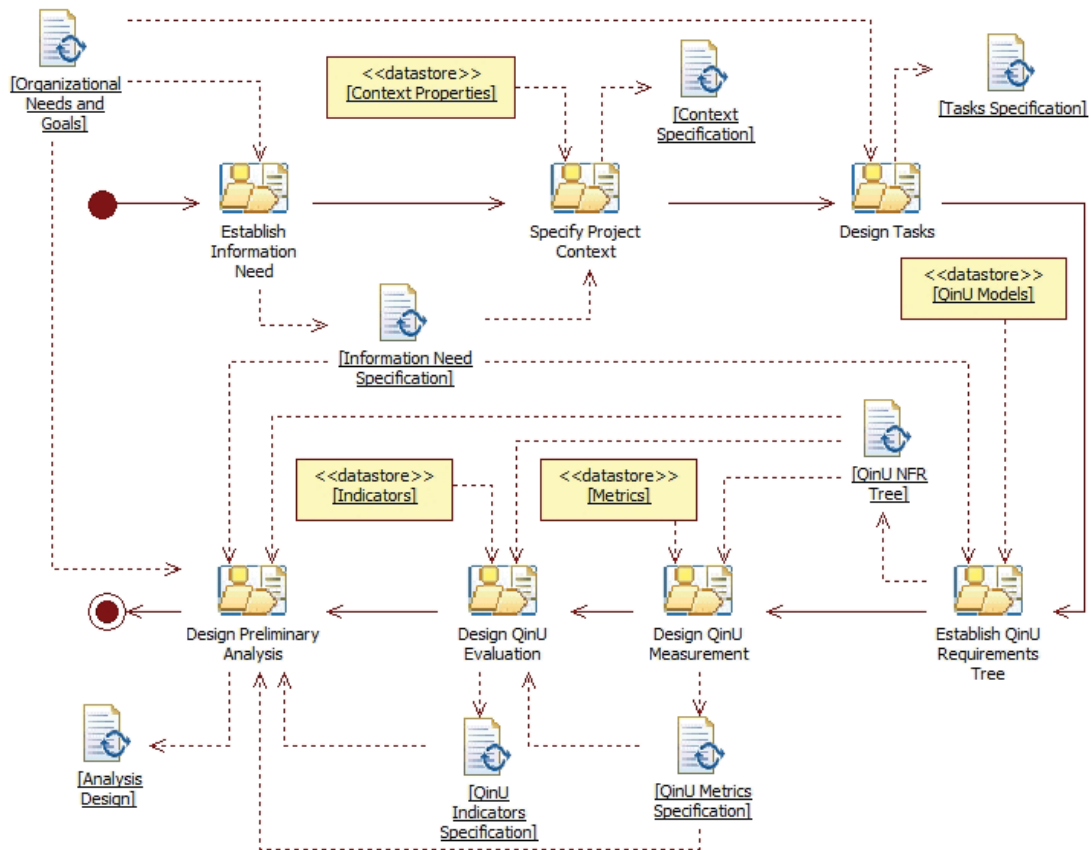


Figura 5-3: Fase (Ph) I: *Specify Requirements and Evaluation Criteria for QinU.*

5.2.1.1 Establish Information Need

Esta actividad, de acuerdo al marco conceptual C-INCAMI (particularmente recordar el paquete *requirements* de la Figura 4-1), involucra cuatro tareas, a saber:

- ✓ Definir el punto de vista del usuario
- ✓ Establecer el objeto
- ✓ Establecer entidad
- ✓ Identificar el foco

Se pueden notar diferencias en esta actividad con la presente en GOCAME (comparar Figuras 4-5 y 5-4). Por ejemplo, la tarea *Definir el propósito* (de GOCAME) en esta fase de SIQinU no es necesaria porque el propósito siempre es “conocer” y “mejorar”. Otra diferencia que se puede encontrar es que en GOCAME, se pueden identificar varias entidades a evaluar (por ejemplo con el propósito de compararlas), mientras que en SIQinU sólo se evalúa una entidad.

Las tareas modeladas en la Figura 5-4 son llevadas a cabo por el rol *Non Functional Requirements (NFR) Manager* considerando las necesidades del *Evaluation Requester*. La salida de la actividad *Establish Information Need* es el documento *Information Need Specification*, el cual contiene la especificación concisa del punto de vista del usuario, el foco de calidad a evaluar (el cual debe ser Calidad en Uso o alguna de sus subcaracterísticas, por ejemplo, Eficiencia), la categoría (objeto) y la entidad que será analizada.

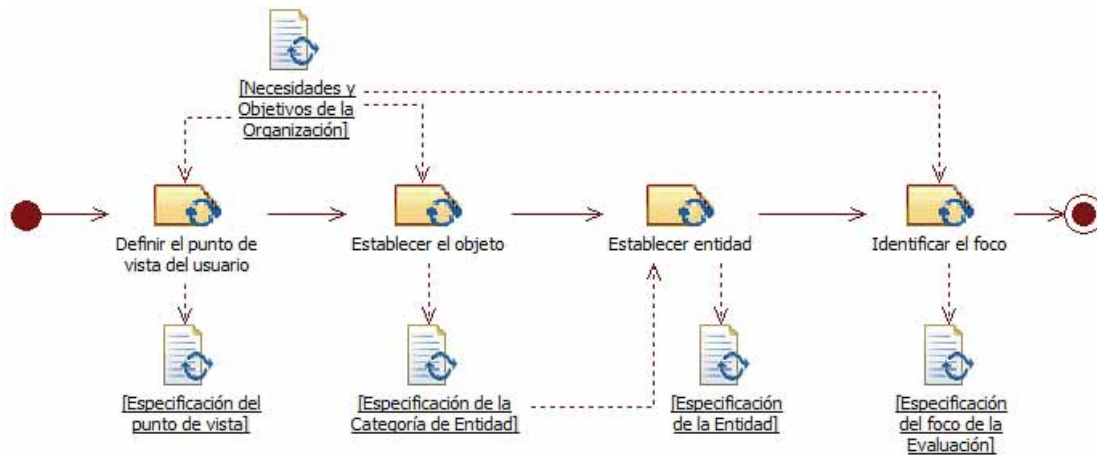


Figura 5-4: Tareas involucradas en la actividad *Establish Information Need*.

5.2.1.2 Specify Project Context

Una vez obtenido el artefacto denominado *Information Need Specification*, y como se observa en la Figura 5-3, se debe llevar a cabo la actividad *Specify Project Context*. Esta actividad reutiliza las tareas de la actividad *Especificación de Propiedades de Contexto* de A1 en GOCAME (recordar Sección 4.4.1.1). Específicamente involucra las siguientes dos tareas:

- ✓ *Seleccionar Propiedades de Contexto*
- ✓ *Cuantificar Propiedad de Contexto*

En primer lugar se deben *Seleccionar Propiedades de Contexto* relevantes para el proyecto en cuestión. Las mismas se seleccionan desde un repositorio organizacional con propiedades de contexto. Luego, por cada propiedad seleccionada, se debe ejecutar la tarea *Cuantificar Propiedad de Contexto*, utilizando la métrica asociada a la propiedad. Las entradas y salidas de estas tareas, así como la secuencia de ellas, puede observarse en la Figura 4-6. Al final, la salida es un documento con la especificación del contexto (artefacto *Context Specification* en la Figura 5-3) para el proyecto en cuestión.

5.2.1.3 Design Tasks

En esta actividad deberían ser diseñadas las tareas¹⁶ (y sub-tareas) más comunes y representativas a realizar utilizando la aplicación software/web. Es importante escoger aquellas tareas que permitan recolectar suficientes datos (a partir de los log-files) acerca del uso de la aplicación. Se debe tener presente que cuando se diseñan las tareas, el objetivo final es poder identificar atributos de Calidad Externa que afecten a atributos de Calidad en Uso. Por lo tanto, el diseño de las tareas es una actividad crítica en SIQinU, ya que partir de ellas se buscará evaluar y mejorar la Calidad en Uso en base a las *Necesidades y Objetivos de la Organización*.

La actividad *Design Tasks* implica las siguientes tareas:

- ✓ *Identificar tareas*
- ✓ *Especificación de la tarea*

¹⁶ El término tarea aquí se refiere a tareas de usuario, es decir, al conjunto de acciones que debe realizar un usuario utilizando una aplicación software/web para lograr un objetivo, y no a las tareas propias del proceso SIQinU.

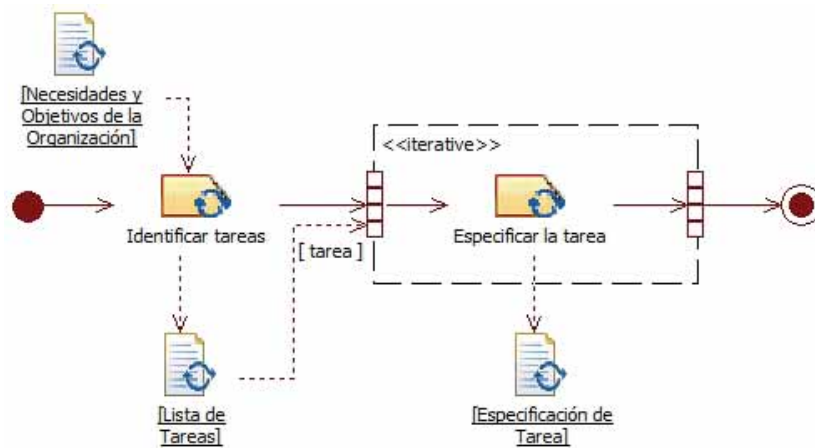


Figura 5-5: Tareas involucradas en la actividad *Design Tasks*.

La primera de ellas, *Identificar tareas*, conlleva nombrar y describir cada una de las tareas del usuario que serán evaluadas, así como también identificar el objetivo de cada una. Como resultado se obtiene una *Lista de Tareas*. Luego, por cada tarea de la *Lista de Tareas*, se debe *Especificar la tarea*. Esta última implica detallar en un documento de *Especificación de Tarea* las acciones o submetas que debe realizar el usuario para completar el objetivo de la tarea (de usuario), definir el orden en que estas acciones deben llevarse a cabo, e identificar las pantallas y componentes asociados a cada una de las acciones de la tarea.

El flujo entre las diferentes tareas de la actividad *Design Tasks* se aprecian en la Figura 5-5. El resultado final es el documento *Tasks Specification* (ver su composición en la Figura 5-6) producido por el rol *NFR Manager*.



Figura 5-6: Vista informacional del artefacto *Tasks Specification*.

5.2.1.4 Establish QinU Requirements Tree

Esta actividad de Ph I reutiliza las tareas de la actividad *Establecer el Árbol de Requerimientos* de la actividad A1 en GOCAME (recordar la Figura 4-7). Por lo tanto las tareas a llevar a cabo son:

- ✓ *Seleccionar un Modelo de Concepto*
- ✓ *Definir el Árbol de Requerimientos*

El modelo de concepto es seleccionado desde un repositorio organizacional. Para ello, el *NFR Manager* debe tener en cuenta el foco de la evaluación -el cual es Calidad en Uso o alguna de sus subcaracterísticas asociadas. La salida de esta actividad (ver Figura 5-3) es el documento *QinU NFR Tree*. Este artefacto contiene los conceptos (características), subconceptos (subcaracterísticas) y atributos

asociados al concepto foco de la evaluación. Un ejemplo de este documento puede observarse en la Figura 4-9.

5.2.1.5 Design QinU Measurement

La actividad *Design QinU Measurement* reutiliza tareas de la actividad A2 de GOCAME (recordar Sección 4.4.1.2). Al igual que la actividad A2, tiene por objetivo escoger las métricas que se utilizarán para cuantificar los diferentes atributos del árbol de requerimientos (artefacto *QinU NFR Tree*). Específicamente, las tareas asociadas a esta actividad son:

- ✓ *Seleccionar una Métrica*
- ✓ *Identificar las Métricas relacionadas*
- ✓ *Seleccionar una Herramienta*

Por cada atributo del árbol de requerimientos el rol *Metrics Expert* debe *Seleccionar una Métrica*. Esta tarea consiste en escoger, desde un catálogo, una métrica adecuada para el atributo, teniendo en cuenta la *Especificación del Contexto*. En el caso de que la métrica seleccionada sea una métrica indirecta, es decir, que esté relacionada a otras métricas (recordar el componente de Medición en la Sección 4.2.4), también deberá *Identificar las Métricas relacionadas*, lo cual incluye identificar los atributos que son cuantificados por estas métricas relacionadas. Esta última tarea permite identificar métricas y atributos extras asociados a las métricas indirectas para que, en el futuro, el *Data Collector* pueda recuperar apropiadamente los datos.

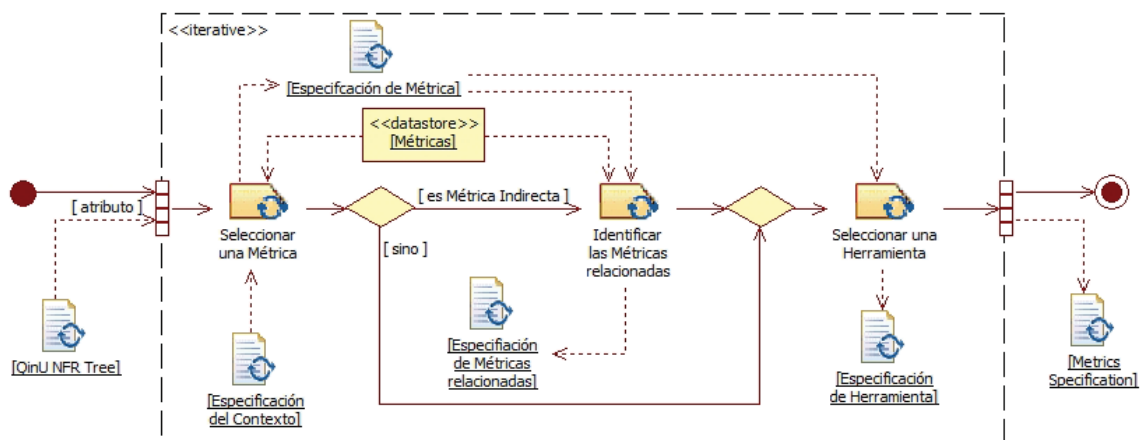


Figura 5-7: Actividad *Design QinU Measurement*.

Por último, se deberá *Seleccionar una Herramienta* software que automatice el procedimiento de medición/cálculo de la métrica. Las herramientas asociadas a las métricas deben tener la capacidad de obtener las medidas de los atributos a partir del análisis de archivos de registro (log files). Estos archivos almacenan diferentes datos acerca de las acciones del usuario realizando las tareas. De esta forma SIQinU promueve realizar mediciones de una manera no intrusiva para el usuario. Esta es otra diferencia que posee SIQinU respecto de GOCAME.

En la Figura 5-7 se puede observar la secuencia de estas tareas, así como sus correspondientes entradas y salidas. Al finalizar la actividad *Design QinU Measurement*, es decir, luego de realizar las tareas mencionadas para cada uno de los atributos, se consigue el documento *QinU Metrics Specification*. Este documento

es denominado *Especificación de Métricas* en GOCAME (recordar actividad A2). Su estructura puede apreciarse en la Figura 4-31.

5.2.1.6 Design QinU Evaluation

El objetivo de la actividad *Design QinU Evaluation* de la Fase I es definir para cada atributo y concepto calculable del árbol de requerimientos un indicador que lo evaluará. Para ello, primero se definen indicadores elementales para interpretar los valores medidos de los atributos y, luego, indicadores derivados para obtener el grado de satisfacción brindado por cada uno de los conceptos calculables. Como entradas a esta actividad, se debe contar con los documentos *QinU RNF Tree* y *QinU Metrics Specification*.

La actividad *Design QinU Evaluation* corresponde con la actividad A4 de GOCAME, por lo tanto las sub-actividades a realizar son (recordar la Figura 4-16):

- ✓ *Identificar Indicadores Elementales*
- ✓ *Identificar Indicadores Derivados*

La actividad *Identificar Indicadores Elementales* consiste en llevar a cabo iterativamente la tarea *Definir Indicador Elemental* para cada uno de los atributos del árbol de requerimientos. Teniendo en cuenta el componente de Evaluación (recordar la Figura 4-1), esta tarea implica establecer un modelo elemental (función), los criterios de decisión, la escala, el tipo de escala y la unidad del indicador. Estos metadatos pueden ser recuperados desde un repositorio de indicadores, lo cual agiliza el diseño. La vista de comportamiento de esta tarea se registra en la Figura 4-17, mientras que en la Figura 4-18 se presenta un ejemplo de una *Especificación de Indicador Elemental*, el cual es el artefacto producido por la tarea *Definir Indicador Elemental*.

Por otro lado, la actividad *Identificar Indicadores Derivados* consiste en realizar de forma iterativa la tarea *Definir Indicador Derivado* para cada concepto calculable del árbol de requerimientos (recordar la Figura 4-19). Al igual que para un indicador elemental, al definir un indicador derivado, se debe indicar el modelo global, fijar los criterios de decisión asociados al modelo global, identificar la escala, el tipo de escala y la unidad que tendrán los indicadores. La salida de esta tarea es una *Especificación de Indicador Derivado*.

5.2.1.7 Design Preliminary Analysis

La última actividad de la Fase I es *Design Preliminary Analysis*. Durante esta actividad, el rol *Analysis Designer* debe llevar a cabo la tarea *Diseñar el análisis*. Esta tarea implica decidir acerca de los métodos y técnicas matemáticas y estadísticas, así como herramientas de software, que se utilizarán para el análisis, presentación y visualización de la información. Dicha tarea es parte de la actividad A6 de GOCAME. Recordar la Figura 4-24 donde se indican las entradas y salidas de la tarea *Diseñar el análisis*.

5.2.2 Fase II: Perform QinU Evaluation and Analysis (Ph II)

Las actividades de la segunda fase están enfocadas en conseguir el primer propósito de la estrategia SIQinU, a saber: conocer el nivel de satisfacción actual de la Calidad en Uso de la WebApp. Para lograr esto, las siguientes cuatro actividades deberán llevarse a cabo durante Ph II:

- ✓ *Collect Data*

- ✓ *Quantify Attributes*
- ✓ *Calculate Indicators*
- ✓ *Conduct Preliminary Analysis*

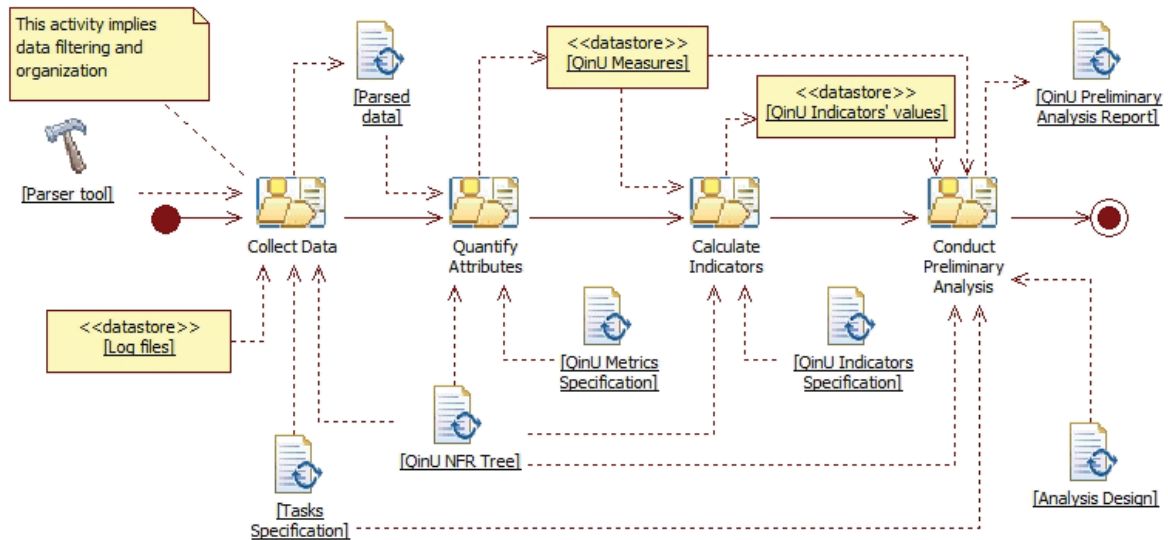


Figura 5-8: Fase (Ph) II: *Perform QinU Evaluation and Analysis*.

En la Figura 5-8 se presenta el flujo entre estas actividades, así como sus entradas y salidas. De las actividades de esta fase, solo *Collect Data* es una actividad propia de SIQinU, las restantes se derivan de la estrategia multi-propósito GOCAME.

La actividad *Collect Data* ha sido incorporada en SIQinU debido a que las medidas de los diferentes atributos deben ser obtenidas a partir de los datos de uso almacenados en archivos de registro (log files), y así lograr una medición no intrusiva. A continuación se detallan esta actividad, así como el resto de las actividades de Ph II.

5.2.2.1 *Collect Data*

Durante esta actividad se procesan los datos de los archivos de registro para ser utilizados luego en la cuantificación de los atributos. A continuación se presentan las tareas a ejecutarse durante esta actividad:

- ✓ *Analyze log files and filter data*
- ✓ *Organize data*

Durante la tarea *Analyze log files and filter data* se debe analizar los diferentes archivos que contienen datos de uso del usuario (representado mediante un objeto etiquetado como *Log files* y con estereotipo <<datastore>> en la Figura 5-9) y filtrar aquellos datos que están relacionados a las tareas evaluadas (definidas en el documento *Tasks Specification*). Al finalizar esta tarea, se deberá generar un archivo que contenga sólo aquellos registros con datos de uso del usuario (artefacto *User usage data*) que sean relevantes para la evaluación.

Seguidamente se lleva a cabo la tarea *Organize data*. Esta tarea consiste en organizar y estructurar de manera conveniente los datos del archivo *User usage data*. Es deseable que la organización de los datos se realice en base a los atributos definidos en el artefacto *QinU NFR Tree* y se presenten de una manera “amigable”.

De esta forma, luego será más fácil obtener las medidas de los atributos. El producto de trabajo obtenido al final se denomina *Parsed data*.

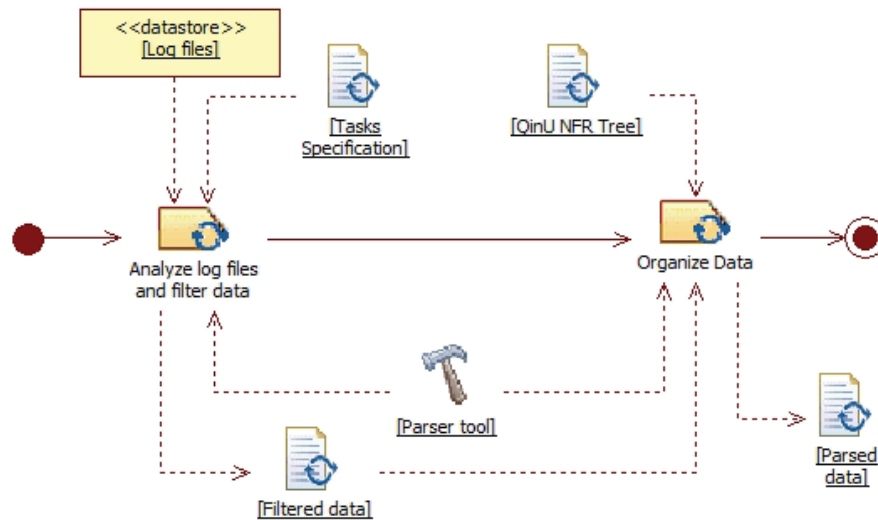


Figura 5-9: Actividad *Collect Data*.

Las diferentes tareas involucradas en la actividad *Collect Data* son realizadas por el rol *Log Analyzer*. Es recomendable que, quien posea este rol, tenga conocimientos acerca de herramientas software que faciliten el análisis de los archivos de registro. Ejemplos de aplicaciones de análisis de archivos de registro son: WebLog Expert (<http://www.weblogexpert.com>), Chainsaw (<http://logging.apache.org/chainsaw>), Deep Log Analyzer (<http://www.deep-software.com>) y Apache Log Viewer (<http://www.apacheviewer.com>), por mencionar algunas. Es deseable el uso de estas herramientas debido al alto volumen de datos que pueden contener los archivos de registro. Por otra parte, el *Log Analyzer* junto con el *Developer* (este rol será introducido en la Fase V) son los encargados de introducir código en la aplicación evaluada con el objetivo de registrar ciertos datos de uso que son relevantes para la evaluación.

5.2.2.2 Quantify Attributes

Luego de producir el artefacto *Parsed data*, se procede a cuantificar (medir) cada atributo del árbol de requerimientos (*QinU NFR Tree*). Las medidas son obtenidas aplicando los procedimientos de medición y de cálculo asociados a las métricas que se encuentran en el documento *QinU Metrics Specification*. Estas medidas son almacenadas en un repositorio (ver objeto *QinU Measures* en la Figura 5-8).

La actividad *Quantify Attributes* se corresponde con la sub-actividad *Medir los atributos* de la actividad A3 de GOCAME (recordar Sección 4.4.1.3). El lector puede observar la vista de comportamiento de la presente actividad remitiéndose a la Figura 4-15.

5.2.2.3 Calculate Indicators

Teniendo en cuenta las medidas (valores) y la especificación de indicadores (*QinU Indicators Specification*), el rol *Indicator Calculator* se encarga de calcular los valores de los indicadores para cada atributo y concepto calculable del árbol de requerimientos. Para ello se deben llevar a cabo las actividades *Calcular Indicadores Elementales* y *Calcular Indicadores Derivados* de la actividad A5 de GOCAME. Recordar las tareas intervinientes, así como sus entradas y salidas, en la Sección 4.4.1.5.

Finalmente, los diferentes valores de los indicadores, junto con sus metadatos, son almacenados en un repositorio para su posterior análisis.

5.2.2.4 Conduct Preliminary Analysis

Por último, el rol *Data Analyzer* realiza un análisis preliminar. Tal análisis se realiza teniendo en cuenta el documento *Analysis Design*, producido en la actividad *Design Preliminary Analysis*, en Ph I. Como salida se obtiene el documento *Preliminary analysis report*, en el cual se describen los problemas de Calidad en Uso descubiertos.

La actividad *Conduct Preliminary Analysis* permite conocer el nivel actual de satisfacción alcanzado por la aplicación evaluada, tanto a nivel global (es decir, respecto de la característica de más alto nivel), como así también respecto a cada atributo particular para cada una de las tareas de usuario. Más información respecto del análisis puede verse en la Sección 4.4.1.6, ya que la actividad A6 de GOCAME es similar a la presente.

5.2.3 Fase III: Derive/Specify Requirements and Evaluation Criteria for EQ (Ph III)

En el caso de que el análisis realizado en Ph II indique que la WebApp contiene aspectos de la Calidad en Uso que deben mejorarse, se debe llevar a cabo la Ph III. De lo contrario, es decir si la WebApp satisface el nivel de Calidad en Uso requerido por el rol *Evaluation Requester*, se culmina con el proceso. Este flujo puede observarse en la Figura 5-2. La Ph III se centra en el diseño de aspectos relacionados a la evaluación de la Calidad Externa. Las actividades a llevarse a cabo durante esta fase son (ver Figura 5-10):

- ✓ *Establish EQ Requirements Tree*
- ✓ *Design EQ Measurement*
- ✓ *Design EQ Evaluation*
- ✓ *Design EQ Analysis*

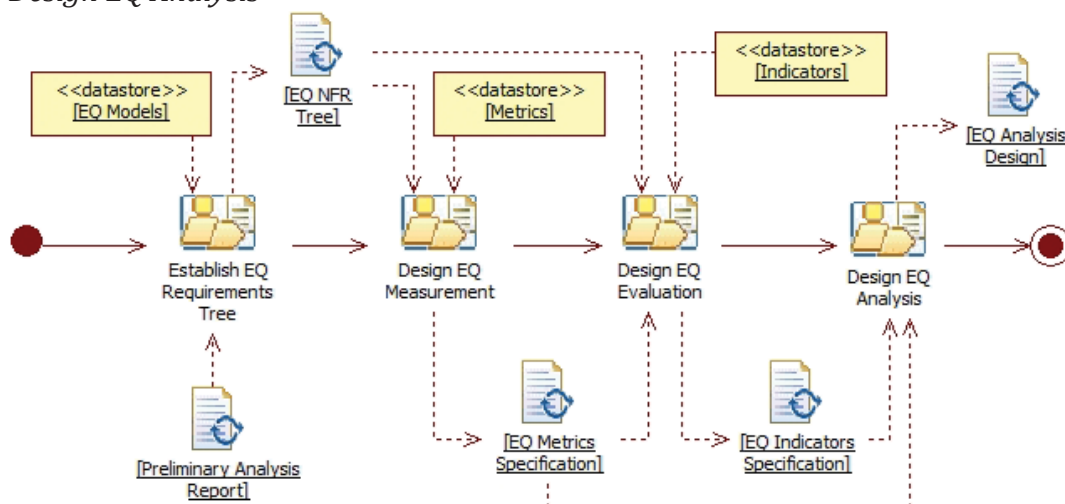


Figura 5-10: Fase (Ph) III: *Derive/specify requirements and evaluation criteria for EQ.*

Notar que las actividades de esta fase son similares a las de Ph I (recordar la Figura 5-3), pero en lugar de enfocarse en la Calidad en Uso lo hacen en la Calidad Externa. Además cabe mencionar que estas actividades se corresponden con las de

Establecer el *Árbol de Requerimientos* (A1) y con las de las actividades A2 y A4 de GOCAME.

Durante esta fase se debe diseñar un nuevo árbol de requerimientos (ahora centrado en Calidad Externa), teniendo en cuenta el *Preliminary analysis report* (producido en Ph II) y el árbol de requerimientos de Calidad en Uso (definido en Ph I). Este nuevo árbol debe construirse considerando aquellas características de Calidad Externa que potencialmente impactarán positivamente en la Calidad en Uso si fueran mejoradas. Además, en esta fase se especificarán las métricas e indicadores que se usarán para evaluar la Calidad Externa de la WebApp. A continuación se detallan las 4 actividades de Ph III.

5.2.3.1 *Establish EQ Requirements Tree*

Esta actividad tiene por objetivo producir un árbol de requerimientos, cuyos atributos y características (conceptos calculables) serán dimensiones de la Calidad Externa. El *NFR Manager* será el encargado de definir un árbol en el cual los atributos y características serán aquellos que luego de un cambio pudieran influir positivamente en la Calidad en Uso, más específicamente en las dimensiones de Calidad en Uso cuyo nivel de satisfacción es bajo según los criterios de decisión definidos para la WebApp evaluada (esta información se encuentra en el *Preliminary analysis report*).

Las tareas a llevar a cabo por el rol *NFR Manager* son: *Seleccionar un Modelo de Concepto* (desde un repositorio de modelos de Calidad Externa) y *Definir el Árbol de Requerimientos* que se ajuste a las necesidades. Las entradas y salidas de estas tareas pueden verse en la Figura 5-11. Notar que estas tareas fueron definidas en la Sección 4.4.1.1

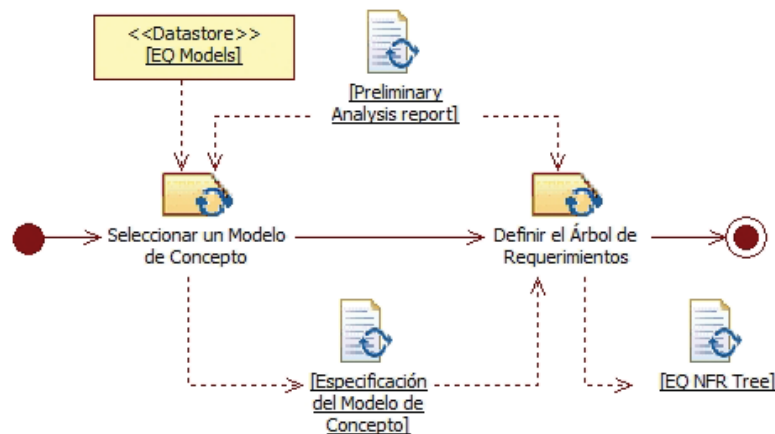


Figura 5-11: Tareas de la actividad *Establish EQ Requirements Tree*.

5.2.3.2 *Design EQ Measurement*

Siguiendo la Figura 5-10, una vez que el árbol de requerimientos para Calidad Externa fue instanciado se debe diseñar la medición, es decir, producir la especificación de las métricas a usar en la siguiente fase. Esta actividad es realizada por el *Metrics Expert* y es similar a *Design QinU Measurement* (recordar Figura 5-7 en Ph I), pero ahora la secuencia de tareas es ejecutada para cada uno de los atributos de Calidad Externa. Por otro lado, en *Design EQ Measurement* la tarea *Seleccionar una Herramienta* es opcional (como en la Figura 4-11) en lugar de ser obligatoria. En *Design QinU Measurement* esta tarea es mandatoria ya que las mediciones para los atributos de Calidad en Uso deben realizarse de manera no

intrusiva para no afectar el desempeño del usuario, en cambio para Calidad Externa esto ya no es necesario.

5.2.3.3 Design EQ Evaluation

Esta actividad es similar a *Design QinU Evaluation* (Ph I). Por lo tanto, el rol *Indicators Expert* debe especificar un indicador por cada atributo y concepto calculable del árbol de requerimientos (*EQ NFR Tree*). Cabe mencionar que, al diseñar los nuevos indicadores, se pueden utilizar rangos/niveles de aceptabilidad (*DecisionCriterion* en Figura 4-1) diferentes a los escogidos para los indicadores de Calidad en Uso.

5.2.3.4 Design EQ Analysis

Finalmente se debe realizar la actividad *Design EQ Analysis*. Ésta implica decidir acerca de los métodos y técnicas matemáticas y estadísticas, así como herramientas software, que se utilizarán para el análisis, presentación y visualización de la información.

5.2.4 Fase IV: Perform EQ Evaluation and Analysis (Ph IV)

Una vez finalizada Ph III, se está en condiciones de realizar la evaluación de la Calidad Externa de la WebApp. Esta fase es similar a Ph II, pero enfocada en la Calidad Externa. Las actividades involucradas en Ph IV son:

- ✓ *Quantify Attributes*
- ✓ *Calculate Indicators*
- ✓ *Conduct Analysis*

En la Figura 5-12 se muestra la secuencia entre estas actividades así como las respectivas entradas y salidas. Si se compara esta fase con Ph II (recordar la Figura 5-8) se observa que en Ph IV no se lleva a cabo la actividad *Collect Data*. Esta actividad se realiza en Ph II para obtener datos de uso y así lograr una medición y evaluación no intrusiva. En cambio en Ph IV, la medición (actividad *Quantify Attributes*) y evaluación (*Calculate Indicators*) se realizan por inspección de expertos.

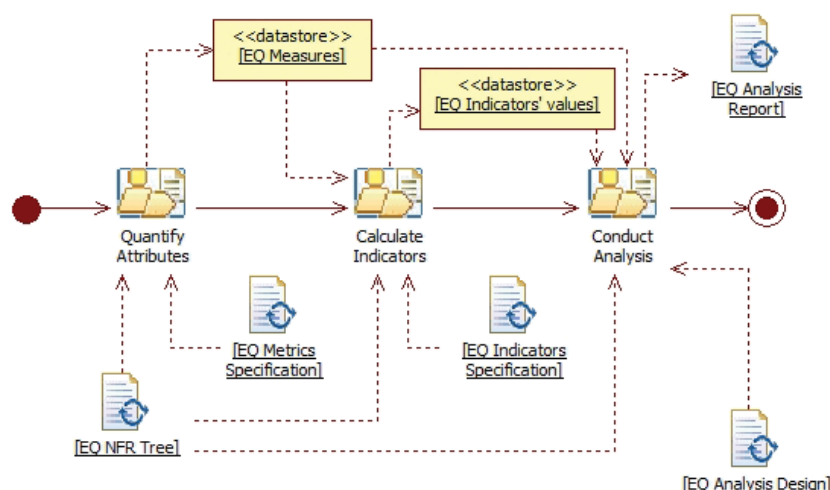


Figura 5-12: Fase (Ph) IV: *Perform EQ Evaluation and Analysis*.

Siguiendo la secuencia de la Figura 5-12, en primer lugar el *Data Collector* debe medir cada uno de los atributos del árbol de requerimientos en la actividad *Quantify Attributes*. Luego, todos los indicadores, elementales y derivados, deben

ser calculados por el *Indicator Calculator* en la actividad *Calculate Indicators*. Finalmente, el rol *Data Analyzer* debe realizar la actividad *Conduct Analysis*. Esta última actividad genera un documento denominado *EQ Analysis Report*, el cual cuenta con información que permitirá identificar, por ejemplo, partes de la WebApp que necesitan ser mejoradas desde el punto de vista de la Calidad Externa.

Las tres actividades de Ph IV se corresponden con actividades de GOCAME. Específicamente, *Quantify Attributes* se corresponde con la actividad *Medir los atributos* de A3 (Sección 4.4.1.3), *Calculate Indicators* con A5 (Sección 4.4.1.5) y *Conduct Analysis* con A6 (Sección 4.4.1.6). Por lo tanto se invita al lector a dirigirse a las respectivas secciones para detalles de las tareas a llevar a cabo en cada actividad.

5.2.5 Fase V: Recommend, Perform Improvement Actions, and Re-evaluate EQ (Ph V)

Considerando el *EQ Analysis Report* generado en la actividad *Conduct Analysis* (Ph IV), se realizan recomendaciones para mejorar aquellos atributos del producto (WebApp) que deben incrementar su nivel de satisfacción. Luego que los cambios recomendados fueron realizados, y se ha producido una nueva versión de la WebApp, se realiza una re-evaluación de la Calidad Externa para determinar la ganancia lograda por las mejoras introducidas. Esta fase incluye (ver la Figura 5-13):

- ✓ *Recommend Improvement Actions*
- ✓ *Design Improvement Actions*
- ✓ *Perform Improvement Actions*
- ✓ *Evaluate Improvement Gain*

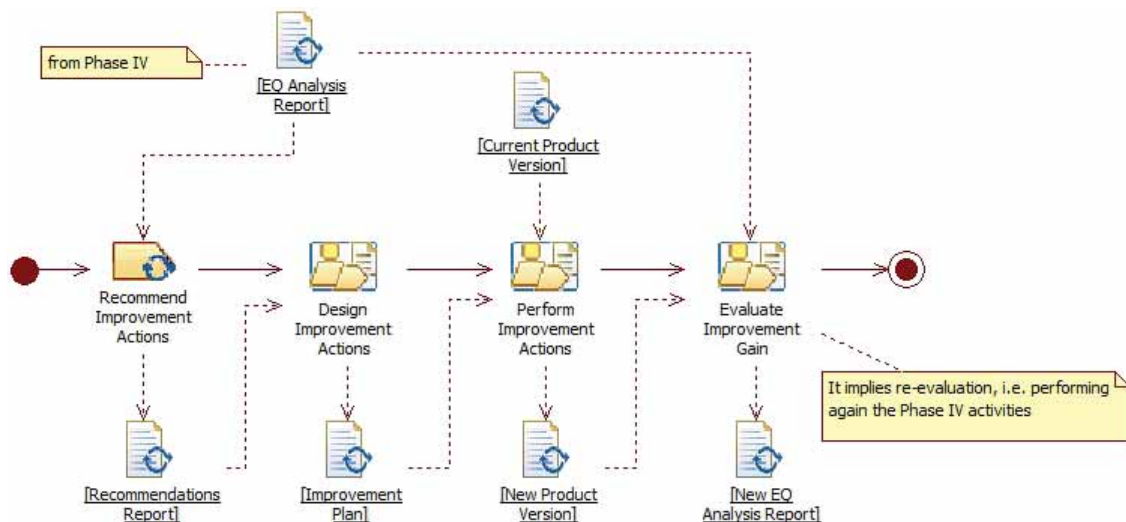


Figura 5-13: Fase (Ph) V: *Recommend, Perform Improvement Actions and Re-evaluate EQ*.

5.2.5.1 Recommend Improvement Actions

El rol *Recommender* es quien lleva a cabo la tarea *Recommend Improvement Actions* y produce un informe con recomendaciones (*Recommendations Report*). Es deseable que en este informe se listen, por ejemplo, ordenados de menor a mayor, los valores de los indicadores de los atributos que no alcanzan un nivel de aceptabilidad satisfactorio. Además, por cada atributo listado en el informe, el

Recommender debe sugerir una o más acciones de mejora para aumentar el nivel de satisfacción del atributo. El rol *Evaluation Requester* se encargará de priorizar las recomendaciones.

5.2.5.2 Design Improvement Actions

En base al *Recommendations Report*, el rol *Maintenance Project Manager* produce un plan de mejoras (*Improvement Plan*) en el cual indicará cómo se realizarán los cambios en la aplicación. El “cómo” implica enumerar cada uno de los métodos y técnicas que se emplearán para hacer efectivas las acciones de mejora en la próxima actividad (*Perform Improvement Actions*). Los métodos y técnicas para cambiar/mejorar la WebApp van desde reconfiguraciones parametrizadas, reestructuración del código y *refactoring* (como se hizo en [Olsina, y otros, 2008a]), hasta el rediseño arquitectural.

Los métodos a utilizar dependen de los recursos disponibles (desarrolladores, tiempo, etc.) y del resultado deseado (por ejemplo, una aplicación más fácil de operar, con mayor seguridad, más rápida, entre otros aspectos). A fin de escogerse los métodos más adecuados, es deseable la colaboración del *Evaluation Requester* ya que es quién puede brindar información acerca de la disponibilidad de los recursos de la organización.

5.2.5.3 Perform Improvement Actions

En base al documento *Improvements Plan*, el rol *Developer* realiza los correspondientes cambios en la WebApp (*Current Product Version*), produciendo así una nueva versión de la aplicación (ver *New Product Version* en la Figura 5-13). Cabe mencionar que las tareas involucradas en las actividades *Perform Improvement Actions* y *Design Improvement Actions* no se presentan debido a que las mismas están íntimamente relacionadas a las acciones de mejoras a realizar. Por lo tanto, las tareas son diferentes para cada caso particular en el cual se aplique SIQinU. Notar que esta actividad y la anterior son exclusivas de la estrategia SIQinU, es decir, no forman parte del proceso genérico de GOCAME.

5.2.5.4 Evaluate Improvement Gain

Una vez que todos los cambios fueron efectuados, la nueva versión de la aplicación debe ser evaluada respecto de su Calidad Externa. Esta evaluación permite determinar, en relación con la evaluación realizada en Ph IV, cuáles atributos han aumentado su nivel de satisfacción gracias a las mejoras introducidas y cuáles no. Las actividades involucradas en *Evaluate Improvement Gain* son las mismas de la Fase IV, a saber:

- ✓ *Quantify Attributes*
- ✓ *Calculate Indicators*
- ✓ *Conduct EQ Analysis*

La salida es un informe, llamado *New EQ Analysis Report* en la Figura 5-13 pero similar al *EQ Analysis Report* producido en Ph IV. Este nuevo informe incluye una comparación entre las dos versiones de la WebApp. Tal comparación debe realizarse a partir de los niveles de satisfacción alcanzados por los diferentes elementos (atributos y conceptos calculables) del árbol de requerimientos. La comparación permite determinar fácilmente la ganancia obtenida en la nueva versión y, además, identificar claramente los cambios que introducen mejoras en la

Calidad Externa, lo cual será útil como base de conocimiento para futuros proyectos de mejora.

Finalizada la re-evaluación, si aún hay atributos que pueden mejorarse, se vuelve a llevar a cabo Ph V. Es decir, se realizan nuevas recomendaciones de mejora a la WebApp, se implementan esas mejoras y se vuelve a evaluar la Calidad Externa. Puede suceder que el rol *NFR Manager* note que existen otros atributos, no considerados en el árbol de requerimientos de Calidad Externa, que pudieran afectar positivamente en la Calidad en Uso. Si este fuera el caso, se puede volver a realizar Ph III para definir un nuevo árbol de requerimientos y especificar las nuevas métricas e indicadores a utilizarse. Ambos flujos se encuentran reflejados en la Figura 5-2 del proceso de SIQinU y se corresponden con los ciclos numerados 1 y 2 en la Figura 5-1.

5.2.6 Fase VI: Re-evaluate Quality in Use and Analyze Improvement Actions (Ph VI)

Producida la nueva versión de la WebApp, y luego de que esta haya sido utilizada durante cierto tiempo por el mismo tipo de usuarios realizando las mismas tareas y en el mismo contexto real de uso que el de la versión previa, es posible llevar a cabo una nueva evaluación de la Calidad en Uso. Esta evaluación, permite determinar si las mejoras introducidas en la Calidad Externa han tenido un impacto positivo en la Calidad en Uso. Las actividades involucradas son:

- ✓ *Collect Data*
- ✓ *Quantify Attributes*
- ✓ *Calculate Indicators*
- ✓ *Conduct Improvement Actions Analysis*

Notar que las tres primeras actividades son las mismas que se llevan a cabo en Ph II. Por otro lado, la actividad *Conduct Improvement Actions Analysis* permite determinar la ganancia de la mejora y también hipotetizar relaciones existentes entre Calidad Externa y Calidad en Uso. A continuación se comenta esta última actividad.

5.2.6.1 Conduct Improvement Actions Analysis

Una vez que todos los indicadores fueron calculados (actividad *Calculate Indicators*), el *Data Analyzer* debe realizar dos tareas, a saber (ver la Figura 5-14):

- ✓ *Conduct Analysis*
- ✓ *Develop EQ/QinU Relationships*

La tarea *Conduct Analysis* consiste en realizar el análisis de la evaluación siguiendo los procedimientos especificados en el documento *Analysis Design* (producido en Ph I). Este análisis es similar al descrito en Ph II (recordar *Conduct Preliminary Analysis*, Sección 5.2.2.4). Como salida se obtiene el artefacto *QinU Analysis Report*. Este informe, al igual que el *New EQ Analysis Report* (producido en Ph IV), también incluye una comparación entre las dos versiones de la WebApp, pero ahora desde el punto de vista de la Calidad en Uso. Esta comparación permite observar la ganancia lograda gracias a los cambios introducidos en la Calidad Externa.

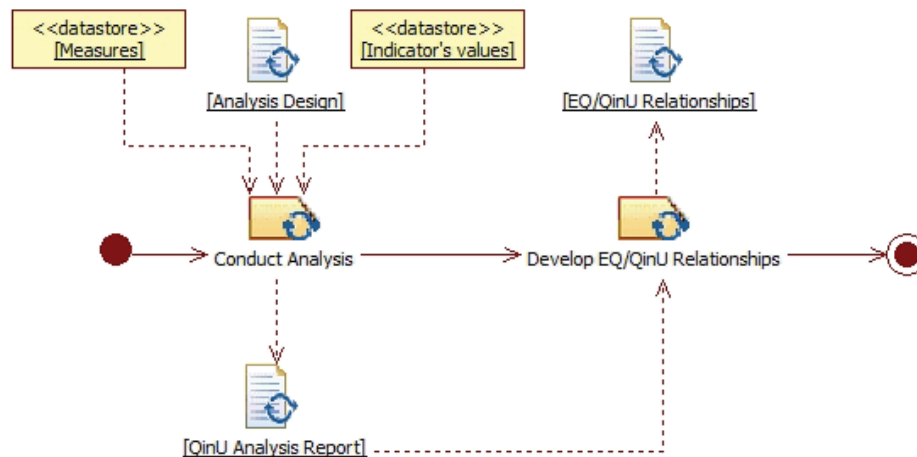


Figura 5-14: Entradas y salidas de la actividad *Conduct Improvement Actions Analysis*.

Luego, teniendo en cuenta que la Calidad Externa influye en la Calidad en Uso y, a su vez, la Calidad en Uso depende de la Calidad Externa [ISO/IEC, 2011], se lleva a cabo la tarea *Develop EQ/QinU Relationships*. Por lo tanto, esta tarea consiste en identificar posibles relaciones “influye” y “depende de”. Es decir, identificar aquellos atributos/conceptos calculables de Calidad Externa que al mejorarlos han impactado positivamente en los atributos/conceptos calculables de Calidad en Uso. Como salida, se produce el documento *EQ/QinU Relationships*.

Contar con estas relaciones, no sólo permite conocer qué dimensiones de la Calidad Externa mejoraron la Calidad en Uso de la WebApp evaluada, sino que tiene el objetivo final de desarrollar una base de conocimiento que ayude al *NFR Manager* (en Ph III) a derivar y definir un árbol de requerimientos cuyos atributos de Calidad Externa realmente dependan de los atributos de Calidad en Uso que deben mejorarse, y en definitiva para diseñar mejores aplicaciones en general.

5.3. Alineación de GOCAME y SIQinU

Según lo mencionado en la Sección 2.1.2, GOCAME es considerada una estrategia de M&E integrada ya que cuenta simultáneamente con las siguientes tres capacidades: i) un marco conceptual para el dominio de M&E, ii) una especificación de las vistas del proceso y, iii) métodos y herramientas específicos que permiten llevar a cabo las actividades del proceso. Particularmente, GOCAME hace uso del marco conceptual de M&E C-INCAMI (recordar la Sección 4.2), tiene especificado el proceso de M&E como se mostró en la Sección 4.4 y cuenta con el soporte de la metodología WebQEM y la herramienta C-INCAMI_Tool (comentadas en la Sección 4.5).

La estrategia SIQinU, al derivarse de GOCAME, también posee de manera integrada las tres capacidades mencionadas. Esto se representa alegóricamente en la Figura 5-15.

Considerando el proceso, SIQinU reutiliza las tareas de GOCAME. Por ejemplo, las tareas de las actividades A1, A2, A4 y en alguna medida de la A6 (recordar Figura 4-3) están incluidas en SIQinU en Ph I y III (ver Figura 5-2). De igual manera, las tareas de A3 y A5 están contenidas en Ph II y IV. Sin embargo, hay ciertas actividades y tareas de SIQinU que no están incluidas en GOCAME. Por ejemplo, en Ph V se agrupan las actividades orientadas a producir cambios para mejorar la WebApp. También en Ph II se incluyen actividades de filtrado y

recolección de datos, ya que SIQinU propone utilizar las capacidades del lado del servidor para recuperar, de una manera no intrusiva, los datos de uso del usuario.

Por otro lado, teniendo presente que la estrategia GOCAME es genérica en el sentido que puede utilizarse para evaluar Calidad Interna, Calidad Externa, Calidad en Uso, Costo, entre otros, los nombres de las actividades no indican el foco de la evaluación. Por el contrario, SIQinU específicamente se enfoca en la Calidad en Uso y en la Calidad Externa. Por lo tanto, SIQinU adapta ligeramente los nombres de las actividades que reutiliza de GOCAME indicando el foco de la evaluación. Por ejemplo, la actividad A5 de GOCAME se llama de manera genérica *Implementar la Evaluación*, mientras que en SIQinU se hallan las actividades *Implementar la Evaluación de Calidad en Uso* (Perform QinU Evaluation) e *Implementar la Evaluación de Calidad Externa y el Análisis* (Perform EQ Evaluation and Analysis), en Ph II y IV respectivamente (comparar Figura 4-3 y Figura 5-2). Lo mismo sucede con algunos nombres de los productos de trabajo, como *Especificación de Métricas* (en GOCAME) y *Especificación de Métricas para Calidad en Uso* (en SIQinU).

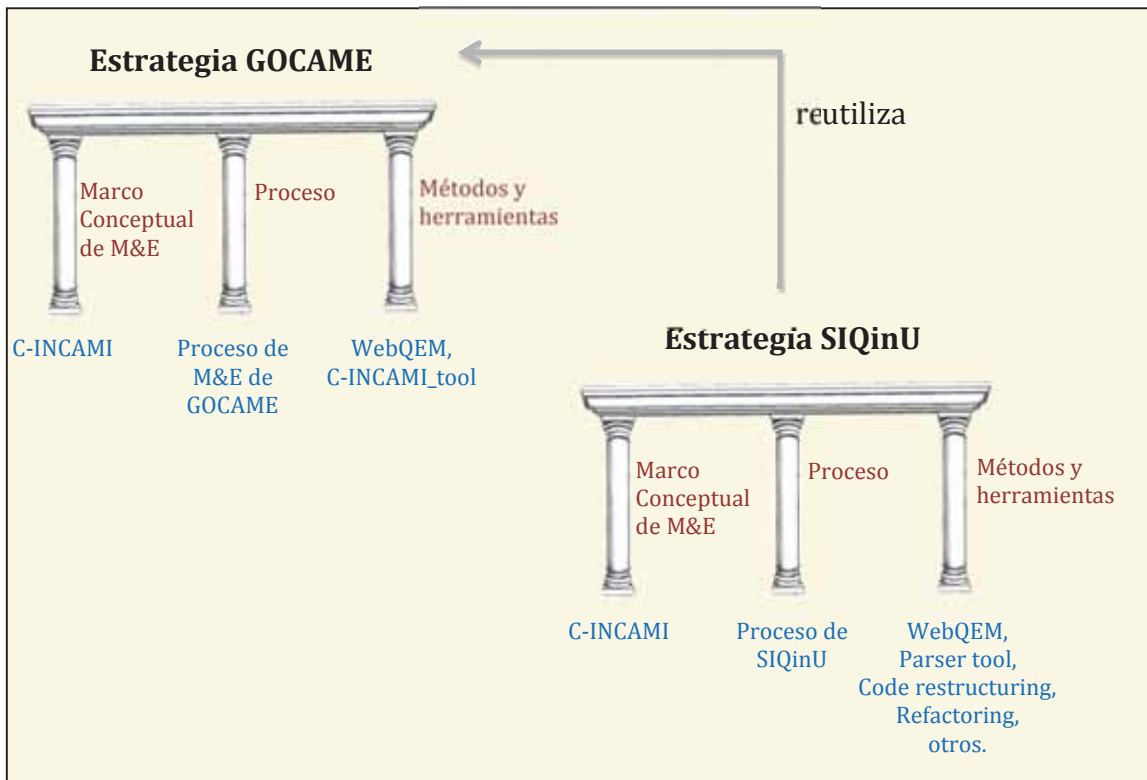


Figura 5-15: Alegoría de las tres capacidades de GOCAME, las cuales son reutilizadas en gran medida por SIQinU.

Respecto del marco conceptual, SIQinU reutiliza totalmente el marco C-INCAMI, esto es, la base conceptual ontológica de M&E y los componentes comentados en la Sección 4.2. Sin embargo, ciertas actividades de SIQinU no incluidas en GOCAME -por ejemplo las actividades de mejora en Ph V y aquellas relacionadas a la recolección de datos de manera no intrusiva en Ph II - actualmente carecen de raíz ontológica en C-INCAMI. Los conceptos de C-INCAMI están orientados a aspectos de requerimientos no funcionales, medición y evaluación, y no a acciones de cambio y mejora, lo cual implica otro alcance y modelo del dominio. No obstante, se debe tener presente que C-INCAMI es un marco flexible y se puede ampliar vinculándolo a otros modelos de dominio y marcos para hacer frente a esta cuestión. Si bien este aspecto escapa al alcance de

esta tesis, un claro ejemplo de que C-INCAMI puede enriquecerse se observa en la Sección 4.3, donde se enlaza el marco C-INCAMI con un marco conceptual de proceso.

Por último, los métodos y herramientas usados en GOCAME pueden ser reutilizados en SIQinU. Por ejemplo, la herramienta C-INCAMI_Tool (desarrollada para dar soporte al proceso de GOCAME) puede utilizarse en SIQinU mediante la creación de diferentes proyectos de M&E: un proyecto enfocado en la Calidad en Uso para llevar a cabo las actividades de Ph I y II, un proyecto centrado en la Calidad Externa para Ph III y IV, y otro proyecto enfocado en la Calidad en Uso para Ph VI. Sin embargo, SIQinU necesita métodos y herramientas adicionales. Por ejemplo, herramientas de *parsing* para obtener medidas a partir de los datos de uso de la WebApp y técnicas de refactoring, entre otras.

Finalmente, los roles definidos para GOCAME (recordar Tabla 4-8) son reutilizados en SIQinU y se agregan otros para las actividades de la fase Ph V, a saber: el *Maintenance Project Manager*, quien es responsable de liderar el proyecto de mantenimiento e identificar los métodos, técnicas y herramientas más apropiadas para introducir los cambios –mejoras- a la aplicación; y el *Developer*, quien es responsable de realizar los cambios a la aplicación.

Claramente, GOCAME y SIQinU, están alineadas al basarse en las mismas capacidades y al reutilizar tareas, productos de trabajo, marco conceptual, métodos, herramientas y roles. Sin embargo, debido a que SIQinU es una estrategia de propósito específico, ésta define su propio proceso y requiere métodos y herramientas que no están especificados en GOCAME. Además, el propósito definido en la necesidad de información siempre es “conocer” y “mejorar”, mientras que en GOCAME el propósito puede ser “conocer”, “predecir”, “controlar”, “comparar”, etc. Asimismo, GOCAME fue diseñada para evaluar diferentes conceptos calculables (como Usabilidad, Mantenibilidad, Costo, etc.) y categorías de entidad (como por ejemplo, productos y recursos, entre otros). En cambio, SIQinU fue pensada para evaluar y mejorar la Calidad en Uso de sistemas en uso, considerando la Calidad Externa de sistemas y productos software, como por ejemplo una WebApp.

5.4. Plantilla de Especificación de Procesos

Al observar el gran número de actividades, tareas, productos de trabajo y roles involucrados en el proceso de SIQinU, se evidencia la necesidad de contar con mecanismos adecuados para la documentación de procesos. En este sentido, por ejemplo, existen diferentes plantillas a fin de especificar los procesos.

Una de las plantillas más conocidas es ETVX (*Entry criteria, Tasks, Validation, eXit criteria*) [Radice, y otros, 1985], la cual presenta i) una lista de requisitos que deben ser satisfechos antes de comenzar las tareas (*Entry criteria*); ii) un conjunto de tareas que indican qué es lo que se debe hacer (*Tasks*); iii) un procedimiento de validación para controlar los productos de trabajo producidos por las tareas (*Validation*); y iv) una lista de criterios de salida que deben cumplirse para que la actividad se considere finalizada (*eXit criteria*). En la Figura 5-16 se presenta el modelo de una plantilla ETVX.

Process		
Entry criteria	Tasks	Exit criteria
	Validation	

Figura 5-16: Plantilla ETVX.

También en [Esteban, y otros, 2003] se considera una plantilla orientada a catalogar actividades de desarrollo web. Esta plantilla reúne la siguiente información: Nombre, Alias, Definición/Objetivo, Descripción, Categoría, Aplicabilidad, Proceso o sub-proceso al que pertenece, Entrada, Salida, Rol del Agente, Método y Herramienta. Finalmente se puede mencionar el trabajo de [Heidrich, y otros, 2006], en el cual se utiliza una plantilla con los siguientes campos: ID, Name, Role, Input, Output, Entry condition, Exit Condition, Description, Methods/Techniques, Guidelines y Materials/Tools (ver ejemplo en Figura 5-17).

ID	Activity 0.5
Name	Problem definition
Role	C (optional), D, U: - C: Checks whether Goal Definition is in line with business goals. - D: Supports U during problem definition. - U: Responsible for problem identification and definition.
Input	Customer Sheet If available: process models and measurement-based quantitative models.
Output	Goal Definition, Project Logfile
Entry condition	U has been identified (cf. Activity 0.4)
Exit condition	Goal Definition exists in written form or SD modeling project has been cancelled.
Description	- Identification of a problem that - if solved - would help U with his/her daily work. - Formal documentation of the problem definition (SDM Goal Definition). - Notes should be taken of all relevant information that could be used to define the dynamic hypothesis in Phase 1, e.g., first assumptions about cause-effect relationships, suggestions of potential problema solutions, relevant existing models, subject matter experts, etc. This kind of information is recorded in the Project Logfile.
Methods / Techniques	Knowledge elicitation techniques: Interview (semi-structured or unstructured) and focused discussion (goal-related)
Guidelines	- The problem definition should be well-focused and be stated in concrete terms. The Goal Definition Template should be used. - In order to be suitable for SD analysis, the problem has to deal with phenomena that show dynamic behavior. - In order to be suitable for SD analysis, the system that is going to be investigated for problem solution, has to be viewed as a feedback (or closed) loop system. This assumption implies that a change in the system structure - and not an alteration of the inputs - is in the focus of interest of the problem solution.
Materials / Tools	Goal Definition Template

Figura 5-17: Ejemplo de la plantilla de descripción de proceso utilizada en [Heidrich, y otros, 2006].

Las plantillas mencionadas comparten ciertos campos, mientras que difieren en otros. Sin embargo, más allá de este aspecto, una característica que es interesante resaltar es que sólo cuentan con descripciones textuales del proceso. Este enfoque tiene sus inconvenientes (rever las razones en la Sección 2.2). Por otro lado contar sólo con diagramas del proceso tampoco es la solución. Debido a que los modelos son una abstracción de la realidad, hay ciertos detalles que deben omitirse o son difíciles de reflejar en el modelo. Por lo tanto se propone considerar una plantilla que integre ambos enfoques, es decir, que cuente con descripciones textuales (como las plantillas mostradas) y, al mismo tiempo, disponga de modelos del proceso -específicamente de las diferentes vistas-. De esta manera las especificaciones de proceso tenderán a ser más fáciles de comprender y comunicar.

En la Figura 5-18 se presenta la plantilla propuesta. Para el diseño de la misma se han tomado como base los conceptos del dominio de proceso (recordar el marco conceptual de proceso de la Figura 3-23), así como otros documentos de especificación de procesos ([Esteban, y otros, 2003], [Heidrich, y otros, 2006] e ISO 12207 [ISO/IEC, 2008], por mencionar algunos). La información que la plantilla contiene es: el *nombre* del proceso (el cual comienza con un verbo en infinitivo) y el *código* que lo identifica de manera unívoca dentro de un catálogo, el *objetivo* que se persigue al ejecutar el proceso, un *modelo* de proceso que integre aspectos de las vistas funcional, de comportamiento y organizacional, una breve *descripción* textual del proceso que permiten agregar detalles y comentarios sobre el modelo de proceso, los *productos de trabajo de entrada y de salida* (se puede hacer uso de la vista informacional), y las *pre* y *post-condiciones* (mediante una lista de requisitos escritos en forma afirmativa).

Nombre	Código
Objetivo	
Modelo	
Descripción	
Entradas	Salidas
Pre-condiciones	Post-condiciones

Figura 5-18: Plantilla de especificación de proceso.

Por otro lado, al diseñar la plantilla también se ha buscado tener presente el principio W5H (*What, Why, Where, Who, When, How*). Este principio indica que la información es completa si permite responder a preguntas que comiencen con qué, por qué, dónde, quién, cuándo y cómo. Particularmente, mediante el campo *descripción* se responde *¿qué se debe hacer?*; a través del campo *objetivo* se responde la pregunta *¿por qué se realiza la actividad?*; a su vez, con el campo *modelo* (vistas de comportamiento y organizacional) se responden las preguntas *¿cuándo*, es decir en qué orden, se realizan las tareas? y *¿quiénes* deben llevarlas a cabo? Los campos *entradas* y *salidas*, como así también la vista funcional incluida en el campo *modelo*, también permiten responder *¿qué* productos de trabajo son necesarios para realizar la actividad? y *¿qué* se va a producir? Por otro lado, la pregunta *¿cómo* se debe llevar a cabo la actividad?, si bien no es tenida en cuenta

en la plantilla por no estar dentro del alcance de esta tesis, se responde mediante la capacidad de métodos y herramientas de la estrategia considerada.

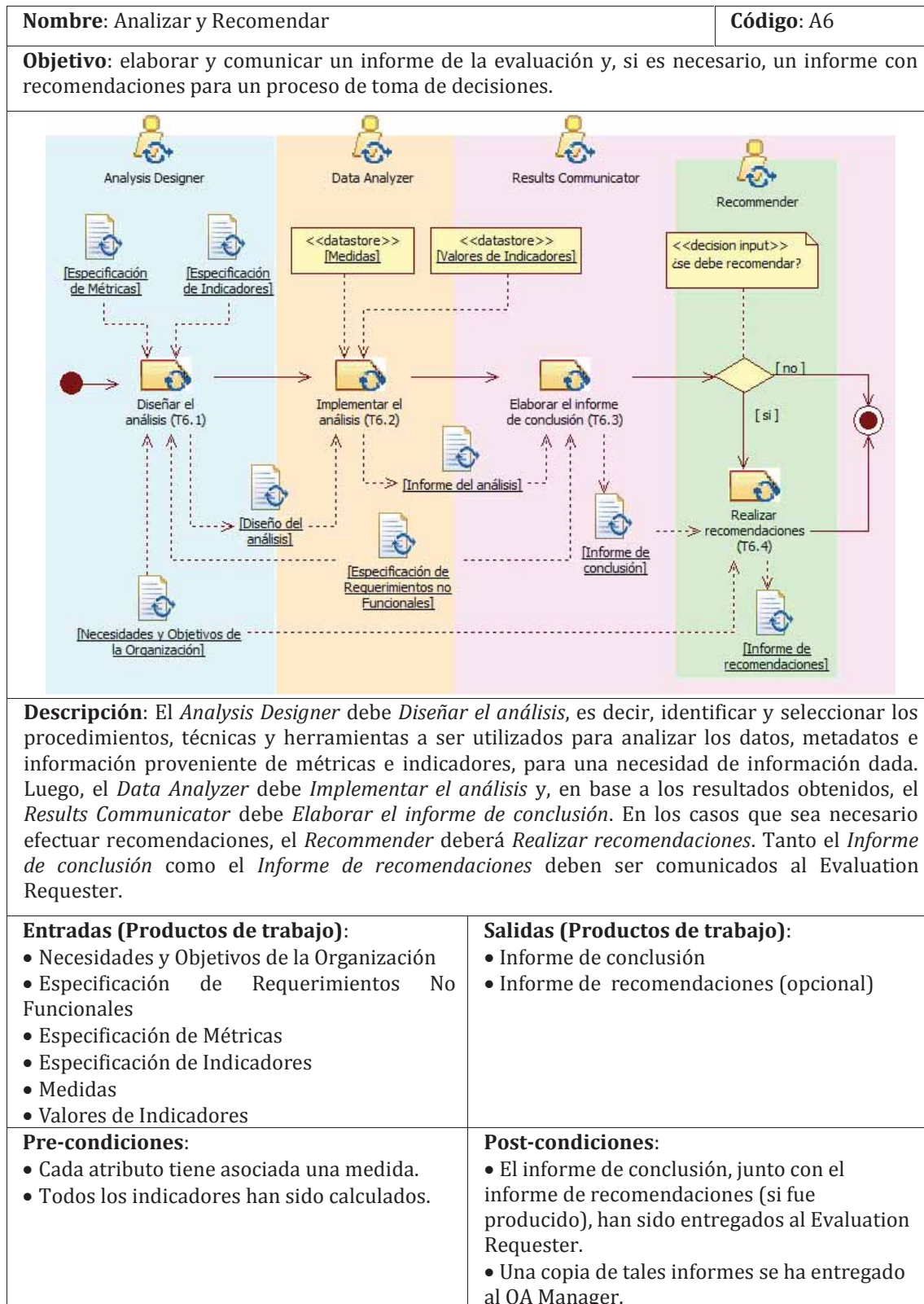


Figura 5-19: Especificación de la actividad Analizar y Recomendar (A6) utilizando la plantilla propuesta.

Cabe mencionar que, de acuerdo al principio de separación de intereses, las diferentes vistas deberían ser modeladas de manera separada [Ge, y otros, 2008].

Claramente, esto permite enfocarnos sólo en ciertos aspectos del proceso. Sin embargo, poder analizar el proceso desde diferentes perspectivas a la vez permite tener una visión más real, completa, detallada e integrada del proceso. Por lo tanto, en la plantilla propuesta se hace uso de modelos que integren varias vistas. En la Figura 5-19 se muestra la especificación de la actividad *Analizar y Recomendar* (A6) de GOCAME mediante nuestra plantilla de especificación de procesos.

En resumen, la plantilla permite especificar procesos (como así también actividades y tareas) de manera completa y sintética al mismo tiempo. Además, provee a los interesados una visión más práctica, consistente e integrada del proceso.

5.5. Conclusiones

Al comienzo de este capítulo se ha presentado la estrategia integrada de M&E SIQinU. A diferencia de GOCAME (Capítulo 4), SIQinU es una estrategia de propósito específico que permite conocer y mejorar el nivel de Calidad en Uso de una aplicación. En líneas generales permite detectar problemas de uso reales, mapearlos a atributos de Calidad Externa y luego, mediante acciones de mejora sobre estos atributos de la aplicación, incrementar la Calidad en Uso.

Al igual que con GOCAME, se ha puesto fuerte énfasis en la especificación del proceso (Sección 5.2). Si bien SIQinU reutiliza la mayoría de las actividades y tareas de GOCAME, y son pocas las que añade, el proceso es considerablemente diferente. Por lo tanto las vistas funcionales y de comportamiento fueron utilizadas para mostrar este contraste. Además, se han identificado los roles y productos de trabajo propios de SIQinU.

También se han considerado, aunque en menor detalle, las otras dos capacidades, a saber: el marco conceptual y los métodos y herramientas. Además, teniendo presente que SIQinU se deriva de GOCAME, se han mostrado diferencias y similitudes entre ambas estrategias de M&E al considerar sus capacidades (Sección 5.3). Cabe mencionar, que así como SIQinU se ha derivado de GOCAME, también se pueden desarrollar otras estrategias de propósito específico derivadas de la estrategia multi-propósito GOCAME.

Finalmente, se presentó una plantilla de especificación de proceso basada en el principio W5H y que integra descripciones textuales junto con modelos gráficos. Considerar estos aspectos contribuye a que las organizaciones puedan contar con mecanismos de documentación de sus procesos que sean más entendibles y comunicables (Sección 5.4). En este sentido es importante mencionar que también existen herramientas software, como por ejemplo EPFComposer [Eclipse]. Sin embargo esta herramienta aún no permite la integración de las diferentes vistas tal como se propone en nuestra plantilla.

En el próximo capítulo, con el fin de mostrar la aplicabilidad de SIQinU, se presenta un caso de estudio realizado en un contexto de empresa en el área de testing [Lew, y otros, 2012]. Además, se pondrá énfasis en los diferentes productos de trabajo producidos a lo largo del proceso de SIQinU.

Capítulo 6: Caso de Estudio usando SIQinU

En este capítulo se muestra la aplicación del proceso de SIQinU en un caso real. Además se ilustran los productos de trabajo más relevantes producidos a lo largo del proceso.

6.1 Introducción

El caso de estudio fue realizado sobre JIRA¹⁷, una WebApp para el seguimiento de errores e incidentes y para la gestión operativa de proyectos. Esta herramienta fue desarrollada por la empresa australiana Atlassian (www.atlassian.com/es), la cual cuenta con 30.000 clientes en más de 135 países. Los productos de Atlassian, entre ellos JIRA, no son de código abierto en su mayor parte, pero son vendidos con una licencia que permite a sus usuarios ver y modificar el software siempre y cuando no lo redistribuyan o utilicen con fines comerciales. Este no es un dato menor ya que SIQinU apunta a evaluaciones no intrusivas. Por lo tanto, es posible introducir código o reparametrización en JIRA para registrar de una manera no intrusiva el comportamiento real del usuario al realizar sus tareas cotidianas utilizando la WebApp.

El estudio se realizó sobre 50 usuarios principiantes pertenecientes al departamento de testing de software de una empresa internacional llamada XBOsoft (www.xbosoft.com/about_us/management_team/), con sede principal en Pekín, China, que se especializa en la calidad y testing de software. Durante la evaluación, los usuarios trabajaron con JIRA en su rutina diaria de testing y de reporte de defectos en su entorno habitual de trabajo. Cabe destacar que en XBOsoft existen diferentes categorías de usuarios que utilizan JIRA, por ejemplo *Test Manager*, *QA Manager* y *Administrator*. Sin embargo, el tipo predominante de los usuarios de JIRA en XBOsoft son del tipo *Tester*. Por lo tanto la evaluación se realizó desde el punto de vista de un *Tester* principiante [Lew, y otros, 2012].

Por otro lado, la tarea más común de un *Tester* al utilizar JIRA es ingresar un nuevo defecto (tarea denominada originalmente *Entering a new defect*). Por ende, se ha decidido evaluar esta tarea para mejorarla. Particularmente, se desea mejorar la Usabilidad de la herramienta en la ejecución de dicha tarea.

A continuación, se comenta el proceso de SIQinU (Capítulo 5) aplicado a la evaluación y mejora de la aplicación JIRA, focalizándonos principalmente en los *productos de trabajo* más relevantes del caso de estudio.

¹⁷ Origen del nombre JIRA: Para el seguimiento de fallos, el equipo de desarrollo de JIRA utilizaba originalmente la aplicación Bugzilla (www.bugzilla.org), al cual llamaban por el nombre japonés de Godzilla, a saber: "Gojira". A medida que el equipo desarrolló su propio gestor de fallos, y luego se convirtió en un gestor de incidencias, el nombre quedó, pero sin Go. De ahí JIRA.

6.2 Fase I: Specify Requirements and Evaluation Criteria for QinU

Las siete actividades de la Fase I, así como la secuencia en que deben realizarse, pueden observarse en la Figura 5-3. La primera de ellas, *Establish Information Need*, produce el documento *Information Need Specification*. Este documento contiene la especificación concisa del punto de vista del usuario, la categoría (objeto) y la entidad que será analizada, y el foco de calidad a evaluar. Las tareas a realizar son (Figura 5-4): *Definir el punto de vista del usuario*, *Establecer el objeto*, *Establecer entidad* e *Identificar el foco*.

<p>Information Need Specification Purpose: understand and improve User viewpoint: beginner tester Entity Category under analysis: defect tracking WebApp Entity: JIRA Evaluation Focus: actual usability Related Calculable Concepts (Characteristics): effectiveness in use, efficiency in use, and learnability in use</p>

Figura 6-1: Artefacto *Information Need Specification*.

Para el caso de estudio JIRA el punto de vista del usuario es “*beginner tester*”¹⁸, la categoría de entidad (objeto de estudio) fue “*defect tracking WebApp*” y la entidad estudiada fue “*JIRA*”. El foco (concepto calculable) evaluado es “*actual usability*” y sus subcaracterísticas; “*effectiveness in use*”, “*efficiency in use*” y “*learnability in use*”. Notar que en la Figura 6-1 se indica que el propósito es “*understand and improve*” (conocer y mejorar), sin embargo no existe una tarea en SIQinU para definirlo ya que el propósito siempre es el mismo; a diferencia de GOCAME.

Siguiendo la secuencia de actividades de Ph I (Figura 5-3), a continuación se llevó a cabo la actividad *Specify Project Context* (recordar las tareas y su secuencia en la Figura 4-6). El resultado de esta actividad es el documento *Context Specification*. Para el caso de estudio, las propiedades de contexto registradas estuvieron relacionadas a las características de las computadoras utilizadas (tal como procesador, memoria RAM, dimensiones de la pantalla, etc.), además del sistema operativo, el navegador web usado para ejecutar JIRA y el ancho de banda de la red, entre otras. Si bien los valores de las propiedades de contexto no se muestran, durante la evaluación, los usuarios *testers* trabajaron en el departamento de *testing* de XBOsoft utilizando las mismas computadoras que usan habitualmente, las cuales reúnen los requisitos técnicos para ejecutar JIRA. Por otro lado, el ancho de banda de la red era suficiente para el trabajo y no fue un factor influyente en el desempeño.

Una vez especificada la información del contexto, se procedió a llevar a cabo la actividad *Design Tasks*. Esta consiste en escoger y detallar aquellas tareas de usuario más comunes y representativas y que, al mismo tiempo, permitan recolectar suficientes datos de uso. En el caso de JIRA, teniendo presente las necesidades del rol *Requester*, se escogió solo una tarea, a saber: *Entering a new defect*, la cual a su vez consta de 5 sub-tareas: 1) *Add summary, steps and results*; 2)

¹⁸ Se consideró beginner (principiante) a aquellos usuarios con menos de 3 meses de experiencia utilizando JIRA.

Add detail info; 3) Add environment info; 4) Add version info; y 5) Add attachment. Al finalizar esta actividad se produjo el documento *Tasks Specification* ejemplificado en la Figura 6-2. Este artefacto, tal como se especificó en la Figura 5-6, posee una lista de las tareas de usuario a evaluar y la especificación de cada una de las tareas listadas. En la Figura 6-2 se observa la lista de tareas seleccionadas (sección *Tasks List*), la cual solo contiene la tarea *Entering a new defect*, y el detalle de esta tarea en la sección *Task Specification*. Los detalles incluyen las sub-tareas asociadas a la tarea, el orden en el que deben ejecutarse, una descripción de las mismas y las pantallas asociadas de JIRA.

<i>Tasks Specification</i>		
Tasks List		
Task name: <i>Entering a new defect</i>		
Description: ...		
Objective: ...		
Task Specification for Entering a new defect		
Sub-taks: 1) Add summary, steps, and results; 2) Add detail info; 3) Add environment info; 4) Add version Info; 5) Add attachment		
Sub-task order	Sub-task Description	Associated Screens
1	<i>Add summary, steps and results:</i> In this sub-task, the user enters the summary of the defect, steps to produce the defect and results, expected and actual.	Initial Defect Entry Screen (SC1-1); Summary, Steps and Results Screen (SC1-2); Summary, Steps, Results Summary Screen (SC1-3)
2	<i>Add detail info:</i> In this sub-task, users enter detailed information about the defect.	Add Detail Info Screen (SC2-1); Detail Summary Screen (SC2-2)
3	<i>Add environment info:</i> In this sub-task, users enter the environment information such as the operating system and browser.	Add Environment Info Screen (SC3-1); Environment Summary Screen (SC3-2)
4	<i>Add version info:</i> In this sub-task the user enters the version of the software being tested and other associated information.	Add Version Info Screen (SC4-1); Version Summary Screen (SC4-2)
5	<i>Add Attachment:</i> In this sub-task the user attaches a file, if necessary to describe the defect in greater detail. This usually is a screenshot showing the defect while the application is in use.	Add Attachment Screen (SC5-1); Attachment Summary Screen (SC5-2)

Figura 6-2: Ejemplo del documento *Tasks Specification* producido en la actividad *Design Tasks*.

A continuación el *NFR Manager* procedió a realizar la actividad *Establish QinU Requirements Tree*. Siguiendo las tareas de la Figura 4-7, primero se seleccionó un modelo de concepto relacionado al concepto foco, el cual recordamos es *Actual Usability* en el caso de estudio JIRA (ver *Information Need Specification* en la Figura 6-1). Luego, se editó el modelo seleccionado, agregando y eliminando adecuadamente características y atributos, dando lugar al árbol de requerimientos (*QinU NFR Tree*) instanciado según se observa en la Figura 6-3.

1. Actual Usability
 - 1.1. Effectiveness in use
 - 1.1.1. *Sub-Task Correctness*
 - 1.1.2. *Sub-Task Completeness*
 - 1.1.3. *Task Successfulness*
 - 1.2. Efficiency in use
 - 1.2.1. *Sub-Task Correctness Efficiency*
 - 1.2.2. *Sub-Task Completeness Efficiency*
 - 1.2.3. *Task Successfulness Efficiency*
 - 1.3. Learnability in use
 - 1.3.1. *Sub-Task Correctness Learnability*
 - 1.3.2. *Sub-Task Completeness Learnability*
 - 1.3.3. *Task Successfulness Learnability*

Figura 6-3: *QinU NFR Tree* para el caso JIRA.

Cabe mencionar que las características ‘*effectiveness in use*’ y ‘*efficiency in use*’ están definidas en ISO 25010 [ISO/IEC, 2011], mientras que ‘*learnability in use*’ está definida en [Lew, y otros, 2010]. Por otro lado, los nombres de los atributos del árbol de requerimientos de la Figura 6-3 incluyen los términos “*correctness*” y “*completeness*”, por lo tanto, fue necesario definir qué se entiende por “correcto” y “completo” en el contexto del caso de estudio JIRA. La Tabla 6-1 muestra las definiciones utilizadas. En el Apéndice B de [Lew, 2010] se definen todos los atributos.

Tabla 6-1: Definiciones de correcto e incorrecto usadas para el caso de estudio.

	Correcto	Incorrecto
Completo	Se han ingresado todos los datos y el <i>test leader</i> no encuentra errores en los datos ingresados por el <i>beginner tester</i> .	Se han ingresado todos los datos, pero el <i>test leader</i> encontró errores.
Incompleto	No se han hallado errores en los datos ingresados, pero existen datos que no han sido ingresados por el <i>beginner tester</i> , ya sea porque se olvidó de completarlo o supuso que no era necesario.	El <i>beginner tester</i> no ingresó todos los datos y se hallaron errores en los datos ingresados.

La siguiente actividad de Ph I es *Design QinU Measurement*. En esta, el *Metrics Expert* identificó una métrica para cada atributo del árbol de requerimientos. Las métricas fueron seleccionadas a partir de un catálogo de métricas, el cual contenía más de 30 métricas indirectas y sus métricas directas asociadas. En la Figura 6-4 se ilustra la métrica indirecta seleccionada para el atributo *Sub-task Correctness* (codificado 1.1.1 en la Figura 6-3). Otras métricas usadas en JIRA pueden encontrarse en el Apéndice C de [Lew, 2010].

Una vez que se especificaron todas las métricas que se van a utilizar, el *Indicators Expert* diseñó, en la actividad *Design QinU Evaluation*, un indicador para cada atributo y característica (concepto calculable) del árbol de requerimientos. Siguiendo el flujo de actividades descrito en la Figura 4-16, primero se diseñaron los seis indicadores elementales y luego los cuatro indicadores derivados.

En JIRA se optó por diseñar todos los indicadores (elementales y derivados) con una escala porcentual (%) y con tres rangos o niveles de aceptabilidad

(criterios de decisión)¹⁹, a saber: un valor entre 0-70% (rango insatisfactorio -rojo) significa se deben realizar urgentemente los cambios; un valor entre 70-90% (rango marginal –amarillo) indica que se necesitan hacer cambios; y un valor entre 90-100% (rango satisfactorio -verde) indica que no es necesario hacer cambios en el atributo o concepto calculable analizado.

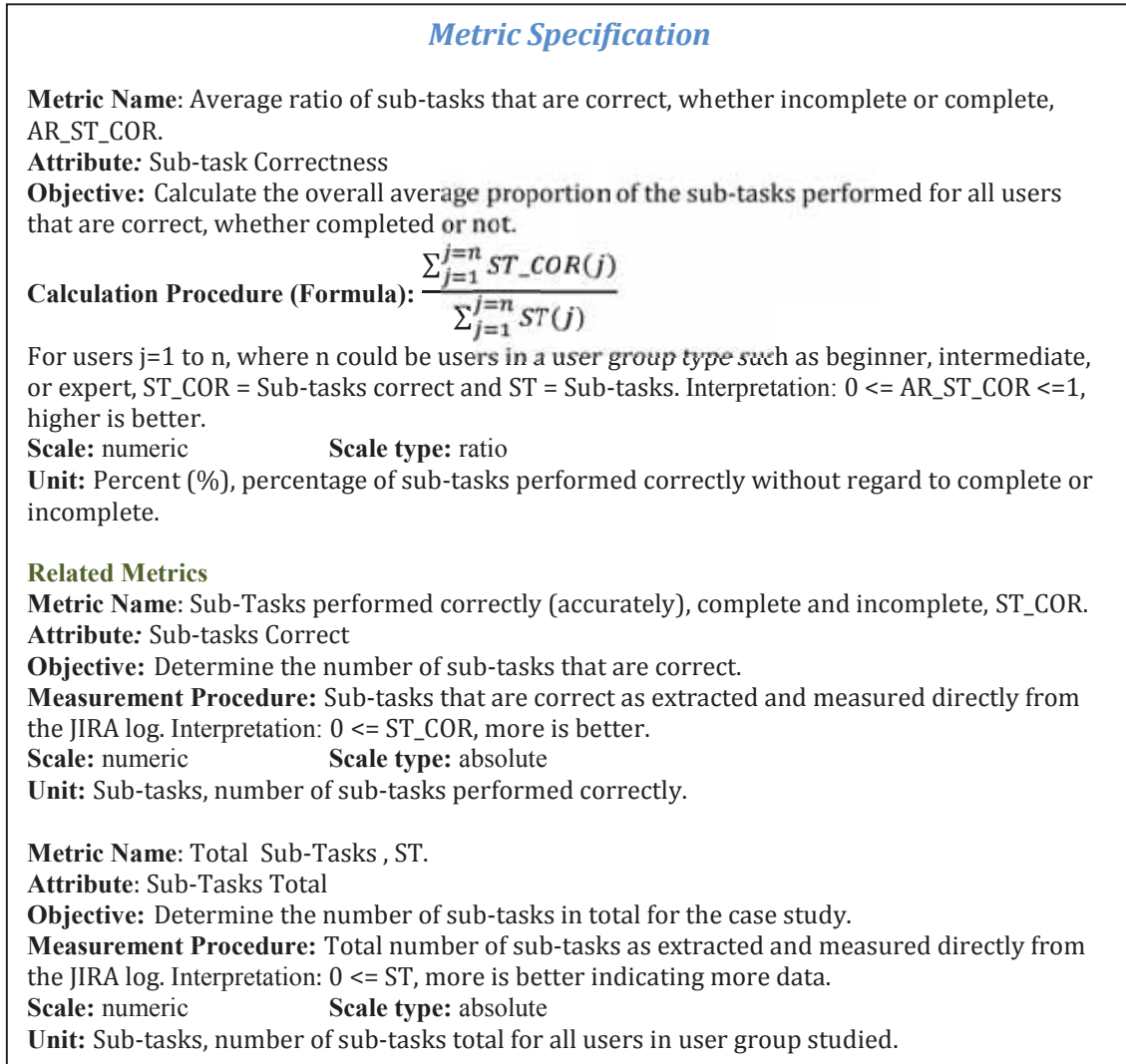


Figura 6-4: Artefacto *Metric Specification*.

Por otro lado, también se decidió utilizar el mismo modelo de agregación lineal aditivo para calcular los indicadores derivados. Debido a que el caso de JIRA fue un estudio exploratorio, los pesos de los elementos (valores de los indicadores de menor nivel en el árbol) fueron iguales. Sin embargo, podrían haberse utilizado diferentes pesos para reflejar diferentes niveles de importancia entre los elementos. Por ejemplo, para *Effectiveness in use*, algunas organizaciones podrían dar más peso a los elementos relacionados a la correctitud que a los asociados a la completitud. O en una aplicación para contaduría, por ejemplo, podrían tener mayor peso aquellos elementos asociados a *Accuracy* (precisión).

¹⁹ Los niveles de aceptabilidad podrían ser diferentes para cada indicador dependiendo de las necesidades del *Evaluation Requester*.

Por último, en la actividad *Design Preliminary Analysis*, en primer lugar se optó por realizar un *ranking* de los atributos que necesitan ser mejorados, listados en orden decreciente, es decir, los primeros deben ser mejorados con mayor urgencia. También se decidió realizar una tabla en la cual se pueda observar el nivel de satisfacción alcanzado por cada sub-tarea, lo cual permitirá conocer cuáles son las sub-tareas (y pantallas) de la tarea evaluada que deben considerarse para mejorar la calidad. Por último, mediante el análisis del nivel de satisfacción alcanzado por cada sub-tarea, se hipotetizarán problemas de la Calidad Externa que probablemente afectaron el nivel de la Calidad en Uso.

6.3 Fase II: Perform QinU Evaluation and Analysis

Las actividades de esta fase permitirán cumplir parte del propósito de SIQinU, a saber: conocer el nivel de satisfacción de Calidad en Uso (específicamente de *Actual Usability*) de JIRA. Para lograrlo se llevaron a cabo las actividades de la Figura 5-8.

```

.
.
### Associate users with user groups by ip address
my %ip_user = ();
my %ip_level = ();
my $sql = qq {select COL_IP, COL_NAME, COL_LEVEL from TBL_ACCOUNT};
my $sth = $dbh->prepare($sql);
$sth->execute();
my ($ip, $name, $level);
.
.
### Read the log file, collect data
$LOGNAME = "atlassian-jira.log";
(-e $LOGNAME) or die "Log file doesn't exist.\n";
open (LOGFILE, "<$LOGNAME") or die "Can't open log file.\n";
while (<LOGFILE>) {
    next if ($_ eq "\n");          # Skip blank lines
    @entry = parseAccessEntry ($_);
    next if (&UnvalidLine ($entry[3]));
}
close (LOGFILE);
.
.
### Mark the completion of the task.
if ($status eq "step4") {
    if (entry[3] eq "secure/CommentAssignIssue.jspa") {
        $status = "finished";
    }
}
.
.
### Generate time object from string
sub genTime {
    my $datetime = shift;
    my ($year, $mon, $mday, $hour, $min, $sec) = split (/[ -:]/, $datetime);
    my $time = timegm($sec, $min, $hour, $mday, $mon, $year);
    return $time;
}

```

Figura 6-5: Parte del código Perl de la herramienta parser usada en JIRA.

En la primera actividad, *Collect Data*, se utilizó una herramienta (*parser*) para extraer los datos relevantes registrados en los archivos de registro (*log files*) y se realizaron agregaciones cuando fue necesario. Aprovechar las capacidades del lado del servidor para registrar ciertos datos permitió determinar objetivamente, por ejemplo, el tiempo que le toma a un usuario realizar una tarea, sin sentir la presión

de un evaluador observándolo. En la Figura 6-5 se muestra una porción del código Perl de la herramienta *parser*. Un extracto más amplio del código se halla en el Apéndice G de [Lew, 2010].

A continuación, se cuantificaron los atributos (actividad *Quantify Attributes*) y luego se calcularon los valores de los indicadores (actividad *Calculate Indicators*) usando respectivamente las métricas e indicadores diseñados en la Fase I. En la Tabla 6-2 se muestran los valores de los indicadores elementales (EI) y derivados (DI) obtenidos al finalizar la actividad *Calculate Indicators*, mientras que en la Tabla 6-3 se presentan los resultados a nivel de sub-tarea (solo para el atributo 1.1.1). Esta última tabla permite identificar más claramente en qué sub-tareas debemos enfocarnos durante el análisis.

Tabla 6-2: Nivel de satisfacción alcanzado por JIRA respecto de Actual Usability (Calidad en Uso).

Characteristics and attributes	EI (%)	DI (%)
1. Actual usability		53,3 ●
1.1. Effectiveness in use		73,2 ●
1.1.1. Sub-task correctness	86,4 ●	
1.1.2. Sub-task completeness	87,9 ●	
1.1.3. Task successfulness	45,5 ●	
1.2. Efficiency in use		29,3 ●
1.2.1. Sub-task correctness efficiency	37,4 ●	
1.2.2. Sub-task completeness efficiency	37,5 ●	
1.2.3. Task successfulness efficiency	13,1 ●	
1.3. Learnability in use		57,3 ●
1.3.1. Sub-task correctness learnability	78,8 ●	
1.3.2. Sub-task completeness learnability	26,4 ●	
1.3.3. Task successfulness learnability	66,7 ●	

Tabla 6-3: Nivel de satisfacción para el atributo *Sub-task correctness* a nivel de sub-tareas.

Beginner Tester	
Sub-task correctness	Rating (%)
Sub-task 1	81,8 ●
Sub-task 2	77,3 ●
Sub-task 3	86,4 ●
Sub-task 4	90,9 ●
Sub-task 5	95,5 ●
Total	86,4 ●

Finalmente, el rol *Data Analyzer* realizó la actividad *Conduct Preliminary Analysis*. Observando los resultados de la Tabla 6-2, se aprecia que todos los atributos y características del árbol de requerimientos poseen un valor menor a 90%, lo cual significa que ninguno alcanza el nivel satisfactorio (recordar los tres criterios de decisión). Incluso seis de los nueve atributos están por debajo de 70% (rango insatisfactorio, rojo), lo cual indica que a estos atributos se les debe realizar cambios urgentes para mejorar la calidad. En la Tabla 6-4 se exhiben los diferentes atributos ordenados según su nivel de satisfacción.

Además, durante el análisis no sólo se observó el nivel de satisfacción de cada atributo, sino que también se examinó el grado de satisfacción alcanzado por cada una de las cinco sub-tareas con respecto a los atributos analizados (recordar Tabla 6-3). Esta información, junto con el documento *Tasks Specification* (ver Figura 6-2), permitió al *Recommender* identificar específicamente en qué sub-tareas y pantallas, los usuarios han tenido mayor dificultad, por ejemplo, un mayor índice de errores, las menos completadas, etc. En la Tabla 6-5 se muestra un extracto del análisis completo, donde se observa que la sub-tarea 1 (Add summary, steps and results) y su pantalla asociada SC1-2 (*Summary, Steps and Results Screen*), obtuvieron un valor 86,4 a nivel de atributo y de 81,8 % a nivel de sub-tarea.

Tabla 6-4: Nivel de satisfacción de los atributos de JIRA analizados para *Actual Usability*.

Ranking by Evaluation	JIRA (%)
1.2.3. <i>Task successfulness efficiency</i>	13,1 ●
1.3.2. <i>Sub-task completeness learnability</i>	26,4 ●
1.2.1. <i>Sub-task correctness efficiency</i>	37,4 ●
1.2.2. <i>Sub-task completeness efficiency</i>	37,5 ●
1.1.3. <i>Task successfulness</i>	45,5 ●
1.3.3. <i>Task successfulness learnability</i>	66,7 ●
1.3.1. <i>Sub-task correctness learnability</i>	78,8 ●
1.1.1. <i>Sub-task correctness</i>	86,4 ●
1.1.2. <i>Sub-task completeness</i>	87,9 ●

En el análisis preliminar también se identificaron las potenciales razones por las que cada atributo obtuvo un bajo nivel de satisfacción (en Calidad en Uso) y se hipotizaron las posibles causas (problemas de Calidad Externa). Por ejemplo, al analizar la pantalla SC1-2, el *Recommender* detectó que no hay ayuda, ni instrucciones, así como tampoco entradas por defecto en los diferentes campos del formulario de entrada, lo cual puede conducir a posibles errores o confusión por parte de un usuario principiante. Estas cuestiones se resaltan en la Figura 6-6. Este examen de la pantalla SC1-2 permitió detectar problemas potenciales en Calidad Externa que posiblemente incidieron en el nivel de Calidad en Uso para el atributo *Sub-task correctness* en la sub-tarea 1. Un extracto de estos detalles se incluyen en la Tabla 6-5.

Tabla 6-5: Extracto del documento *QinU Preliminary analysis report*.

Sub task	Screen ID	Related QinU Attribute	Attribute Evaluation	Sub-task Evaluation	Potential QinU Effect	Potential EQ Problem
1	SC1-2	Sub-task correctness (Effectiveness in Use)	86,40%	81,80%	No instructions could lead to mistakes especially for new user.	Steps, result, expected result have no instructions or examples, difficult for new user.
1	SC1-2	Sub-task correctness (Effectiveness in Use)	86,40%	81,80%	Typing mistakes could be made.	All free form text entry fields with no default information.
2	SC2-1	Sub-task correctness (Effectiveness in Use)	86,40%	77,30%	Possible confusion leading to mistake.	Priority does not match priority on left hand side of screen, no mandatory, no default.
3	SC3-1	Sub-task correctness learnability (Learnability in Use)	78,80%	84,90%	User does not learn if platforms are incorrect.	Some platform combinations don't make sense therefore do not prevent user from entering bad information.

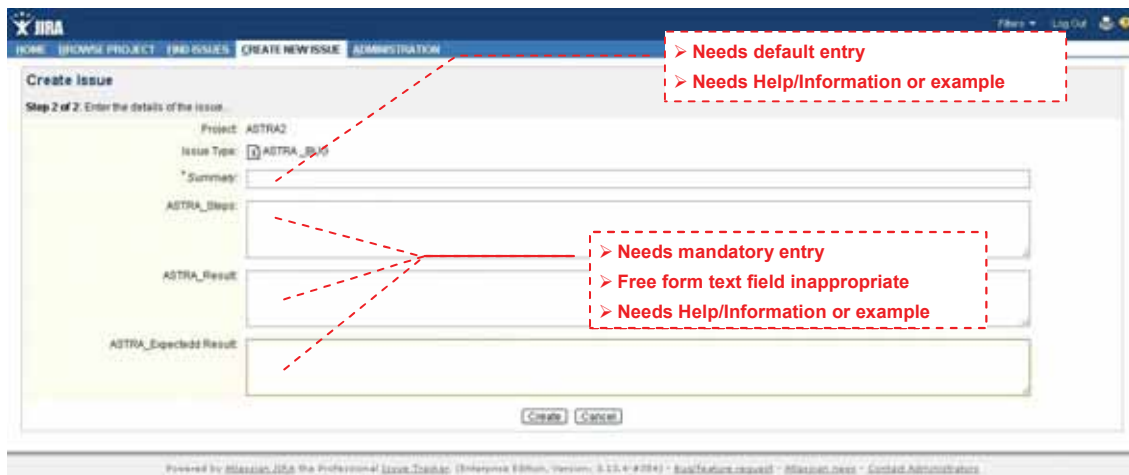


Figura 6-6: Pantalla SC1-2 en la cual se resaltan diversas cuestiones que podrían impactar en el valor del atributo *Sub-task correctness*.

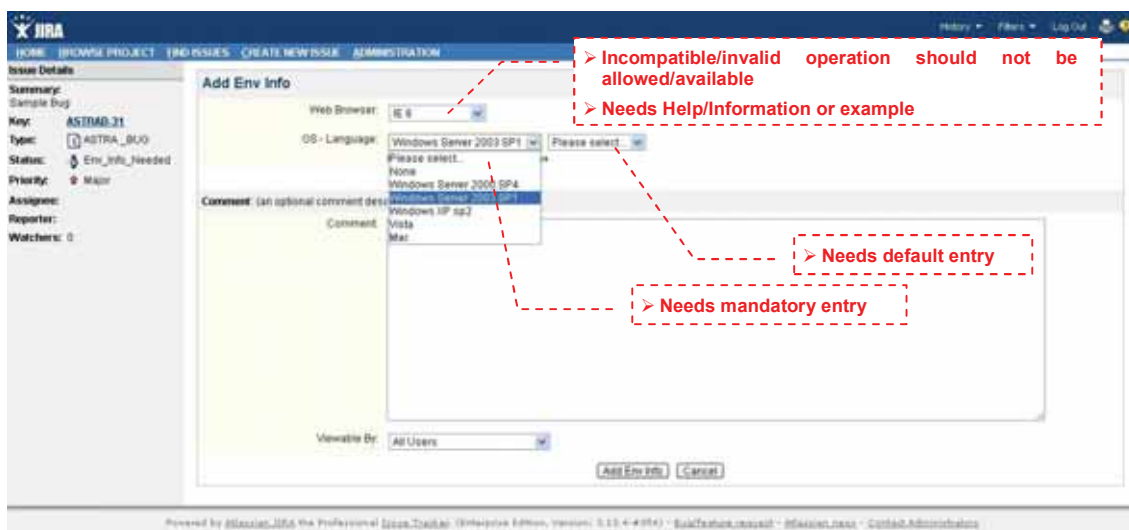


Figura 6-7: Pantalla SC3-1 en la cual se resaltan diversas cuestiones que podrían impactar en el valor del atributo *Sub-task correctness*.

Igualmente, al examinar el atributo 1.3.1 (*Sub-task correctness learnability*), para la sub-tarea 3 (*Add environment info*) y teniendo en cuenta la pantalla asociada SC3-1 (ver Figura 6-7), se encontró -entre otros posibles problemas- que no se cuenta con asistencia o información que permita prevenir el ingreso de plataformas incompatibles. Debido a este hecho, un usuario principiante podría ingresar plataformas no válidas y, sin embargo, no saber que está cometiendo un error.

6.4 Fase III: Derive/Specify Requirements and Evaluation Criteria for EQ

Una vez evaluada la Calidad en Uso de JIRA, y detectados los potenciales problemas de la WebApp, se procedió a la ejecución de Ph III de SIQInU. Esta fase consiste en diseñar una nueva evaluación pero ahora desde el punto de vista de la Calidad Externa considerando la aplicación (no la aplicación en uso). Para ello en primera instancia se llevó a cabo la actividad *Establish EQ Requirements Tree* (ver Figura 5-10). Esta actividad produjo el árbol de requerimientos (*EQ NFR Tree*) de la Figura 6-8.

- 1. External Quality
 - 1.1. Operability
 - 1.1.1. Learnability
 - 1.1.1.1. Feedback suitability
 - 1.1.1.1.1. Navigability feedback completeness
 - 1.1.1.1.2. Task progress feedback appropriateness
 - 1.1.1.1.3. Entry form feedback awareness
 - 1.1.1.2. Helpfulness
 - 1.1.1.2.1. Context-sensitive help availability
 - 1.1.1.2.2. Help completeness
 - 1.1.2. Ease of use
 - 1.1.2.1. Controllability
 - 1.1.2.1.1. Permanence of main controls
 - 1.1.2.1.2. Stability of main controls
 - 1.1.2.2. Error management
 - 1.1.2.2.1. Error prevention
 - 1.1.2.3. Data entry ease
 - 1.1.2.3.1. Defaults
 - 1.1.2.3.2. Mandatory entry
 - 1.1.2.3.3. Control appropriateness
 - 1.2. Information quality
 - 1.2.1. Information suitability
 - 1.2.1.1. Consistency
 - 1.2.1.2. Information coverage
 - 1.2.1.2.1. Appropriateness
 - 1.2.1.2.2. Completeness

Figura 6-8: EQ NFR Tree derivado a partir del QinU NFR Tree y los problemas detectados en JIRA.

Este nuevo árbol de requerimientos fue construido por el *NFR Manager* usando como base el framework de calidad llamado 2Q2U (*External/Internal Quality, Quality in Use, Actual Usability, and User Experience*) [Lew, y otros, 2010], el cual es una extensión del framework de calidad ISO 25010. El artefacto *EQ NFR Tree* se centra en la Calidad Externa (concepto calculable raíz) y cuenta con características como *Operability* e *Information quality* para determinar posibles efectos sobre *Effectiveness in use, Efficiency in use* y *Learnability in use* (características de Calidad en Uso que se desean mejorar).

A continuación, se llevó a cabo la actividad *Design EQ Measurement* para identificar una métrica para cada atributo del nuevo árbol y luego se procedió a realizar la actividad *Design EQ Evaluation* a fin de producir los indicadores para los atributos y conceptos calculables. En la Figura 6-9 se muestra una de las *Metric Specification* producida por el rol *Metrics Expert*. Cabe mencionar que el catálogo de métricas para Calidad Externa contenía alrededor de 25 métricas (ver más detalles en [Lew, 2010]).

En cuanto a los indicadores, el rol *Indicators Experts* acordó con el *Evaluation Requester* utilizar un rango de aceptabilidad diferente al empleado para los indicadores de Calidad en Uso. Los nuevos rangos fueron: un valor entre 0-60 (rango insatisfactorio –rojo) significa que los cambios deben realizarse urgentemente; un valor entre 60-80 (rango marginal –amarillo) indica que se necesitan realizar acciones de mejora; y un valor entre 80-100 implica un rango satisfactorio –verde- para el atributo analizado. Cabe mencionar que para estos indicadores también se podrían haber utilizado un número diferente de rangos, por ejemplo cuatro niveles en lugar de tres.

Metric Specification

Metric Name: Task Navigability Feedback Completeness (TNFC).

Attribute: Navigability feedback Completeness

Objective: Calculate the average of completeness considering the navigational feedback completeness level for all sub-task screens for the given task.

Calculation Procedure (Formula): $TNFC = \sum_{j=1}^{j=n} (\sum_{i=1}^{i=m} NFC_{ij} / m) / n$

for $j=1$ to n , where n is the number of sub-tasks of the given task

for $i=1$ to m , where m is the number of screens for sub-task j

Interpretation: $0 \leq TNFC \leq 3$, more is better.

Scale: numeric

Scale type: ratio

Unit: Completeness level (Note: this metric can be converted to percentage unit, i.e. $TNFC / 0.03$).

Related Metrics

Metric Name: Navigation Feedback Completeness Level (NFC).

Attribute: Screen Navigation Feedback

Objective: Determine the screen meet the criteria for navigation feedback completeness. Note: This metric is similar to the breadcrumb or path capability available in many WebApps.

Measurement Procedure: The screen is inspected to determine the rating (0-3), where evaluators through inspection observe the availability of the previous (backward), current, and next location (forward) mechanism. Screens should support the completeness of this navigation feedback.

Interpretation: $0 \leq ST_COR$, more is better.

Scale: numeric

Scale type: absolute

Allowed Values: (0) has none; (1) has one of them; (2) has two of them; (3) has all of them

Unit: Completeness level.

Figura 6-9: Especificación de la métrica para el atributo *Navigability feedback Completeness* (codificado 1.1.1.1.1 en la Figura 6-8).

6.5 Fase IV: Perform EQ Evaluation and Analysis

Haciendo uso de las especificaciones de métricas e indicadores de Ph III, se llevaron a cabo las actividades de la Figura 5-12. Esta fase es similar a Ph II, pero ahora desde el punto de vista de la Calidad Externa. Además, mientras que en Ph II los valores se obtenían a partir de archivos de registros de uso, en Ph IV la medición y la evaluación se realizan por inspección del experto.

Una vez que todos los atributos fueron medidos por el rol *Data Collector* en la actividad *Quantify Attributes* se calcularon todos los indicadores en la actividad *Calculate Indicators*. En la Tabla 6-6 se aprecian los resultados obtenidos luego de ejecutar esta actividad. Finalmente, el *Data Analyzer* realizó la actividad *Conduct EQ Analysis*, para producir el *EQ analysis report* con la información que permitió identificar, por ejemplo, partes de la WebApp que necesitan mejorarse desde la perspectiva de la Calidad Externa.

Observando la Tabla 6-6 se puede notar, según la evaluación realizada, el bajo nivel de satisfacción (38%) aportado por JIRA. La mayoría de los atributos necesitan ser mejorados urgentemente ya que sus valores caen en el rango insatisfactorio (rojo). Como ejemplos, se pueden citar los atributos *Error prevention* (código 1.1.2.2.1.) y *Context-sensitive help availability* (codificado 1.1.1.2.1). Solo un atributo satisface plenamente el nivel de aceptabilidad requerido (rango satisfactorio, verde), a saber: *Stability of main controls* (1.1.2.1.2).

Tabla 6-6: Nivel de satisfacción alcanzado por JIRA respecto de *External Quality*.

Characteristics and attributes	EI (%)	DI (%)
1.External Quality		38●
1.1. Operability		30●
1.1.1. Learnability		26●
1.1.1.1. Feedback suitability		38●
1.1.1.1.1. <i>Navigability feedback completeness</i>	33●	
1.1.1.1.2. <i>Task progress feedback appropriateness</i>	30●	
1.1.1.1.3. <i>Entry form feedback awareness</i>	50●	
1.1.1.2. Helpfulness		15●
1.1.1.2.1. <i>Context-sensitive help availability</i>	20●	
1.1.1.2.2. <i>Help completeness</i>	10●	
1.1.2. Ease of use		34●
1.1.2.1. Controllability		80●
1.1.2.1.1. <i>Permanence of main controls</i>	60●	
1.1.2.1.2. <i>Stability of main controls</i>	100●	
1.1.2.2. Error management		0●
1.1.2.2.1. <i>Error prevention</i>	0●	
1.1.2.3. Data entry ease		23●
1.1.2.3.1. <i>Defaults</i>	10●	
1.1.2.3.2. <i>Mandatory entry</i>	10●	
1.1.2.3.3. <i>Control appropriateness</i>	50●	
1.2. Information quality		45●
1.2.1. Information suitability		45●
1.2.1.1. Consistency		40●
1.2.1.2. Information coverage		50●
1.2.1.2.1. <i>Appropriateness</i>	50●	
1.2.1.2.2. <i>Completeness</i>	50●	

6.6 Fase V: Recommend, Perform Improvement Actions and Re-evaluate EQ

Considerando los resultados obtenidos en Ph IV, el *Recommender* realiza recomendaciones acerca de aquellos atributos de Calidad Externa que necesitan mejorarse para aumentar la calidad de la WebApp. Una vez que los cambios recomendados fueron llevados a cabo en la versión actual de JIRA, se produjo una nueva versión, la cual se procedió a re-evaluar desde la perspectiva de la Calidad Externa para determinar si los cambios propuestos provocaron una mejora respecto de la versión previa. Las actividades de esta fase pueden ser recordadas observando la Figura 5-13.

La actividad *Recommend Improvement Actions* produjo el documento *Recommendations report*, el cual consta de un conjunto de recomendaciones sobre la WebApp que posiblemente mejoren el nivel de satisfacción alcanzado por los atributos de JIRA. Algunas de las recomendaciones listadas en el *Recommendations report* aparecen en la Figura 6-10.

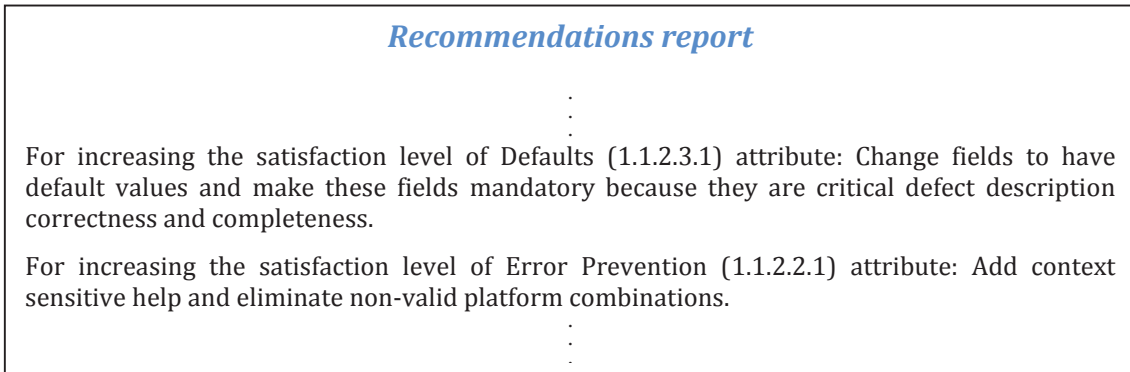


Figura 6-10: Extracto del *Recommendations report* para el caso de estudio JIRA.

A partir de este informe, el rol *Evaluation Requester* puede priorizar unas recomendaciones sobre otras, por ejemplo, poniendo en primer lugar aquellas relacionadas a los atributos con menor nivel de satisfacción. Una vez escogidas las recomendaciones que se tendrían en cuenta, el rol *Maintenance Project Manager* procedió a producir un plan de mejoras (*Improvement Plan*) mediante la actividad *Design Improvement Actions*. En la Figura 6-11 se muestran las acciones a realizar para llevar a cabo las mejoras recomendadas en la Figura 6-10.

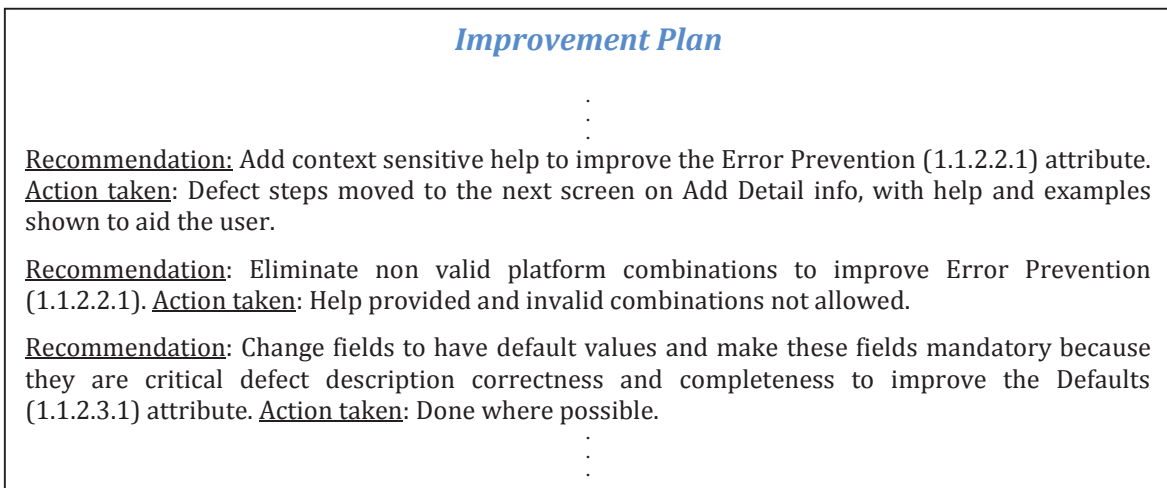


Figura 6-11: Extracto del documento *Improvement Plan* producido en la actividad *Design Improvement Actions*.

Una vez obtenido el plan de mejoras, el equipo de desarrollo de XBOsoft (en el rol de *Developer*) ya está en condiciones de llevar a cabo los cambios correspondientes, resultando así en una nueva versión de JIRA -la cual llamaremos JIRA B para distinguirla de la versión original. Esta nueva versión de JIRA tuvo muchas otras mejoras no mencionadas en el *Improvement Plan* de la Figura 6-11, como por ejemplo la reducción de trabajo mediante la eliminación de una sub-tarea y ubicando juntos los campos más relacionados con el fin de conseguir un diseño general más eficiente de la tarea “*Entering a new defect*”. Por lo tanto JIRA B cuenta solo con 4 sub-tareas en lugar de 5. Debido a que no se pudo tener acceso completo al código fuente de JIRA, el equipo de desarrollo no realizó todos los cambios propuestos en el *Improvement Plan*. Específicamente, la mayor parte de los cambios se efectuaron a través de modificaciones en la configuración de JIRA. Cabe acotar que, si bien no se pudieron realizar todos los cambios recomendados, esto no fue debido a una debilidad de SIQinU, sino a restricciones de la aplicación

que se estudia.

Una vez realizados los cambios, la WebApp fue re-evaluada para determinar qué atributos han mejorado, cuáles no y, además, para comparar el nivel de Calidad Externa alcanzado por JIRA B respecto del conseguido por JIRA en Ph IV. Para ello nuevamente se volvieron a ejecutar las actividades *Quantify Attributes*, *Calculate Indicators* y *Conduct EQ Analysis*. La salida final es un nuevo *EQ analysis report*, en el cual se comparan las dos versiones de JIRA para conocer la ganancia de la mejora. En la Tabla 6-7 se muestra el resultado de la nueva evaluación (JIRA B) y se la compara con la evaluación previa (JIRA).

Tabla 6-7: Comparación de Calidad Externa entre JIRA y JIRA B.

Characteristics and attributes	JIRA		JIRA B	
	EI (%)	DI (%)	EI (%)	DI (%)
1.External quality		38●		74●
1.1. Operability		30●		60●
1.1.1. Learnability		26●		59●
1.1.1.1. Feedback suitability		38●		38●
1.1.1.1.1. Navigability feedback completeness	33●		33●	
1.1.1.1.2. Task progress feedback appropriateness	30●		30●	
1.1.1.1.3. Entry form feedback awareness	50●		50●	
1.1.1.2. Helpfulness		15●		80●
1.1.1.2.1. Context-sensitive help availability	20●		80●	
1.1.1.2.2. Help completeness	10●		80●	
1.1.2. Ease of use		34●		61●
1.1.2.1. Controllability		80●		80●
1.1.2.1.1. Permanence of main controls	60●		60●	
1.1.2.1.2. Stability of main controls	100●		100●	
1.1.2.2. Error management		0●		30●
1.1.2.2.1. Error prevention	0●		30●	
1.1.2.3. Data entry ease		23●		73●
1.1.2.3.1. Defaults	10●		50●	
1.1.2.3.2. Mandatory entry	10●		80●	
1.1.2.3.3. Control appropriateness	50●		90●	
1.2. Information quality		45●		88●
1.2.1. Information suitability		45●		88●
1.2.1.1. Consistency		40●		90●
1.2.1.2. Information coverage		50●		85●
1.2.1.2.1. Appropriateness	50●		90●	
1.2.1.2.2. Completeness	50●		80●	

Como puede observarse, el nivel de satisfacción a nivel de Calidad Externa ha aumentado considerablemente de 38% a 74%, con lo cual JIRA B cae en el rango marginal (amarillo). También se observan incrementos considerables en atributos como *Help completeness* (1.1.1.2.2) y *Mandatory entry* (1.1.2.3.2) que pasaron de un valor de 10% a 80%. También hubo casos en los cuales, si bien aumentó el valor de satisfacción, el incremento no fue lo suficiente para caer en un mejor rango. Por ejemplo, *Error prevention* (1.1.2.2.1) pasó de 0 a 30, pero permaneció en el rango

insatisfactorio (rojo). Por otro lado, también hubo ciertos atributos, como *Task progress feedback appropriateness* (1.1.1.1.2), cuyo nivel de satisfacción no mejoró a pesar de los cambios introducidos. Como conclusión, se puede decir que, en general, los cambios introducidos fueron apropiados y mejoraron la WebApp. La siguiente fase (Ph VI) nos permitirá examinar si estas mejoras en la Calidad Externa afectan la Calidad en Uso de JIRA B al evaluarla en el contexto real de uso.

6.7 Fase VI: Re-evaluate Quality in Use and Analyze Improvement Actions

Finalizada la Fase V, la nueva versión, es decir JIRA B, fue utilizada por el mismo tipo de usuarios y en el mismo entorno de uso que la versión original. Transcurrido un período de 12 semanas se realizó la re-evaluación de la Calidad en Uso. Esta nueva evaluación permitió determinar si los cambios introducidos para mejorar la Calidad Externa tuvieron un impacto positivo en la Calidad en Uso al efectuar la misma tarea.

Las actividades llevadas a cabo fueron las realizadas durante Ph II, a saber: *Collect Data*, *Quantify Attributes* y *Calculate Indicators*. Para la actividad *Collect Data*, es importante mencionar que los usuarios utilizaron JIRA B durante el mismo período de tiempo que se usó la versión anterior. Además, para las actividades *Quantify Attributes* y *Calculate Indicators* se utilizaron las mismas métricas e indicadores diseñados en Ph I. Los valores de los indicadores para JIRA B pueden observarse en la Tabla 6-8.

Tabla 6-8: Nivel de satisfacción alcanzado por JIRA B respecto de *Actual Usability* (Calidad en Uso).

Characteristics and attributes	EI (%)	DI (%)
1. Actual usability		67,0 ●
1.1. Effectiveness in use		86,7 ●
1.1.1. <i>Sub-task correctness</i>	91,9 ●	
1.1.2. <i>Sub-task completeness</i>	95,5 ●	
1.1.3. <i>Task successfulness</i>	72,7 ●	
1.2. Efficiency in use		42,8 ●
1.2.1. <i>Sub-task correctness efficiency</i>	44,3 ●	
1.2.2. <i>Sub-task completeness efficiency</i>	47,3 ●	
1.2.3. <i>Task successfulness efficiency</i>	36,8 ●	
1.3. Learnability in use		71,6 ●
1.3.1. <i>Sub-task correctness learnability</i>	75,1 ●	
1.3.2. <i>Sub-task completeness learnability</i>	77,3 ●	
1.3.3. <i>Task successfulness learnability</i>	62,5 ●	

Finalmente, se llevó a cabo la actividad *Conduct Improvement Actions Analysis* para determinar la ganancia lograda gracias a las mejoras realizadas sobre la WebApp y para identificar las posibles relaciones entre Calidad Externa y Calidad en Uso. A nivel general (*Actual Usability*) se observa que el nivel de satisfacción pasó de 53% (Tabla 6-2) a 67% (Tabla 6-8). Si bien no se alcanzó el nivel satisfactorio (verde), los cambios incluidos aumentaron el nivel de satisfacción. A fin de facilitar la comparación a nivel de atributos, la Tabla 6-9 reúne los valores de

los atributos obtenidos por las dos versiones de JIRA y muestra la diferencia entre los mismos.

Como se observa en la Tabla 6-9, todos los atributos han aumentado su nivel de satisfacción a excepción de *Task Successfulness Learnability* y *Sub-task Correctness Learnability*. Una posible explicación para estos dos casos es que el proceso de aprendizaje no mejoró tanto como se esperaba ya que los usuarios no lograron adaptarse a la nueva versión en el período de 12 semanas de uso, durante el cual se recolectaron los datos para realizar las mediciones. Pero se debe tener presente que el comportamiento de los usuarios puede cambiar con el correr del tiempo, por lo tanto, si el estudio hubiera durado algunas semanas más, seguramente el comportamiento de los usuarios sería diferente y, en consecuencia, las mediciones. Sin embargo, la disminución de los valores de los atributos es pequeña comparada con los aumentos conseguidos en el resto de los atributos. Por ejemplo, *Task successfulness* pasó de 45,5% a 72,7% (incrementando su nivel de satisfacción en un 27,2%) y el atributo *Sub-task completeness learnability* pasó de 26,4% a 77,3% (aumentando así un 50,9%), por citar algunos casos.

Tabla 6-9: Comparación entre JIRA y JIRA B respecto del nivel de satisfacción alcanzado por los atributos de *Actual Usability*.

Attributes	JIRA (%)	JIRA B (%)	Change (%)
1.1.1. <i>Sub-task correctness</i>	86,4 ●	91,9 ●	5,5
1.1.2. <i>Sub-task completeness</i>	87,9 ●	95,5 ●	7,6
1.1.3. <i>Task successfulness</i>	45,5 ●	72,7 ●	27,2
1.2.1. <i>Sub-task correctness efficiency</i>	37,4 ●	44,3 ●	6,9
1.2.2. <i>Sub-task completeness efficiency</i>	37,5 ●	47,3 ●	9,8
1.2.3. <i>Task successfulness efficiency</i>	13,1 ●	36,8 ●	23,7
1.3.1. <i>Sub-task correctness learnability</i>	78,8 ●	75,1 ●	-3,7
1.3.2. <i>Sub-task completeness learnability</i>	26,4 ●	77,3 ●	50,9
1.3.3. <i>Task successfulness learnability</i>	66,7 ●	62,5 ●	-4,2
Average change			13,7

Teniendo en cuenta los aumentos y disminuciones de los diferentes valores, en promedio, se logró aumentar 13,7% el nivel de satisfacción de los atributos. Obviamente, los indicadores muestran que la mayoría de los atributos de JIRA B necesitan seguir mejorándose, pero es claro que ha habido un cambio notable respecto de la versión previa. Considerando que las recomendaciones y cambios hechos en Ph V tuvieron un impacto positivo en el uso real de la WebApp, se pueden recomendar estos cambios a otras tareas de la aplicación JIRA para seguir aumentando su Calidad en Uso. A continuación se listan algunas de las recomendaciones:

- *Add context sensitive help,*
- *Change fields to include most important information before the details,*
- *Make all valid operations available and disable or don't show invalid choices,*
- *Reduce workload by using dropdowns instead of text boxes when possible,*
- *Change fields to have default values and make these fields mandatory when appropriate,*
- *Ensure consistency of information,*
- *Eliminate non valid combinations,*

- *Ensure view is not blocked from drop-down lists,*
- *Let users know the status at all times.*

Por otro lado, SIQinU busca descubrir relaciones entre los atributos de Calidad Externa y los de Calidad en Uso. Identificar estas relaciones no sólo permite seguir mejorando la WebApp estudiada, JIRA en este caso, sino también mejorar los diseños de software en general. Para ello se procedió a llevar a cabo la tarea *Develop EQ/QinU Relationships*, la cual produjo el documento *EQ/QinU Relationships*. Para el caso de estudio se identificaron las relaciones presentadas en la Figura 6-12, la cual muestra, entre otros casos, que el atributo *Help completeness* influye en los atributos *Sub-task completeness learnability* y *Sub-task correctness*.

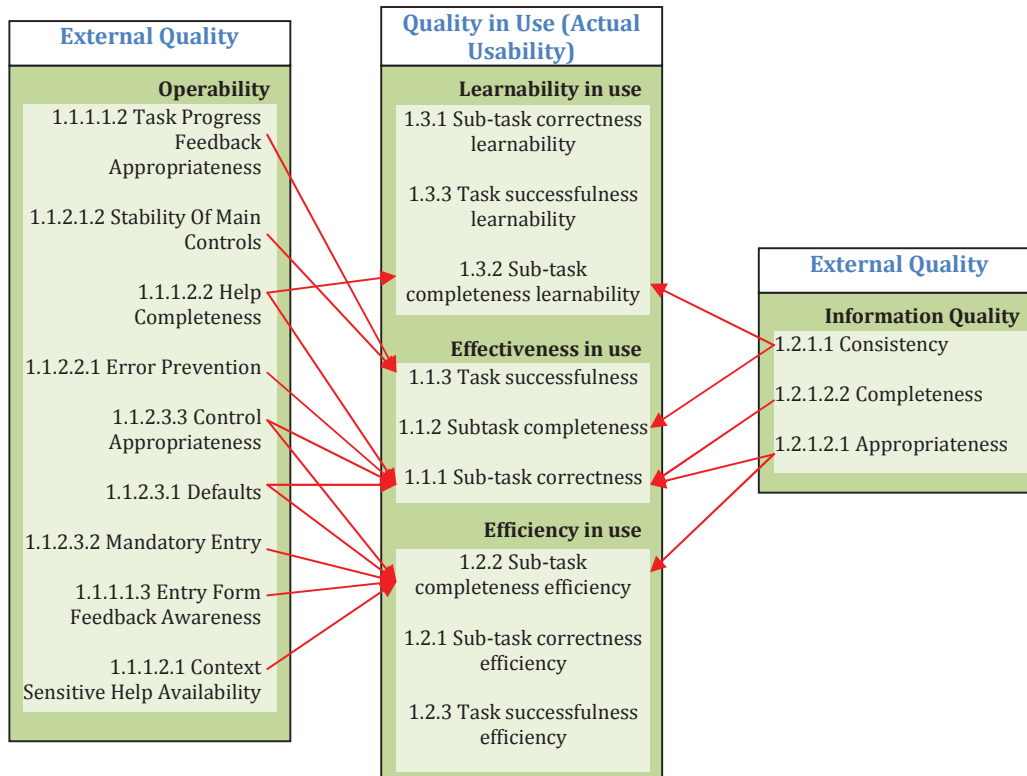


Figura 6-12: Relaciones hipotéticas de influencia entre los atributos de *Calidad Externa* y los de *Calidad en Uso*.

6.8 Consideraciones Finales

A lo largo de este capítulo se ha desarrollado un caso de estudio llevado a cabo en la empresa XBOsoft. Este estudio permitió mostrar la aplicabilidad de SIQinU en un entorno real. Implementar la estrategia SIQinU en esta empresa fue posible, en gran medida, gracias a disponer de las especificaciones de las vistas de proceso detalladas en el Capítulo 5. Los diferentes modelos proporcionaron una guía clara a los agentes involucrados acerca de qué se debía hacer, en qué orden, quiénes eran los agentes, en cumplimiento de roles, más capaces para realizar las tareas, cuáles productos de trabajo debían producirse y/o consumirse, y además, cuál era su estructura.

En el caso de estudio se pudo también observar la necesidad de contar con roles no incluidos en GOCAME, como por ejemplo el *Developer* y el *Project Maintenance Manager*. En la Figura 6-13 se muestra una vista organizacional, donde se aprecian los roles de SIQinU y las fases en las que participan. (Notar que

algunos roles han sido agrupados en uno solo con el fin de no sobrecargar la figura.)

Por otro lado, el caso de JIRA fue ilustrado mostrando los diferentes productos de trabajo producidos en cada fase. Presentar ejemplos reales de los productos de trabajo sirve como material de referencia para aquellas organizaciones que deseen implementar esta estrategia de M&E.

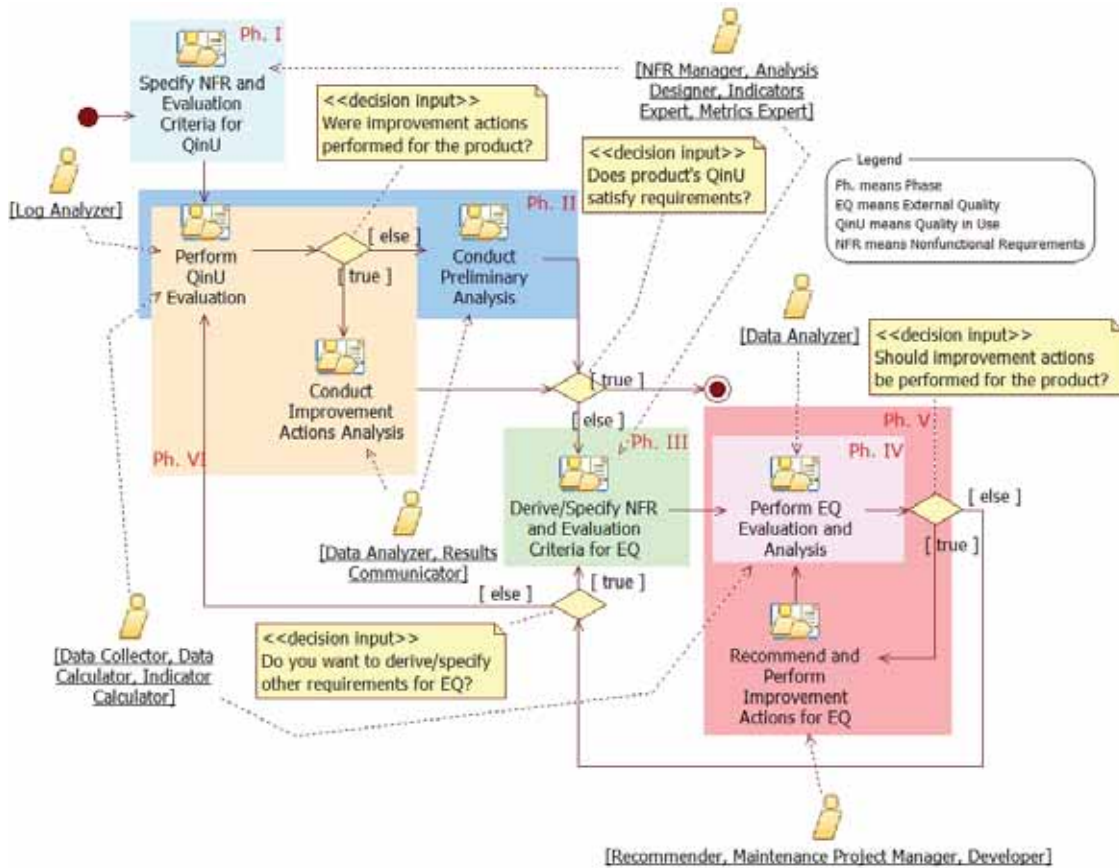


Figura 6-13. Roles involucrados en cada fase de SIQinU.

En la Tabla 6-10 se listan los productos de trabajo producidos en cada fase. Por ejemplo, el artefacto *Tasks Specification* mostrado en la Figura 6-2 es una de las salidas de la Fase I, específicamente de la tarea *Design Tasks* (Sección 5.2.1.3), también mostrada en la Figura 5-3. Este artefacto (Figura 6-2) está en conformidad con la estructura presentada en la Figura 5-6. Es decir, se compone de una lista de tareas (*Tasks List*) y de un conjunto de especificaciones de tarea (*Task Specification*). Otro ejemplo se observa en la Tabla 6-2, donde se muestran los valores de cada indicador producidos en la actividad *Calculate Indicators* (Sección 5.2.2.3) de la Fase II. Cada uno de estos valores es un resultado (*outcome*) que es producido por el rol *Indicator Calculator*, según se aprecia en la Figura 4-29.

Finalmente, cabe acotar que un aspecto a considerar del análisis llevado a cabo en el caso de estudio JIRA es que solo fue posible *hipotetizar* relaciones entre los atributos de Calidad Externa y de Calidad en Uso. El motivo es que al realizarse varios cambios a la vez en Calidad Externa, no se puede determinar exactamente en qué medida afectó cada uno de estos cambios a la Calidad en Uso. Sin embargo, el proceso de SIQinU permite que se puedan realizar hipótesis más precisas acerca de dichas relaciones si se realiza un solo cambio a la vez en Calidad Externa (es

decir, en la Fase V) y luego se evalúan los efectos en la Calidad en Uso (Fase VI). Si bien parece poco práctico, notar que esto es totalmente válido desde el punto de vista del proceso debido a los ciclos de mejora y re-evaluación (recordar Figura 5-1) de SIQinU. Este detalle se puede observar gracias a que el proceso se ha especificado utilizando la vista de comportamiento (ver Figura 5-2).

Tabla 6-10. Productos de trabajo producidos en cada fase de SIQinU.

Fase	Productos de trabajo
Ph. I Specify Requirements and Evaluation Criteria for QinU	<ul style="list-style-type: none"> • Information Need Specification • Context Specification • Tasks Specification • QinU NFR Tree • QinU Metrics and Indicators Specification • Analysis Design
Ph. II Perform QinU Evaluation and Analysis	<ul style="list-style-type: none"> • Parsed Data • Measures and Indicators' values for QinU • QinU Preliminary Analysis Report
Ph. III Derive/ Specify Requirements and Evaluation Criteria for EQ	<ul style="list-style-type: none"> • EQ NFR Tree • EQ Metrics and Indicators Specification • EQ Analysis Design
Ph. IV Perform EQ Evaluation and Analysis	<ul style="list-style-type: none"> • Measures and Indicators' values for EQ • EQ Analysis Report
Ph. V Recommend, Perform Improvement Actions, and Re-evaluate EQ	<ul style="list-style-type: none"> • Recommendations Report • Improvement Plan • New Product Version • New EQ Analysis Report
Ph. VI Re-evaluate QinU and Analyze Improvement Actions	<ul style="list-style-type: none"> • New Measures and Indicators' values for QinU • QinU Analysis Report • EQ/QinU Relationships

Capítulo 7: Conclusiones

En este último capítulo se presentan las conclusiones de esta tesis, abordando los problemas encontrados en el área de M&E y las contribuciones realizadas en este campo.

Además, se exponen futuras líneas de investigación afines a la temática aquí presentada.

7.1. Problemas Observados

Producto de la presente investigación, se han presentado una serie de debilidades y oportunidades de mejora detectadas en los trabajos existentes relacionados al área de M&E. Específicamente, los problemas observados fueron los siguientes:

- ✓ Los enfoques de M&E actuales generalmente no cuentan con tres capacidades básicas, a saber: i) una especificación del proceso de M&E, ii) un marco conceptual de M&E, y iii) una especificación de los métodos para llevar a cabo el proceso. Contar con estas tres capacidades es esencial ya que las mismas permiten desarrollar programas y proyectos de M&E en los cuales los agentes ejecutores saben claramente qué hacer (*proceso*), cómo hacerlo (*métodos y herramientas*) y, además, tienen un entendimiento común del dominio (*marco conceptual*).
- ✓ En los enfoques de M&E que cuentan con las tres capacidades mencionadas, habitualmente no existe una relación entre las mismas y, por lo tanto, no pueden considerarse como estrategias *integradas* de M&E. La relación entre las capacidades es importante ya que, por ejemplo, al utilizar el marco conceptual de M&E para especificar el proceso y los métodos, se brinda consistencia en la terminología empleada y, por ende, se favorece el entendimiento. Además, los métodos y herramientas provistos por la estrategia deben ser aquellos que permitan llevar a cabo las actividades del proceso de M&E.
- ✓ Las estrategias de M&E existentes en la literatura suelen dar pocos detalles en cuanto a su proceso. Por ejemplo, muchas veces no se presenta la secuencia en que deben llevarse a cabo las tareas, o no se indican los productos de trabajo consumidos y producidos. Sin embargo, una robusta especificación del proceso es primordial ya que permite tener una guía al momento de realizar las actividades. Además, contar con la especificación del proceso facilita la definición de métodos y herramientas que permitan llevar a cabo las actividades del proceso, ya sea de forma total o parcialmente automatizada.
- ✓ En las especificaciones del proceso, los productos de trabajo son mencionados pero, a menudo, no existe una plantilla o ejemplos claros que permitan a los usuarios conocer su composición o estructura.
- ✓ Si bien existen diferentes plantillas de especificación de proceso, estas no integran descripciones textuales junto con modelos gráficos. Integrar estos dos

enfoques permite disponer de especificaciones del proceso que tenderán a ser más fáciles de comprender y comunicar.

✓ Revisando la literatura actual de proceso, se ha observado que, a la fecha, no existe un consenso en cuanto a la terminología empleada en el dominio de proceso. Esta falta de acuerdo hace que se sigan utilizando indistintamente algunos términos que tienen significado diferente. Esto provoca imprecisión e inconsistencias en las especificaciones y, por lo tanto, confusión a los lectores, entre otras consecuencias.

7.2. Contribuciones Realizadas

Con el fin de mitigar los problemas encontrados, a lo largo de este trabajo se han realizado los siguientes aportes:

✓ Modelar el proceso de la estrategia de M&E multi-propósito denominada GOCAME y de la estrategia de propósito específico llamada SIQinU. Así, ambas cuentan con las tres capacidades consideradas esenciales, ya que poseen: i) un marco conceptual de M&E, con base ontológica, denominado C-INCAMI, ii) la especificación del proceso de M&E, y iii) un conjunto de métodos y herramientas específicas, como por ejemplo la metodología WebQEM y la herramienta C-INCAMI_Tool.

✓ Con el fin de disponer de una estrategia *integrada* de M&E, al modelar el proceso de las estrategias mencionadas se ha tenido en cuenta el marco conceptual C-INCAMI. De esta manera, los nombres de las actividades, tareas, productos de trabajo y roles involucrados en cada proceso hacen uso de la terminología de M&E proporcionada en C-INCAMI.

✓ En este trabajo nos hemos concentrado, principalmente, en la *especificación del proceso de M&E*, tanto de GOCAME como de SIQinU. Con el fin de proporcionar una visión completa, detallada e integrada de los procesos, estos se modelaron desde diferentes vistas, a saber: funcional, organizacional, de comportamiento e informacional. De esta manera se ha mejorado sustancialmente la especificación del proceso de GOCAME presentado en [Olsina, y otros, 2008b]. Tales vistas, o modelos, permiten conocer qué actividades y tareas deben llevarse a cabo y en qué orden, cuáles son los roles involucrados, qué productos de trabajo son requeridos y cuáles son producidos, así como las relaciones que existen entre algunos productos de trabajo. Por lo tanto, estas estrategias aportan una guía clara y repetible de las actividades que deben llevarse a cabo para definir los requerimientos no funcionales, establecer el contexto y diseñar e implementar la medición y la evaluación, entre otros. Además, contar con la especificación del proceso de M&E utilizando diferentes vistas facilita la comunicación y la realización de las tareas de las partes intervinientes, como así también la selección del personal asociado a las actividades en función del rol que deben cumplir.

✓ Teniendo en cuenta que los modelos deben facilitar la comunicación y el entendimiento, se empleó el lenguaje de modelado SPEM 2.0. Utilizar este lenguaje, estándar ampliamente aceptado por la comunidad facilita el entendimiento del proceso aquí propuesto, además de proveer los mecanismos necesarios para cubrir las diferentes vistas del proceso.

✓ Se han modelado los procesos de GOCAME y SIQinU con una profundidad tal que

permita dejar al descubierto cuáles son todas las tareas involucradas junto a sus entradas y salidas, así como secuencias y otros aspectos relevantes. Este nivel de detalle favorece el aseguramiento de la repetibilidad y reproducibilidad de los procesos.

✓ Dada la necesidad de contar con mecanismos adecuados para la documentación de procesos, se consideró una plantilla que integra el uso de descripciones textuales junto con modelos gráficos del proceso -específicamente de las diferentes vistas. De esta manera, las especificaciones de proceso tenderán a ser más fáciles de comprender y comunicar. Por otro lado, al diseñar la plantilla, también se ha buscado que ésta sea lo más completa posible. Para ello, se tuvo presente el principio W5H (*What, Why, Where, Who, When, How*). Este principio indica que la información es completa si permite responder a preguntas que comiencen con *qué, por qué, dónde, quién, cuándo y cómo*.

✓ Se han presentado dos casos de estudio conducidos en ambientes de trabajo reales con el fin de mostrar la aplicabilidad de las estrategias presentadas y, al mismo tiempo, para proveer ejemplos claros de los productos de trabajo que puedan servir de guías o plantillas para aquellos interesados en utilizar las estrategias.

✓ A raíz de la falta de consenso que existe en la terminología utilizada en el dominio de proceso, se ha desarrollado una nueva ontología de proceso. La misma fue desarrollada teniendo en cuenta fuentes reconocidas en el área de proceso como ISO, CMMI y SPEM. Utilizar los términos actuales más comunes de trabajos ampliamente aceptados como los citados permite que sea más fácil lograr un amplio consenso en la terminología. Por otro lado, la plantilla de especificación de procesos aquí propuesta hace uso de la terminología de la ontología de proceso. De esta manera, las especificaciones de los procesos de GOCAME y de SIQinU cuentan con una terminología consistente desde el punto de vista de proceso y del dominio de M&E.

Como soporte a estas contribuciones, en la Sección 1.4 se listaron las publicaciones que hemos realizado en revistas y congresos nacionales e internacionales.

7.3. Otras Consideraciones

Una cuestión importante a considerar es la validación de los modelos de proceso aquí presentados. Los criterios según los cuales un modelo de proceso puede analizarse se dividen en dos categorías: análisis estático y dinámico. Por un lado, los análisis estáticos controlan la completitud, correctitud y consistencia estructural del modelo de proceso. La prueba de completitud se enfoca en controlar si el modelo incluye toda la información necesaria para alcanzar los objetivos. La correctitud de un modelo implica que el mismo no contiene información contradictoria. Y la consistencia estructural suele ser chequeada usando ciertas reglas, como por ejemplo: si una actividad es refinada, todos los productos de trabajo usados por la actividad tienen que ser utilizados (o al menos alguna de sus partes) por sus tareas. Por otro lado, el análisis dinámico examina el comportamiento del modelo durante su ejecución. Esto incluye, por ejemplo, el control de posibles situaciones de bloqueos (*deadlocks*) y actividades inalcanzables. Para este tipo de análisis se suele usar simuladores de procesos. Una

discusión más detallada sobre análisis de proceso puede hallarse en [Briand, y otros, 1996a].

Por un lado, los casos de estudio presentados, tanto para GOCAME como para SIQinU, dan prueba que estas estrategias son de aplicabilidad en situaciones reales. Por otro lado, en [Olsina, y otros, 2011] se ha diseñado y ejecutado un caso de estudio para conocer y comparar la *calidad de las capacidades* de GOCAME y de GQM+Strategies. Ambas estrategias son consideradas recursos desde el punto de vista de categoría de entidad. Dicho estudio permitió conocer el nivel de satisfacción alcanzado por cada estrategia respecto de las tres capacidades consideradas fundamentales en la Sección 2.1.2. El objetivo de la evaluación era proveer recomendaciones para mejorar GOCAME. Particularmente, la *calidad de la capacidad especificación del proceso* [Papa, y otros, 2010] permitió evaluar cuán adecuadas y apropiadas son las actividades, productos de trabajo, modelos de proceso y sus vistas y la adhesión del proceso a estándares. Algunos de los atributos considerados en esta evaluación están relacionados con la disponibilidad, completitud y la descomposición estructural de los elementos del proceso, como son las actividades y los productos de trabajo. Notar que, según se indicó al inicio de esta Sección, estos aspectos están asociados al análisis estático del proceso.

El caso de estudio citado arrojó para GOCAME, a nivel general, un valor de satisfacción del 66,48% contra 45,89% de GQM+Strategies. Y, específicamente, para la *calidad de la capacidad especificación del proceso*, GOCAME consiguió un nivel de satisfacción del 58,88% mientras que GQM+Strategies obtuvo 54,34%. Estos valores revelaron que ambas estrategias debían dar atención a esta capacidad. En el caso de GOCAME, por ejemplo, se observó que la vista organizacional no estaba presente y que existían actividades que aun no estaban descriptas.

En base a esta evaluación se introdujeron mejoras en la especificación del proceso de GOCAME y se hizo un nuevo estudio registrado en [Papa, 2012]. Los cambios realizados consiguieron que la *calidad de la capacidad especificación del proceso* pase de 58,88% a un nivel de satisfacción del 73,26%. La presente tesis ha tenido en cuenta estas evaluaciones y buscó seguir haciendo frente a las debilidades aun no resueltas del proceso de GOCAME. Actualmente, se está realizando una tesis doctoral en la cual se evaluará nuevamente GOCAME teniendo en cuenta los cambios aquí introducidos en el proceso.

En cuanto al análisis dinámico, no estaba dentro del alcance de esta tesis el desarrollar modelos ejecutables (como se mencionó en la Sección 2.2.4), por lo que no se han abordado análisis de este tipo. Algunos aspectos relacionados a la verificabilidad de los modelos de proceso se abordan al final de la presente Sección.

Por otra parte, es digno de mención que las actividades propuestas para el proceso de M&E de GOCAME están alineadas con las actividades propuestas en los estándares ISO 12207, 14598 y 15539. Sin embargo, existen actividades como “Revisar, aprobar y proveer recursos para las tareas de medición” y “Adquirir e implantar tecnologías de soporte” [ISO/IEC, 2004], que no han sido tenidas en cuenta en el presente trabajo. La razón es que nos hemos concentrado en el proceso de M&E propiamente dicho y no en las actividades de planificación de proyectos de M&E. Por otro lado, el proceso de M&E de GOCAME proporciona una

visión integrada del proceso de Medición (ISO 15539) con el de Evaluación (ISO 14598), no sólo desde el punto de vista de las actividades que componen los procesos sino también desde el punto de vista de los términos empleados al usar el marco conceptual C-INCAMI.

Además, es importante recordar que a partir de GOCAME pueden instanciarse diversas estrategias, específicas para resolver problemas particulares. Estas reutilizan las tareas, métodos, herramientas, roles y productos de trabajo de GOCAME, pero, al mismo tiempo, definen un proceso adaptado a las necesidades específicas del problema que se esté considerando. Un claro ejemplo de esto es SIQinU.

Respecto de la ontología de proceso, si bien se ha presentado la ontología propiamente y se ha mostrado su aplicación a la terminología del dominio de M&E, no se ha considerado el proceso de construcción de la misma. Por lo tanto cabe mencionar que de alguna manera se han seguido las etapas propuestas en METHONTOLOGY [Fernández-López, y otros, 1997], a saber: (1) especificación, (2) conceptualización, (3) formalización, (4) implementación y (5) mantenimiento; aunque aún no se ha realizado una implementación mediante OWL²⁰ u otro lenguaje de ontologías.

En cuanto al enlace entre el marco de M&E con el de proceso, vale aclarar el uso de estereotipos en lugar de relaciones de herencia. La razón es que “unir” los dos marcos conceptuales mediante relaciones de herencia hubiera generado un único marco conceptual, el cual no sería específico del dominio de proceso ni del de M&E. Por lo tanto, se buscó una solución asociada a los Perfiles UML. Un Perfil UML, definido a través de un conjunto de estereotipos, es un mecanismo que proporciona el propio UML para extender su sintaxis y su semántica y, así, expresar conceptos específicos de un determinado dominio de aplicación. UML 2.0 señala varias razones por las que un diseñador puede querer extender y adaptar un metamodelo existente, como por ejemplo la de disponer de una terminología y vocabulario propio de un dominio de aplicación, añadir cierta semántica que no aparece determinada de forma precisa en el metamodelo, o añadir información que puede ser útil a la hora de transformar el modelo a otros modelos o a código.

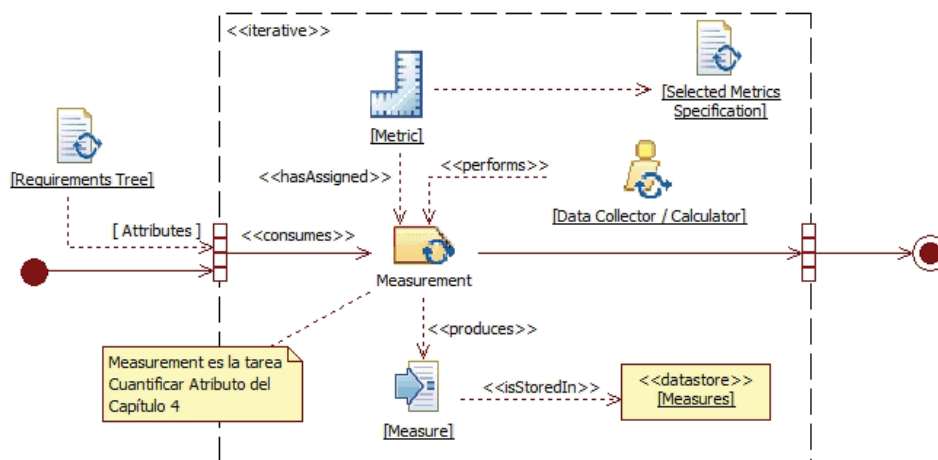


Figura 7-1: Vistas funcional y de comportamiento de la actividad *Medir los atributos*.

²⁰ <http://www.w3.org/TR/owl-overview/>

Por lo tanto, el uso de estereotipos brindaba una solución más adecuada a nuestro problema, ya que el marco conceptual C-INCAMI sigue siendo un marco conceptual específico de M&E y, al mismo tiempo, los estereotipos de proceso le otorgan una mayor semántica a los términos de M&E. Incluso, el disponer de un Perfil de Proceso, permite que sea más fácil adaptar las herramientas existentes de UML para soportar el marco conceptual de proceso o enriquecer otros dominios, aparte del de M&E.

Finalmente, pero no menos importante, vale mencionar que el marco conceptual C-INCAMI enriquecido con la ontología de proceso tuvo un impacto positivo en la capacidad *especificación del proceso*, ya que no sólo asegura uniformidad terminológica sino que también incrementa la verificabilidad de los modelos. Por ejemplo, el modelo de la actividad *Medir los atributos*, representado en la Figura 7-1, puede chequearse contra el marco conceptual C-INCAMI (la Figura 7-2 muestra un extracto del marco original). Observando la Figura 7-1, y sin entrar en muchos detalles, se aprecia que la tarea *Measurement* (la cual representa la tarea *Cuantificar atributo* en la actividad *Implementar la Medición -A3*) consume (*consume*) un atributo (*Attribute*) y produce (*produce*) una medida (*Measure*). Esto es correcto ya que estas mismas relaciones pueden observarse en la Figura 7-2. También, en el modelo de la Figura 7-1 se observa que la medida (*Measure*) es almacenada en (*isStoredIn*) un repositorio de medidas (*Measures datastore*). Teniendo en cuenta que una medida (*Measure*) tiene la semántica del concepto resultado (*Output*), definido como “un producto de trabajo intangible, *almacenable* y procesable” (recordar Tabla 3-3), entonces es correcto que la medida se almacene en un repositorio. De igual manera se puede verificar el resto del modelo.

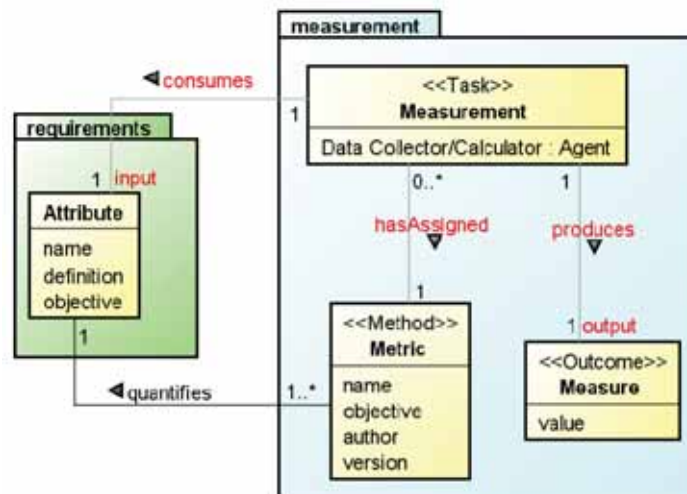


Figura 7-2: Conceptos del marco C-INCAMI involucrados en la medición.

7.4. Trabajos Futuros

A partir de este trabajo son varias las líneas de investigación en las que se pretende avanzar. Una de ellas consiste en la definición de una estrategia llamada GOCAME* (léase GOCAME estrella) que permita enlazar los objetivos de medición a nivel de proyecto (nivel operativo) a los objetivos de alto nivel de la organización (táctico) y de negocios (estratégico), de manera similar a GQM+Strategy. Enlazar y alinear objetivos de diferentes niveles permite que los resultados obtenidos en M&E contribuyan a las decisiones de alto nivel. En esta dirección GOCAME* proveerá mecanismos para dar soporte a la especificación integrada y enlazada de

necesidades de información proveniente de distintos niveles de una organización. Para ello será necesario: i) extender el marco conceptual C-INCAMI para que contemple múltiples instancias de necesidades de información de distintos niveles organizacionales y permita tener en cuenta diferentes focos; y ii) adaptar el proceso de M&E para especificar las actividades que se planifican y ejecutan en los distintos niveles organizacionales (principalmente en la etapa de definición de requerimientos no funcionales) para satisfacer las distintas necesidades de información.

Otra línea de investigación consiste en desarrollar ontologías de Estrategia y de Proyecto. Teniendo presente que una estrategia de M&E puede ser aplicada a proyectos concretos de M&E para hacer frente a diferentes necesidades de información a distintos niveles organizacionales, y que un proyecto puede ser definido como un “esfuerzo planificado, temporal y único que comprende un conjunto de actividades para crear, mantener y evaluar productos de trabajo de acuerdo a recursos y requerimientos específicos”, el lector puede discernir que tanto las ontologías de proceso como de M&E pueden ser reutilizadas en este trabajo futuro.

Una tercer línea consiste en extender el proceso de M&E de GOCAME. Teniendo en cuenta que por lo general la M&E se realiza con el propósito de mejorar la calidad de un producto, el proceso genérico de M&E de GOCAME puede ser extendido con el objetivo de incluir ciclos de mejora y reevaluación luego de las actividades de análisis y recomendación. Por otro lado, actualmente se han definido roles como *Measurement Manager* y *Evaluation Manager*, los cuales no intervienen directamente en la ejecución de las tareas del proceso de M&E descrito, ya que sus tareas están relacionadas a supervisar el trabajo de los *Data Collector*, *Data Calculator* e *Indicators Calculator* y se encargan de las tareas de gestión de sus respectivos proyectos, como puede ser la asignación de recursos. Por lo tanto, el proceso de GOCAME puede extenderse para que incluya actividades de planificación. A su vez, esta línea puede relacionarse con la ontología de Proyecto mencionada anteriormente.

Por último, teniendo presente que a partir de GOCAME se pueden desarrollar estrategias de propósito específico, como SIQinU, en las cuales el proceso varía pero algunas de sus actividades se mantienen, otro posible trabajo futuro está relacionado a las Líneas de Proceso. Estas surgen de la aplicación del enfoque de Líneas de Productos Software en los procesos de software [Bayer, y otros, 1999; Clements, y otros, 2002]. El adentrarnos en este campo y aprovechar los mecanismos de variabilidad permitirá desarrollar procesos de M&E adaptados a cada proyecto y organización, como así también reutilizar y propagar las mejores prácticas.

Referencias

- Abran, A., & Sellami, A. (2002). Initial modelling of the measurement concepts in the ISO vocabulary of terms in metrology. In Proc. of 12th International Workshop on Software Measurement (IWSM2002). Magdeburg, Germany.
- Abran, A., Buglione, L., & Sellami, S. (2004). Software Measurement Body of Knowledge – Initial Validation using Vincenti's Classification of Engineering Knowledge types. 14th International Workshop on Software Measurement and Metrik Kongress (IWSM 2004 / MetriKon 2004 Software Metrik Kongress) (pp. 255-270). Konigs Wusterhausen (Germany): Shaker Verlag.
- Acuña, S., Juristo, N., Merona, A., & Mon, A. (2005). A Software Process Model Handbook for Incorporating People's Capabilities. Springer, ISBN 0387244328.
- Acuña, S., De Antonio, A., Ferré, X., López, M., & Maté, L. (2001b). The Software Process: Modeling, Evaluation and Improvement. In S. K. Chang, Handbook of Software Engineering and Knowledge Engineering. Vol.1 (pp. 193-237). World Scientific Publishing Company.
- Acuña, S., & Ferré, X. (2001a). Software Process Modelling. (pp. 1-6). Orlando, Florida, USA: ISBN 980-07-7541-2.
- Ahern, D., Clouse, A., & Turner, R. (2008). CMMI Distilled: A Practical Introduction to Integrated Process Improvement, Third Edition. Addison Wesley Professional.
- Baffini, M., Rivera, M., & Olsina, L. (2006). Sistema Colaborativo de Revisión de Métricas. 3th Engineering Workshop of Software Engineering and Data Bases, XII CACIC. San Luis, Argentina.
- Barcellos, M., Falbo, R., & Dal Moro, R. (2010). A Well-Founded Software Measurement Ontology. Proceedings of the Sixth International Conference FOIS 2010 (pp. 213-226). Amsterdam, The Netherlands: A. Galton and R. Mizoguchi (Eds.). IOS Press.
- Basili, V., Lindvall, M., Regardie, M., Seaman, C., Heidrich, J., Jurgen, M., et al. (2010). Linking Software Development and Business Strategy through Measurement. IEEE Computer, vol. 43, núm. 4, 57-65.
- Basili, V., Heidrich, J., Lindvall, M., Münch, J., Regardie, M., Rombach, H., et al. (2007b). GQM strategies: A comprehensive methodology for aligning business strategies with software. MetriKon 2007, (pp. 253-266). Kaiserslautern, Germany.
- Basili, V., Heidrich, J., Lindval, M., Münch, J., Regardie, M., Rombach, D., et al. (2007a). Bridging the Gap between Business Strategy and Software Development. Proc. Int'l Conf. Information Systems, Association for Information Systems Electronic Library, Montreal .

- Basili, V., Caldiera, G., & Rombach, H. (1994). The goal question metric paradigm. In J. J. Marciniak, *Encyclopedia of Software Engineering* (pp. 528-532 vol. 1). John Wiley & Sons.
- Basili, V., & Rombach, H. (1988). The TAME Project: Towards Improvement-Oriented Software Environments. *IEEE Transactions on Software Engineering*. Vol. 14 nro 6 , 758-773.
- Bayer, J., Flege, O., Knauber, P., Laqua, R., Muthig, D., Schmid, K., et al. (1999). PuLSE: A Methodology to Develop Software Product Lines. *Proceedings of the 5th ACM SIGSOFT Symposium on Software Reusability (SSR99)*, (pp. 122-131). Los Angeles, California, United States.
- Becker, P., Papa, M., & Olsina, L. (2014). Process Conceptual Base for Enriching a Measurement and Evaluation Ontology. En CD de la XVII Conferencia Iberoamericana en Software Engineering (CIbSE'14) (pp. 53-66). Pucón, Chile: ISBN: 978-956-236-247-4.
- Becker, P., Papa, M., & Olsina, L. (2013). Enhancing the Conceptual Framework Capability for a Measurement and Evaluation Strategy. *ICWE 2013 Workshops* (pp. 104-116). Aalborg, Denmark: Springer International Publishing Switzerland, LNCS 8295, Q.Z. Sheng and J. Kjeldskov (Eds.).
- Becker, P., Lew, P., & Olsina, L. (2012). Specifying Process Views for a Measurement, Evaluation and Improvement Strategy. *Advances in Software Engineering, Software Quality Assurance Methodologies and Techniques* , 2012, 27 págs.
- Becker, P., Lew, P., & Olsina, L. (2011). Strategy to Improve Quality for Software Applications: A Process View. En *ACM proceedings of ICSE, Int'l Conference of Software and System Process (ICSSP)* (pp. 129-138). Honolulu, Hawaii, USA: ISBN: 978-1-4503-0730-7, DOI 10.1145/1987875.1987897.
- Becker, P., Molina, H., & Olsina, L. (2010c). Measurement and Evaluation as Quality Driver. *Journal ISI (Ingénierie des Systèmes d'Information)*, Special Issue "Quality of Information Systems", vol. 15/6, Hermes, Lavoisier. Paris, France, DOI:10.3166/ISI.15.6.33-62, ISSN 1633-1311 , pp. 33-62.
- Becker, P., & Olsina, L. (2010b). Towards Support Processes for Web Projects. En LNCS 6385, Springer book: *Current Trends in Web Engineering - 10th International Conference on Web Engineering, ICWE 2010 Workshops*, Daniel and Faca (Eds) (pp. 102-113). Vienna, Austria: ISBN: 978-3-642-16984-7, DOI: 10.1007/978-3-642-16985-4_10.
- Becker, P., Molina, H., & Olsina, L. (2010a). Vista Funcional del Proceso de Medición y Evaluación. En *ASSE'10, Simposio Argentino de Ingeniería de Software (39 JAIIO)* (pp. 294-308). ISSN: 1850-2792.
- Becker, P., Molina, H., & Olsina, L. (2009). Integrando Proceso y Marco de Medición y Evaluación. En *Memorias de la XII Conferencia Iberoamericana de Software Engineering (IDEAS09-CIbSE)* (pp. 261-266). Medellín, Colombia: ISBN 978-958-44-5028-9.
- Becker, P., Olsina, L., & Molina, H. (2008). Modelado de Proceso para un Marco de Medición y Evaluación. En *Actas (CD-ROM) del XIV Congreso Argentino de*

- Ciencias de la Computación (CACIC 2008), Workshop WISBD. Chilecito, La Rioja: ISBN 978-987-24611-0-2.
- Becker, U., Hamann, D., & Verlage, M. (1997). Descriptive Modeling of Software Processes. Proceedings of the 3rd Conference on Software Process Improvement, (p. 15).
- Benali, K., & Derniame, J. C. (1992). Software processes modeling: what, who, and when. Proceedings of the Second European Workshop on Software Process Technology.
- Bendraou, R., Jézéquel, J., Gervais, M., & Blanc, X. (2010). A Comparison of Six UML-Based Languages for Software Process Modeling. IEEE Transactions on Software Engineering (Vol: 36 , Issue: 5) , pp. 662-675.
- Bendraou, R., Sadovykh, A., M.P., G., & Blanc, X. (2007b). Software Process Modeling and Execution: The UML4SPM to WS-BPEL Approach. Proceedings of the 33rd EUROMICRO Conference on Software Engineering and Advanced Applications (EUROMICRO '07) (pp. 314-321). Lubeck, Germany: IEEE Computer Society (ISBN:0-7695-2977-1).
- Bendraou, R., Combemale, B., Crégut, X., & Gervais, M. P. (2007a). Definition of an Executable SPEM 2.0. 14th Asia-Pacific Software Engineering Conference (pp. 390-397). Aichi: IEEE Computer Society.
- Briand, L., Morasca, S., & Basili, V. (2002). An operational process for goal-driven definition of measures. In IEEE Transactions on Software Engineering, vol. 28, número 12, 1106-1125.
- Briand, L., Differding, C. M., & Rombach, H. (1996b). Practical guidelines for measurement based process improvement.
- Briand, L., Höltje, D., & Kempter, H. (1996a). Process modelling guidelines: Version 1.0. Technical report, Centre de recherche informatique de Montréal (CRIM) & Daimler-Benz AG, Montréal, Québec, Canada H3A 2N4, Daimler-Benz Software Quality Project.
- Bringunte, A., Falbo, R. A., & Guizzardi, G. (2011). Using a Foundational Ontology for Reengineering a Software Process Ontology. Journal of Information and Data Management, v. 2 , 511-526.
- Bröckers, A., Lott, C., Rombach, H., & and Verlage, M. (1995b). MVP-L Language Report Version 2 (TR 265/95). University of Kaiserslautern.
- Bröckers, A., Differding, C., Hoisl, B., Kollnischko, F., Lott, C. M., Münch, J., et al. (1995a). A graphical representation schema for the software process modeling language MVP-L (TR 270/95). Germany, Kaiserslautern: Department of Computer Science, Universität Kaiserslautern.
- Clements, P., & Northrop, L. (2002). Software Product Lines: Practices and Patterns. Addison-Wesley.
- CMMI Product Team. (2010). CMMI for Development, Version 1.3 (CMU/SEI-2010-TR-033). Retrieved 05/02/2014, from the Software Engineering Institute, Carnegie Mellon University website: <http://www.sei.cmu.edu/library/abstracts/reports/10tr033.cfm>.

- Colla, P. E. (2009). Gestión de Mejoras en Procesos en Ingeniería de Software con el soporte de modelos sistémicos. Aplicación al sector de desarrollo de software de Argentina. Tesis de Doctorado, Univ. Tecnológica Nacional Regional Santa Fe.
- Conradi, R., Fernström, C., & Fuggetta, A. (1993). A Conceptual Framework for Evolving Software Processes. SIGSOFT Softw. Eng. Notes , 18 (4), 26-35.
- Cooper, J., & Fisher, M. (2002). Software Acquisition Capability Maturity Model (SA-CMM) Version 1.03 (CMU/SEI-2002-TR-010, ADA399794). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.
- Corcho, O., Fernández-López, M., & Gómez-Pérez, A. (2003). Methodologies, tools and languages for building ontologies. Where is their meeting point? Data & Knowledge Engineering 46(1) , 41-64.
- Covella, G., & Olsina, L. (2006). Assessing Quality in Use in a Consistent Way. In ACM proceedings, Int'l Congress on Web Engineering, (ICWE06) (pp. 1-8). San Francisco, USA: ISBN 1-59593-352-2.
- Cree, R., Dean, J., & Jones, C. (2008). Acquisition Measurement (v1.1).
- CTN-ISSI. (2006). NTP-ISO/IEC 12207: Tecnología de la Información. Procesos del Ciclo de Vida del Software. Lima, Perú.
- Curtis, B., Kellner, M., & Over, J. (1992). Process Modelling. Communications of the ACM , 35 (9), 75-90.
- Dal Moro, R., & Falbo, R. (2008). Uma Ontologia para o Domínio de Qualidade de Software com Foco em Produtos e Processos de Software. In Proc. of the 3rd Workshop on Ontologies and Metamodels for Software and Data Engineering (WOMSDE'08), (pp. 37-48).
- Derniane, J., Kaba, B., & Wastell, D. (1999). Software Process: Principles, Methodology and Technology. Lecture Notes in Computer Science, 1500. Springer.
- Differding, C., Hoisl, B., & Lott, C. M. (1996). Technology Package for the Goal Question Metric Paradigm. Kaiserslautern: Internal Report 281/96, AG Software Engineering, .
- Dowson, M. (1993). Software Process Themes and Issues. Proceedings of the International Conference on the Software Process (ICSP'93): Continuous Software Process Improvement, (pp. 54-62).
- Dujmovic, J. (1996). A Method for Evaluation and Selection of Complex Hardware and Software Systems. CMG 96 Proceedings. The 22nd Int'l Conference for the Resource Management and Performance Evaluation of Enterprise.Vol. 1. , (pp. 368-378).
- Eclipse. (n.d.). Eclipse Process Framework Project . Retrieved 2 05, 2013, from http://www.eclipse.org/epf/tool_component/tool_vision.php
- Ellner, R., Al-Hilank, S., Drexler, J., Jung, M., Kips, D., & Philippsen, M. (2011). A FUML-based distributed execution machine for enacting software process models. Proceeding of the 7th European conference on Modelling foundations

- and applications (ECMFA'11) (pp. 19-34). Springer-Verlag Berlin (ISBN: 978-3-642-21469-1).
- Eloranta, L., Kallio, E., & Terho, I. (2006). A Notation Evaluation of BPMN and UML Activity Diagrams. Disponible en http://www.soberit.hut.fi/T-86/T-86.5161/2006/BPMN_vs_UML_final.pdf (accedido en año 2014).
- Esteban, N., & Olsina, L. (2003). Hacia un Catálogo de Actividades para el Desarrollo de Sitios y Aplicaciones Web. En Proceedings del VI Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes Software (IDEAS) .
- Falbo, R., & Bertollo, G. (2005). Establishing a Common Vocabulary for Software Organizations Understand Software Processes. International Workshop on Vocabularies, Ontologies and Rules for the Enterprise, (pp. 25-32). Enschede, The Netherlands.
- Feiler, P. H., & Humphrey, W. S. (1993). Software Process Development and Enactment: Concepts and Definitions. International Conference of Software Process (ICSP) (pp. 28-40). Berlin, Germany: IEEE Computer Society.
- Feng, Y., Mingshu, L., & Zhigang, W. (2006). Spem2xpdl: Towards Spem Model Enactment. The 2006 International Conference on Software Engineering Research and Practice. Las Vegas.
- Fenton, N., & Pfleeger, S. (1997). Software Metrics: a Rigorous and Practical Approach. (2nd Ed). PWS Publishing Company.
- Fernández-López, M., Gómez-Pérez, A., & Juristo, N. (1997). METHONTOLOGY: From Ontological Art Towards Ontological Engineering. Spring Symposium on Ontological Engineering of AAAI, (pp. 33-40). Stanford University, California.
- Fig, K., Mendling, J., & Strembeck, M. (2009). Towards a Usability Assessment of Process Modeling Languages. In: Proc. of the 8th Workshop Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (EPK 2009) (pp. 118-136). Berlin, Germany: CEUR Workshop Proceedings, vol. 554.
- Finkelstein, A., Kramer, J., & Nuseibeh, B. (1994). Software Process Modelling and Technology. Research Studies Press .
- García, F., Piattini, M., Ruiz, F., Canfora, G., & Visaggio, C. A. (2006b). FMESP: Framework for the modeling and evaluation of software processes. Journal of Systems Architecture , pp. 627-639 Vol. 52, Issue 11, November 2006. ISSN 1383-7621.
- García, F., Bertoa, M., Calero, C., Vallecillo, A., Ruiz, F., Piattini, M., et al. (2006a). Towards a consistent terminology for software measurement. Information and Software Technology, 48 (8) , 631-644.
- García, F., Ruiz, F., Bertoa, M., Calero, C., Genero, M., Olsina, L., et al. (2004). An ontology for software measurement. Technical Report UCLM DIAB-04-02-2. Spain: Computer Science Department, University of Castilla-La Mancha.
- Ge, J., Hu, H., & Lü, J. (2008). Order Constraints for Multi-view Software Process Model. Computer Science and Software Engineering, 2008 International Conference on (Volume:2) (pp. 639-642). Wuhan, Hubei: ISBN 978-0-7695-3336-0.

- Gibson, D., Goldenson, D., & Kost, K. (2006). Performance Results of CMMI-Based Process Improvement (CMU/SEI-2006-TR-004). <http://www.sei.cmu.edu/library/abstracts/reports/06tr004.cfm>: Software Engineering Institute, Carnegie Mellon University.
- Goethert, W., & Siviy, J. (2004). Applications of the Indicator Template for Measurement and Analysis. Software Engineering Measurement and Analysis Initiative, Technical Note CMU/SEI-2004-TN-024.
- Goethert, W., & Fisher, M. (2003). Deriving Enterprise-Based Measures Using the Balanced Scorecard and Goal-Driven Measurement Techniques. Software Engineering Measurement and Analysis Initiative.
- Goethert, W., & Hayes, W. (2001). Experiences in Implementing Measurement Programs. Software Engineering Measurement and Analysis Initiative, Technical Note CMU/SEI-2001-TN-026.
- Goldenson, D. R. (2007). Understanding CMMI Measurement Capabilities & Impact on Performance: Results from the 2007 SEI State of the Measurement Practice Survey. Software Engineering Institute, CMMI Technology Conference.
- Gresse, C., Hoisl, B., & Wüst, J. (1995). A process model for GQM-based measurement. Technical report STTI-KL(Software Technologie Transfer Initiative Kaiserslautern) .
- Guizzardi, G., Falbo, R., & Guizzardi, R. (2008). Grounding Software Domain Ontologies in the Unified Foundational Ontology (UFO): The case of the ODE Software Process Ontology. En Proceedings de la XI Conferencia Iberoamericana de Software Engineering (CIBSE 2008), (pp. 127-140).
- Heidrich, J., Munch, J., Riddle, W., & Rombach, D. (2006). People-oriented Capture, Display, and Use of Process Information. In S. S.-S. Acuña, New Trends in Software Process Modeling, Series in Software Engineering and Knowledge Engineering.Vol. 18 (pp. 121-179). Singapore: World Scientific Publishing Company.
- Henderson-Sellers, B., & Gonzalez-Perez, C. (2005). A comparison of four process metamodels and the creation of a new generic standard. Information and Software Technology (Vol. 47, Issue 1), (pp. 49-65).
- Humphrey, W. S. (1989). Managing the Software Process. New York: Addison-Wesley.
- IEEE. (2004). Guide to the Software Engineering Body of Knowledge (SWEBOK) - 2004 Version. Los Alamitos, CA: IEEE Computer Society.
- IRAM. (2006). IRAM-ISO/IEC 14598-1: Tecnología de la Información - Ingeniería de Software - Evaluación del producto de software - Parte 1: Descripción General.
- ISO/IEC. (2011). ISO CD 25010-3: Systems and software engineering. Systems and software Quality Requirements and Evaluation (SQuaRE). System and software quality models.
- ISO/IEC. (2008). ISO 12207: Systems and software engineering - Software life cycle processes.

- ISO/IEC. (2007b). ISO 15939: Systems and software Engineering - Measurement Process (2nd Edition). ISO/IEC. (2004). ISO 12207: Information Technology / Software Life Cycle Processes (1995/Amd 2:2004).
- ISO/IEC. (2007a). ISO/IEC Guide 99: International vocabulary of metrology - Basic and general concepts and associated terms (VIM). International Organization for Standardization.
- ISO/IEC. (2005). ISO/IEC 25000: Software Engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Guide to SquaRE.
- ISO/IEC. (2002b). Systems Engineering – Guide for ISO/IEC 15288 (System Life Cycle Processes).
- ISO/IEC. (2002a). ISO 15939: Software Engineering - Software Measurement Process.
- ISO/IEC. (2001c). ISO 14598: Information Technology - Software Product Evaluation.
- ISO/IEC. (2001b). ISO 14598-5: Information Technology - Software Product Evaluation - Part 5: Process for evaluators. ISO/IEC. (1998). ISO 14598-5: Information Technology - Software Product Evaluation - Part 5: Process for evaluators.
- ISO/IEC. (2001a). ISO 9126-1: Software Engineering Product Quality - Part 1: Quality Model.
- Kasunic, M. (2006). The State of Software Measurement Practice: Results of 2006 Survey (CMU/SEI-2006-TR-009). Pittsburgh: Software Engineering Institute, Carnegie Mellon University.
- Kitchenham, B., Hughes, R., & Linkman, S. (2001). Modeling Software Measurement Data. *IEEE Transactions on Software Engineering*, 27(9), pp. 788-804.
- Lavazza, L. (2000). Providing Automated Support for the GQM Measurement Process. *IEEE Software*, vol. 17, issue 3, pp. 56 - 62.
- Lew, P. (2010). An Integrated Strategy to Understand and Improve Quality in Use (SIQinU) for Web Applications (PhD Thesis). Beihang University, Beijing, China: School of Computer Science & Engineering.
- Lew, P., & Olsina, L. (2013). Relating User Experience with MobileApp Quality Evaluation and Design. *Current Trends in Web Engineering, ICWE Int'l Workshops*, Q.Z. Sheng and J. Kjeldskov (Eds.) (pp. 253–268). Aalborg, Denmark: LNCS 8295, Springer.
- Lew, P., Olsina, L., Becker, P., & Zhang, L. (2012). An Integrated Strategy to Systematically Understand and Manage Quality in Use for Web Applications. *En Requirements Engineering Journal, Special Issue on Quality Requirement Engineering for Systems & Architecting (Vol.17, N. 4)*, 299-330. Springer London, DOI 10.1007/s00766-011-0128-x, ISSN 0947-3602.
- Lew, P., Olsina, L., & Zhang, L. (2010). Quality, quality in use, actual usability and user experience as key drivers for web application evaluation. 10th international congress on web engineering (ICWE2010) (pp. 218–232). Vienne, Austria: Proceedings of the LNCS 6189, Springer.

- Lonchamp, J. (1993). A Structured Conceptual and Terminological Framework for Software Process Engineering. International Conference on the Software Process (ICSP) (pp. 41-53). Berlin, Germany: IEEE Computer Society Press.
- Lott, C., Hoisl, B., & Rombach, H. D. (1995). The use of roles and measurement to enact project plans in MVP-S. Software Process Technology, Lecture Notes in Computer Science Vo. 913 , 30-48.
- Lott, C. M., & Dieter Rombach, H. (1990). A MVP-L1 Solution for the Software-Process Modeling Problem. In Collected Solutions from the 6th International Software Process Workshop , 31.
- McChesney, I. R. (1995). Toward a classification scheme for software process modelling approaches. Information and Software Technology (Vol. 37, Núm. 7) , 363-374.
- McGarry, J., Card, D., Jones, C., Layman, B., Clark, E., Dean, J., et al. (2001). Practical Software Measurement: Objective Information for Decision Makers. Addison-Wesley Professional. ISBN-13: 978-0-201-71516-3.
- METI. (2011). Investigative Report on Measure for System/Software Product Quality Requirement Definition and Evaluation. Software Metrics Advanced Project. Japan: Ministry of Economy, Trade and Industry (METI).
- Ministry of Forests, Lands and Natural Resource Operations. (1996). Process Modelling Guidelines (Guide S47). Data Administration Section of the Information Management Group.
- Molina, H. (2012). Soporte organizacional de medición y evaluación orientada a objetivos y sensible al contexto. La Palata: Tesis Doctoral en la Facultad de Informática de la UNLP. Director: Dr. Luis Olsina, Co-Director: Gustavo Rossi.
- Molina, H. (2005). Soporte al proceso de Medición para el Aseguramiento de Calidad en Proyectos de Software y Web. Luján: Tesis de Licenciatura en la Facultad de Ciencias Exactas de la UNLu.
- Molina, H., & Olsina, L. (2008). Assessing Web Applications Consistently: A Context Information Approach. 8th Int'l Congress on Web Engineering (ICWE08) (pp. 224-230). NY, USA: IEEE Computer Society. ISBN 978-0-7695-3261-5.
- Moody, D., & Hillegersberg, J. (2008). Evaluating the Visual Syntax of UML: An Analysis of the Cognitive Effectiveness of the UML Family of Diagrams. Berlin, Heidelberg: Springer-Verlag.
- Nielsen, J., & Levy, J. (1994). Measuring usability: preference versus performance. Commun ACM 37(4) , 66-75.
- Niknafs, A., & Asadi, M. (2009). Towards a Process Modeling Language for Method Engineering Support. 2009 WRI World Congress on Computer Science and Information Engineering, (pp. 674-681). Los Angeles, CA.
- Nitto, E. D., Lavazza, L., Schiavoni, M., Tracanella, E., & Trombetta, M. (2002). Deriving Executable Process Descriptions from UML. Proceedings of the 24th International Conference on Software Engineering (ICSE '02) (pp. 155-165). Orlando, Florida: ACM.

- Olsina, L. (1998). Functional View of the Hypermedia Process Model. The Fifth International Workshop on Engineering Hypertext Functionality at International Conference on Software Engineering (ICSE'98), (pp. 1-10). Kyoto, Japan.
- Olsina, L. (1997). Applying the Flexible Process Model to build Hypermedia Products. Hypertext and Hypermedia: Tools, Products, Methods (HHTPM 97) (pp. 211-221). Paris, Francia: Hermes Ed.
- Olsina, L., Covella, G., & Dieser, A. (2013). Metrics and Indicators as Key Organizational Assets for ICT Security Assessment. In book titled "Emerging Trends in ICT Security", chapter 2 (pp. 25-44). Elsevier (Morgan Kaufmann), 1st Edition, Akhgar & Arabia (Eds.). ISBN: 9780124114746.
- Olsina, L., Papa, M., & Becker, P. (2011). Assessing Integrated Measurement and Evaluation Strategies: A Case Study. 7^o Central & Eastern European Software Engineering Conference CEE-SEC(R) (pp. 1-10). Moscow, Russia: IEEE Xplore (ISSN 978-1-4673-0844-1/11).
- Olsina, L., Papa, M., & Molina, H. (2008b). How to Measure and Evaluate Web Applications in a Consistent Way. In Rossi, Pastor, Schwabe, & Olsina (Eds.), Web Engineering: Modeling and Implementing Web Applications (pp. 385-420). Springer HCIS.
- Olsina, L., Rossi, G., Garrido, A., Distanto, D., & Canfora, G. (2008a). Web applications refactoring and evaluation: a quality-oriented improvement approach. Journal of Web Engineering, Volume 7 Issue 4. Rinton Press , 258-280.
- Olsina, L., & Martín, M. (2004). Ontology for Software Metrics and Indicators. Journal of Web Engineering. Santiago de Chile. Rinton Press. US. 4 Vol. 2. , 262-281.
- Olsina, L., & Rossi, G. (2002). Measuring Web Application Quality with Web-QEM. MultiMedia, IEEE (Volume:9 , Issue: 4) , 20-29, 9 (4).
- Olsina, L., Papa, M., Souto, M., & Rossi, G. (2001). Providing Automated Support for the Web Quality Evaluation Methodology. 4th Workshop on Web Engineering, at the 10th International WWW Conference, (pp. 1-11). Hong Kong.
- OMG-BPMN. (2011). Business Process Model and Notation (BPMN), v2.0.
- OMG-SPEM. (2008). Software & Systems Process Engineering Meta-Model Specification V2.0.
- OMG-UML. (2010). OMG Unified Modeling Language, Superstructure, v2.3.
- OMG-UML. (2004). Unified Modeling Language Specification, Version 2.0. - Document formal/05-07-04.
- OPF. OPEN Process Framework. Retrieved 05/02/2014, from <http://www.opfro.org/>
- Osterweil, L. (1987). Software Processes are Software Too. Proceedings of the 9th International Conference on Software Engineering (pp. 2-13). Monterey, California, United States: IEEE Computer Society.
- Pall, G. A. (1987). Quality Process Management. Prentice-Hall.

- Papa, M. (2012). Toward the Improvement of a Measurement and Evaluation Strategy from a Comparative Study. Proceedings of 12th International Conference of Web Engineering (ICWE2012). Current Trends in Web Engineering, M. Grossniklauss and M. Wimmer (Eds.) (pp. 189-203). Berlin, Germany: LNCS 7703, Springer Berlin Heidelberg, ISBN: 978-3-642-35622-3.
- Papa, M. (2005). Diseño e Implementación de la Evaluación a partir del Marco Conceptual INCAMI. Luján: Tesis de Licenciatura en la Facultad de Ciencias Exactas de la UNLu.
- Papa, M., Becker, P., & Olsina, L. (2010). Estrategias de Medición y Evaluación: Diseño de un Estudio Comparativo. En ASSE'10, Simposio Argentino de Ingeniería de Software (39 JAIIO) (pp. 323-337). Buenos Aires, Argentina: ISSN: 1850-2792.
- Paulik, M., Weber, C., Curtis, B., & Chrissis, M. (1995). Capability Maturity Model: Guidelines for Improving the Software Process. Carnegie-Mellon University, SEI.
- Pérez-Jiménez, J. D., Durán, A., & Bernárdez, B. (2009). Fundamentos para un Entorno de Application Lifecycle Management Dirigido por Procesos. Actas de los Talleres de las Jornadas de Ingeniería del Software y Bases de Datos (SISTEDES), Vol. 3, No. 3 (pp. 41-48). San Sebastián: ISSN 1988-3455.
- PMI. (2009). Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK) Cuarta Edición. Pennsylvania, USA: Project Management Institute, Inc.
- Portela, C., Vasconcelos, A., Silva, A., Silva, E., Gomes, M., Ronny, M., et al. (2012b). xSPIDER_ML: Proposal of a Software Processes Enactment Language Compliant with SPEM 2.0. Journal of Software Engineering and Applications, Vol. 5 No. 6 , 375-384.
- Portela, C., Vasconcelos, A., Silva, A., Sinimbu, A., Silva, E., Ronny, M., et al. (2012a). A comparative analysis between BPMN and SPEM modeling standards in the software processes context. Journal of Software Engineering and Applications (JSEA). Vol. 5 No. 5 , pp. 330-339.
- PSM. (2000). Practical Software and Systems Measurement: A Foundation for Objective Project Management (v 4.0). Department of Defense and US Army.
- Radice, R., Roth, N., O'Hara Jr, A., & Ciarfella, W. (1985). A Programming Process Architecture. IBM Systems Journal, Volume 24 Issue 2 , 79-90.
- Riesco, D., Montejano, G., Debnath, N., & Perez Cota, M. (2009). Formalizing the Management Automation with Workflow of Software Development Process Based on the SPEM Activities View. Sixth International Conference on Information Technology: New Generations (ITNG 2009) (pp. 131-136). Las Vegas, Nevada : IEEE Computer Society.
- Rivera, M. B., Molina, H., & Olsina, L. (2007). Sistema Colaborativo de Revisión para el soporte de información de contexto en el marco C-INCAMI. XIII Congreso Argentino de Ciencias de la Computación (CACIC 2007), Workshop WISBD (pp. 518-529). Corrientes, Argentina: ISBN 978-950-656-109-3.

- Rodríguez, M., Verdugo, J., Coloma, R., Genero, M., & Piattini, M. (2010b). Metodología para la evaluación de la calidad en los modelos UML. *Revista Española de Innovación, Calidad e Ingeniería del Software (REICIS)*, 4-35.
- Rodríguez, M., Genero, M., Torre, D., Blasco, B., & Piattini, M. (2010a). A Methodology for Continuous Quality Assessment of Software Artefacts. In *Proceedings of 10th International Conference on Quality Software (QSIC)*, 254-261.
- Rolland, C. (1998). A Comprehensive View of Process Engineering. *Proceedings of the 10th International Conference on Advanced Information Systems Engineering (CAiSE'98)*, (pp. 1-24).
- Russel, N., van der Aalst, W., Hofstede, A., & Wohed, P. (2006). On the suitability of uml activity diagrams for business process modelling. In *Proceedings of the Third Asia-Pacific Conference on Conceptual Modelling (APCCM)*. Vol. 53, (pp. 195-204). Hobart.
- Software Engineering Laboratory. (1995). *Software Measurement Guidebook (Revision 1)*. Greenbelt, Maryland: Goddard Space Flight Center.
- Solingen, R. v., & Berghout, E. (1999). *The Goal/Question/Metric Method: a practical guide for quality improvement of software development*. Maidenhead, England: The McGraw-Hill Company.
- Solingen, R., & Berghout, E. (1997). Improvement by goal-oriented measurement. In *Proc. of the European Software Engineering Process Group conference (E6-SEPG)*. Amsterdam, The Netherlands .
- StarUML. StarUML Project. Retrieved 05/02/2014, from <http://staruml.sourceforge.net/en/>
- van der Aalst, W., ter Hofstede, A., Kiepuszewski, B., & Barros, A. (2003). *Workflow Patterns. Distributed and Parallel Databases. 1 : Vol. 14* (pp. 5-51). Netherlands Springer (ISSN 0926-8782).
- Verlage, M. (1994). *Multi-View Modeling of Software Processes*. Proc. 3rd European Workshop Software Process Technology .
- Wei, W., Hongwei, D., Jin, D., & Changrui, R. (2006). A Comparison of Business Process Modeling Methods. *IEEE International Conference on Service Operations and Logistics, and Informatics, 2006. SOLI '06*, pp. 1136-1141 .
- White, S. A. (2004). *Process Modeling Notations and Workflow Patterns*. IBM Corporation. - BPTrends.
- Zamli, K. Z. (2001). *Process Modeling Languages: A Literature Review*. *Malaysian Journal of Computer Science*. Vol. 14, No 2, 26-37.

Apéndice A: Lenguajes de Modelado de Proceso

Si bien la idea de este trabajo no es hacer una descripción de todos los lenguajes de modelado de proceso, se cree que es importante comentar algunas de las propuestas existentes.

Los lenguajes brevemente descriptos a continuación han sido escogidos por tener un gran uso dentro de la industria, la academia, o por ser los que están apoyados por organismos que tienen un gran peso dentro del ámbito de la Ingeniería del Software o del modelado y definición de procesos.

A.1. IDEF0 (Integration DEFINition 0)

Es una técnica de modelado funcional creada para representar de manera estructurada y jerárquica las actividades que conforman un sistema o empresa y los objetos o datos que soportan la interacción de esas actividades. Fue desarrollada en la Fuerza Aérea de los Estados Unidos en 1970 y forma parte de un conjunto de herramientas de modelado denominada IDEF (*Integration DEFINition*), las cuales cubren un rango de usos como modelado de actividades (IDEF0), de información (IDEF1), de Datos (IDEF1x), orientado a objetos (IDEF4), entre otros. Estos lenguajes de definición se han convertido en técnicas de modelado estándares. De hecho, ISO utiliza IDEF0 para modelar el proceso de evaluación de software [ISO/IEC, 2001b].

Un modelo IDEF0 está formado por un conjunto de diagramas jerárquicos con diferentes niveles de detalle que permiten describir las funciones especificadas en el nivel superior. Es decir, en las vistas superiores se visualizan los procesos o actividades fundamentales que ejecuta la organización. Los elementos gráficos utilizados para la construcción de los diagramas IDEF0 son cajas o cuadros y flechas. La semántica de estos elementos gráficos es la siguiente (en la Figura A-1 se muestra un ejemplo):

Actividad: se representa con un cuadro, indica una función, proceso o transformación.

Entrada: se representa con una flecha entrando por el lado izquierdo de la actividad, indica los materiales o informaciones que se transformarán en la actividad para obtener la salida.

Salida: se representa con una flecha saliendo del lado derecho de la actividad, indica los objetos o informaciones producidos por la ocurrencia de la actividad.

Control: se representa con una flecha entrando por la parte superior, indica las regulaciones que determinan si una actividad se realiza o no. Por ejemplo, normas, guías, reglas, políticas, etc.

Sujeto: se representa con una flecha entrando por la parte inferior, indica los recursos que ejecutan una actividad. Por ejemplo, personas, maquinarias, etc.

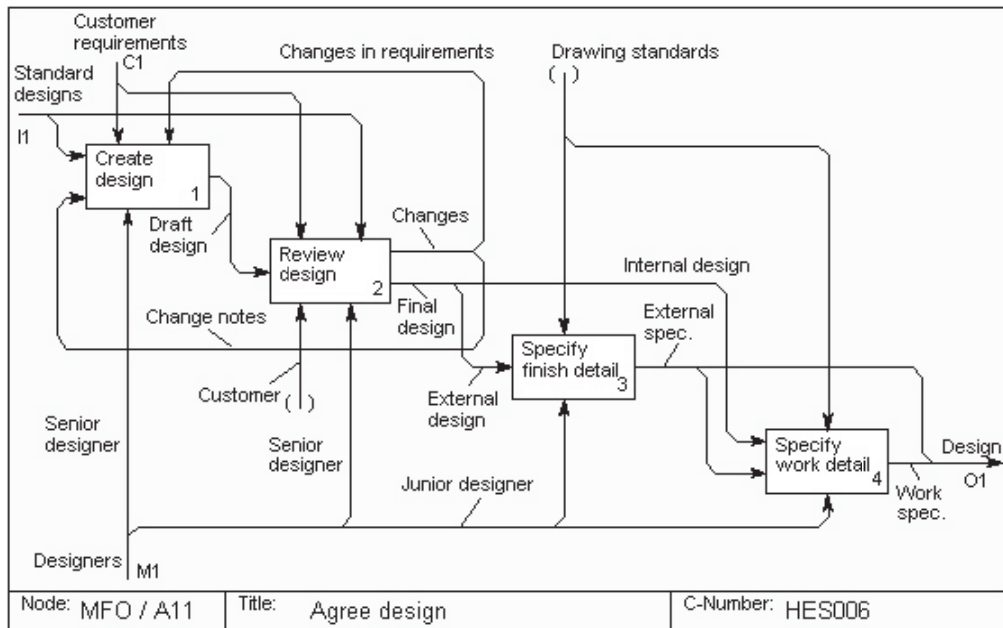


Figura A-1: Ejemplo de un modelo de proceso utilizando IDEF0.

A.2. Diagrama de Actividad de UML

La aparición de UML (*Unified Modeling Language*) fue un cambio muy importante, de hecho un gran avance, en lo que respecta al modelado orientado a objetos (aunque es un lenguaje de modelado de propósito general), por lo cual llegó a convertirse en un estándar. Para el modelado de proceso se intentó utilizar los diagramas de actividad de UML, pero se pudo observar que muchos de los aspectos de un proceso no podían ser representados. Con el tiempo apareció la versión 2.0 de UML [OMG-UML, 2004], donde se especifican nuevos constructores, lo cual lo hizo mucho más flexible y robusto, permitiendo un modelado más amplio de los procesos. Algunos de los elementos que forman parte de un Diagrama de Actividad de UML son:

Activity (Actividad): una actividad es la especificación de una o varias acciones.

Action (Acción): una acción representa un solo paso dentro de una actividad.

Control Flow (Flujo de Control): indica el flujo de control de una acción u actividad a otra.

Initial Node (Nodo Inicial): Un nodo inicial o de comienzo indica el punto de inicio del proceso o actividad.

Final Node (Nodo Final): existen dos tipos de nodos finales, los nodos finales de actividad (Activity Final) y los de flujo (Flow Final). La diferencia entre los dos tipos de nodos es que el nodo final del flujo denota el final de un solo flujo de control, y el nodo final de actividad denota el final de todos los flujos finales dentro de la actividad.

Object Flow (Flujo de Objeto): un flujo de objeto es la ruta a lo largo de la cual pueden pasar objetos o datos.

Object (Objeto): representan los productos de trabajo involucrados en el proceso

(documentos, productos software, datos, etc.). Se les puede agregar el estereotipo «datastore», y utilizarlos como un almacenador de datos.

Decision Node (Nodo de Decisión) y Merge Node (Nodo de Combinación): el primero se utiliza para indicar que se debe tomar una decisión (es decir, existen varias alternativas a seguir, pero sólo una de ellas se puede realizar), mientras que el segundo permite unir los diferentes caminos luego de haber tomado alguna decisión.

Fork Node (Nodo de Bifurcación) y Join Node (Nodo de Unión): muestran el inicio y fin de actividades que se realizarán en paralelo.

Lane (Partición): permite mostrar qué actividades son realizadas por una organización, departamento o individuo.

En la Figura A-2 se pueden observar algunos de los símbolos utilizados para representar los elementos antes mencionados.

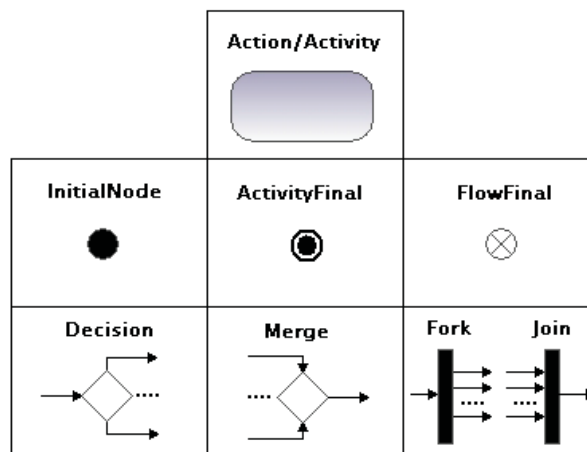


Figura A-2: Algunos símbolos de los Diagramas de Actividad de UML.

A.3. BPMN (Business Process Modeling Notation)

Fue desarrollado por la iniciativa *Business Process Management* (BPMP), y fue adoptada por la *OMG* (*Object Management Group*). BPMN es un estándar para el modelado de procesos de negocios y provee una notación gráfica para especificar procesos de negocio en un *Diagrama de Proceso de Negocio*.

BPMN no solo está pensado para ser un lenguaje utilizado por expertos en procesos de negocios, sino también con el fin de que sea entendido de una manera intuitiva por cualquier usuario [OMG-BPMN, 2011].

Sus elementos se dividen en cuatro categorías:

- ✓ Flow Objects (Objetos de Flujo): definen el comportamiento del proceso.
 - ❖ Events (Eventos)
 - ❖ Activities (Actividades)
 - ❖ Gateways
- ✓ Connecting Objects (Objetos de Conexión): conectan los diferentes objetos de flujo y artefactos.
 - ❖ Sequence Flow (Flujo de Secuencia)

- ❖ Message Flow (Flujo de Mensaje)
- ❖ Association (Asociación)
- ✓ Swimlanes: son un mecanismo para organizar las actividades en categorías visuales separadas para ilustrar distintas capacidades funcionales o responsabilidades.
 - ❖ Pool
 - ❖ Lane
- ✓ Artifacts (Artefactos): permiten agregar más información al modelo.
 - ❖ Data Objects (Objetos de Datos)
 - ❖ Group (Grupo)
 - ❖ Annotation (Anotación)

A continuación se describen cada uno de los elementos de las categorías mencionadas (en la Figura A-3 se pueden observar los símbolos utilizados para algunos de estos elementos):

Eventos: indican la posible ocurrencia de algún suceso que afecte el flujo normal del proceso.

Actividades: indican el tipo de trabajo que puede ser realizado. Puede ser atómica o no atómica (compuesta).

Gateway: muestra que se debe tomar alguna decisión. También pueden llegar a indicar bifurcaciones, uniones y combinaciones de caminos.

Flujo de Secuencia: muestra el orden (secuencia) en el cual se ejecutan las actividades.

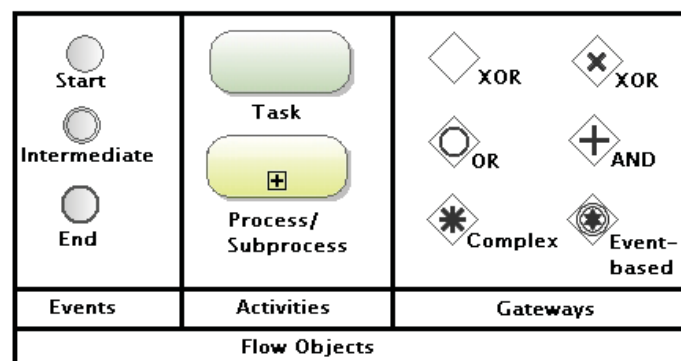


Figura A-3: Algunos símbolos de BPMN.

Flujo de Mensajes: muestra el flujo de mensajes entre dos procesos participantes (entidades del negocio o roles del negocio).

Asociación: se usa para asociar Artefactos, datos o texto a Objetos de Flujo. Permiten indicar cuáles son las entradas y salidas de las actividades.

Swimlanes (Pool – Lane): permiten organizar las actividades de tal forma que sea fácil observar quiénes son los encargados de realizarlas, o en qué lugar.

Objetos de datos: muestran cuáles son los datos requeridos o producidos por una actividad

Grupo: Se utiliza para documentación o propósitos de análisis, pero no afecta el

flujo de las actividades.

Anotación: permite agregar información textual adicional al diagrama.

A.4. SPEM 2.0 (Software & Systems Process Engineering Meta-Model 2.0)

Es un metamodelo definido por la OMG (*Object Management Group*) para la descripción de procesos de desarrollo de software. SPEM también ofrece un marco de trabajo proveyendo los conceptos necesarios para el modelado, documentación, presentación, gestión e intercambio de los procesos de desarrollo software y sus componentes, dando una sintaxis y una estructura para cada uno de los diferentes elementos del proceso de desarrollo, incluyendo: roles, actividades, tareas, y productos de trabajo, entre otros. La versión 2.0 de SPEM [OMG-SPEM, 2008] utiliza UML 2.0 como notación para el modelado y define un perfil UML (es decir, un conjunto de estereotipos UML que permiten representar métodos y procesos usando UML), permitiendo que las herramientas actuales de UML puedan extenderse fácilmente para soportar modelos SPEM.

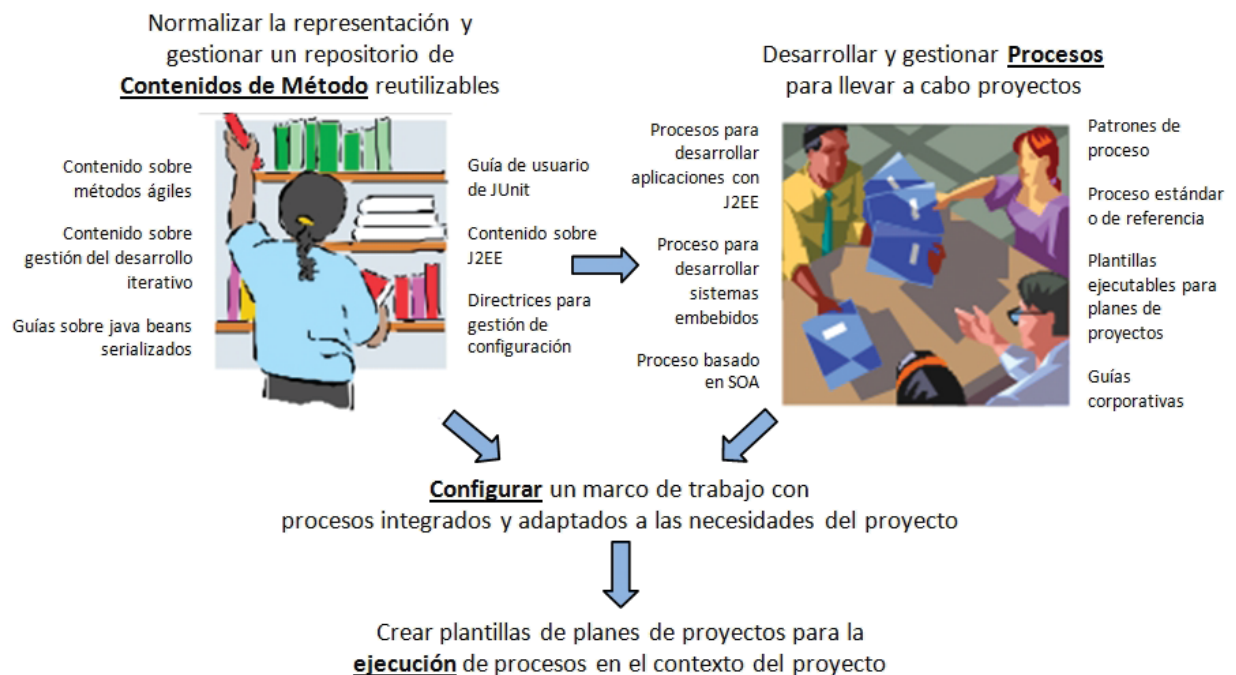


Figura A-4: Marco de trabajo de SPEM 2.0 (adaptado de [OMG-SPEM, 2008])

El marco de trabajo de SPEM (mostrado en la Figura A-4), contribuye en los siguientes escenarios:

✓ **Normalizar la representación y gestionar un repositorio de contenidos de métodos reutilizables:** Los desarrolladores de software necesitan conocer y familiarizarse con los métodos y prácticas claves para el desarrollo de software. SPEM 2.0 contribuye a dar soporte a los desarrolladores en estos aspectos, facilitando el establecimiento de una base de conocimiento, utilizando un formato estandarizado, que permita gestionar contenido sobre el desarrollo de software y sistemas. Dicho contenido incluye, por ejemplo, definiciones de método, documentos técnicos, guías, plantillas, procedimientos internos, material didáctico y cualquier otro material útil sobre cómo desarrollar software. Esta base de

conocimiento puede ser usada como referencia y/o entrenamiento y constituye la base para los procesos de desarrollo.

✓ **Soportar el desarrollo sistemático, gestión y crecimiento de los procesos de desarrollo:** Para llevar a cabo un proyecto, los desarrolladores deben contar con algún proceso de desarrollo. Los modelos de proceso deben permitir identificar cuáles métodos son aplicables en cada una de las tareas. Además el equipo de desarrollo debe conocer cómo las tareas se relacionan unas con otras. SPEM 2.0 soporta la creación sistemática de procesos basada en elementos reutilizables de contenido de método. Los procesos pueden ser representados mediante flujos de actividades y/o estructuras de desglose de trabajo (WBS, Work Breakdown Structure). SPEM 2.0 también provee un fundamento conceptual para seleccionar, adaptar y rápidamente ensamblar procesos para proyectos concretos de desarrollo.

✓ **Configurar un marco de trabajo con procesos adecuados a las necesidades del proyecto:** Ningún proyecto de desarrollo es exactamente igual a otro y el mismo proceso de desarrollo nunca se ejecuta dos veces. Las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigen que el proceso sea configurable o adaptable. Según CMMI, un proceso de nivel de capacidad 3 se caracteriza como un "proceso definido", es decir, un proceso gestionado (nivel de capacidad 2) que se adapta o configura a partir de un conjunto de procesos estándares de la organización, de acuerdo a las guías de adaptación de la organización. SPEM 2.0 permite contar con procesos definidos, al proveer la capacidad para reutilizar procesos o patrones de procesos, para modelar variabilidad (es decir, procesos compuestos de partes que se pueden intercambiar con el propósito de mejorar o corregir el proceso u obtener un producto diferente) y para adaptar los procesos existentes permitiendo a los usuarios definir su propias extensiones, omisiones y puntos de variabilidad sobre los procesos estándares reutilizados. De esta manera SPEM favorece a que las organizaciones puedan disponer de bibliotecas de procesos y contenidos de métodos que sean reutilizables utilizando las capacidades descritas en los dos puntos anteriores. A partir de estas bibliotecas, los jefes de equipo puedan seleccionar y adaptar los procesos y contenidos de métodos que requieran para las necesidades de sus proyectos.

✓ **Soportar la ejecución (enactment) de un proceso para proyectos de desarrollo:** Contar con la definición de procesos sólo ofrece valor si ésta afecta y dirige el comportamiento de los equipos de desarrollo. Los procesos, así como las guías de contenidos de método, deben estar disponibles en el contexto de la labor diaria de los directores de proyectos, líderes técnicos y desarrolladores. Por lo tanto, deben ser desplegados en formatos listos para su utilización con los distintos sistemas de ejecución (enactment) de proceso utilizados por el equipo de desarrollo. Sistemas de ejecución típicos son los sistemas de planificación de proyecto y de recursos, los sistemas de seguimiento de trabajo, y los motores de *workflow*. SPEM 2.0 proporciona estructuras de definición de procesos que permiten a los ingenieros de proceso expresar cómo debería ejecutarse un proceso dentro de estos sistemas. Por ejemplo, SPEM 2.0 permite modelar iteraciones para indicar que ciertas actividades deben ejecutarse repetidas veces en un proyecto, o que podría haber varias actividades que se pueden realizar en paralelo.

El metamodelo SPEM 2.0 está estructurado en 7 paquetes, a saber: *Core*,

Process Structure, Process Behavior Managed Content, Method Content, Process with Method, y Method Plug-In. Estos se describen brevemente a continuación, y se comentan algunos de sus elementos más importantes. En la Figura A-5 se puede apreciar cómo se relacionan muchos de los conceptos de estos paquetes.

Paquete **Core**: contiene las clases y abstracciones utilizadas como base para el resto de los paquetes del metamodelo. En lo que sigue se comentan algunos de los conceptos de este paquete:

- ✓ Work Definition (Definición de trabajo): es una abstracción que representa el comportamiento para realizar un trabajo. Puede contener un conjunto de pre y post condiciones, es decir, restricciones que deben cumplirse antes de realizar el trabajo o después que se haya realizado para poder decir que este ha finalizado.
- ✓ Work Definition Performer (Ejecutor de una definición de trabajo): es una abstracción que representa a quien realiza/ejecuta un trabajo.

Paquete **Method Content**: el objetivo principal de este paquete es el de definir tareas (Task Definition), organizarlas en distintos pasos (Steps), definir cuáles son los productos (Work Product Definition) de entrada/salida de cada una de ellas y especificar quién ha sido el que ha realizado dicha tarea (Rol Definition). A continuación se describen estos elementos clave (ver en Figura A-6 los íconos utilizados en SPEM 2.0 para modelar estos conceptos):

- ✓ Task Definition (Definición de tarea): describe una unidad asignable de trabajo que se asocia a una definición de rol específica. La granularidad de una Task Definition es generalmente de unas pocas horas a unos pocos días. Usualmente afecta a uno o un pequeño número de productos de trabajo.
- ✓ Rol Definition (Definición de rol): define un conjunto de habilidades, competencias y responsabilidades de un participante o de un conjunto de participantes. Los roles son usados en las Task Definition para indicar quiénes son los que las llevan a cabo, así como en los Work Product Definition para indicar quienes son los responsables de estos. Se debe tener en cuenta que un rol no es una persona en particular. Los miembros de una organización pueden jugar diferentes roles.
- ✓ Work Product Definition (Definición de producto de trabajo): define cualquier producto de trabajo tangible que sea consumido, producido o modificado en las tareas.
- ✓ Step (Paso): sirve para organizar una tarea en partes o subunidades de trabajo. El conjunto de Steps definidos para una Task Definition representa todo el trabajo que debería realizarse para alcanzar el objetivo de una Task Definition.
- ✓ Tool Definition (Definición de herramienta): describe las capacidades de una herramienta CASE (*Computer Aided Software Engineering*), una herramienta de propósito general, o cualquier otra unidad de automatización.

Paquete **Process Structure**: contiene los elementos estructurales básicos para definir modelos de procesos. Soporta la creación de diferentes tipos de procesos, tales como procesos en cascada, iterativos o incrementales. Cuenta con estructuras de desglose para modelar diferentes elementos del proceso, así como de diferentes relaciones estructurales y atributos descriptivos. Algunos de los elementos más relevantes de este paquete son los siguientes (en Figura A-6 se

pueden observar los íconos utilizados en SPEM 2.0 para modelar estos conceptos):

✓ Activity (Actividad): es una Work Definition concreta que representa una unidad general de trabajo dentro de un proceso (también puede representar un proceso en sí). Una Activity es asignable a un ejecutor específico representado por un Role Use. Una Activity insume entradas y produce salidas representadas por Work Product Use.

✓ Role Use (Rol en uso): representa a un ejecutor o participante de una Activity. Un Role Use simboliza a una persona real realizando una actividad específica y teniendo responsabilidades específicas de la actividad que lleva a cabo. Notar que no es una definición genérica reutilizable de un rol de la organización, sino un elemento específico.

✓ Work Product Use (Producto de trabajo en uso): representa un tipo de entrada/salida de una Activity. Una instancia de Work Product Use es un objeto específico de una actividad y no una definición genérica reutilizable de un producto de trabajo.

✓ Work Product Use Relationship (Relación de producto de trabajo en uso): expresa relaciones entre productos de trabajo, por ejemplo, puede representar relaciones de composición, agregación, dependencia, etc.

✓ Milestone (Hito): representa un evento importante para un proyecto de desarrollo.

✓ Process Element (Elemento de Proceso): es cualquier elemento que sea parte de un proceso, por ejemplo: una actividad, un rol, un producto de trabajo, etc.

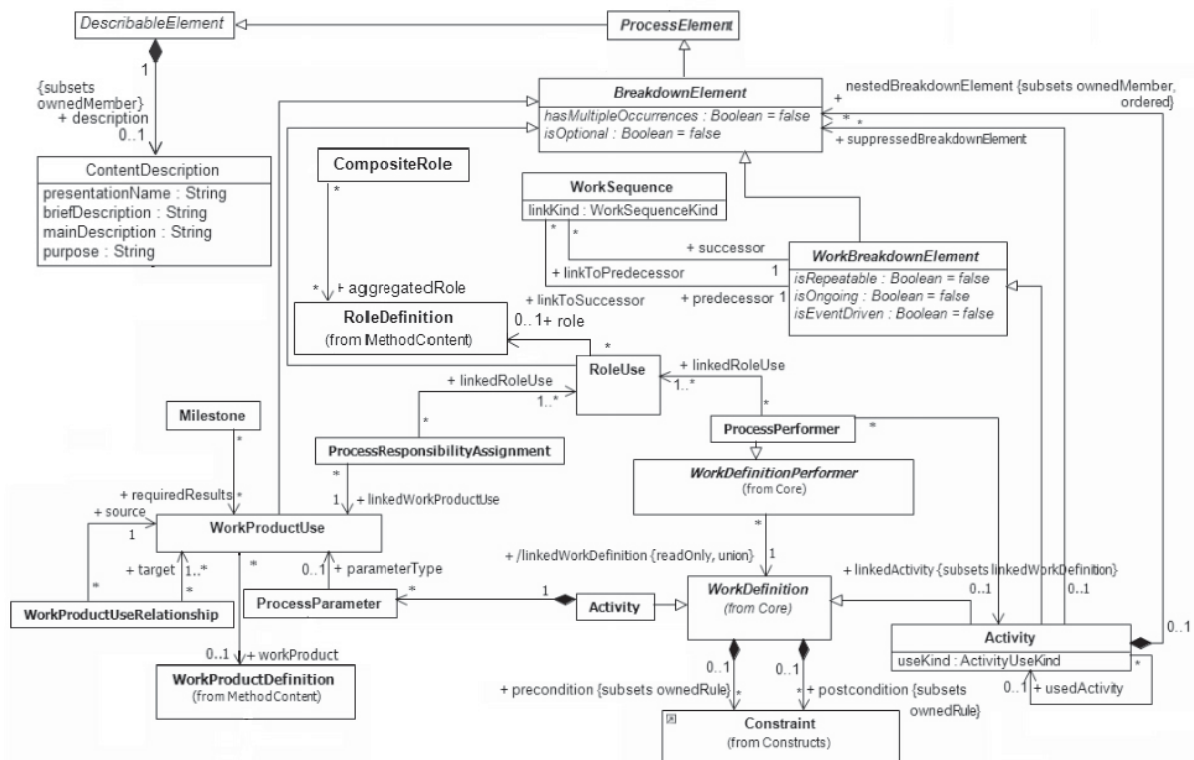


Figura A-5: Extracto del Modelo conceptual de SPEM (adaptado de [OMG-SPEM, 2008]).

Paquete **Process Behaviour**: permite modelar el comportamiento de los procesos, es decir, la parte dinámica de los mismos. SPEM no incluye un

mecanismo propio para representar el comportamiento, sino que opta por reutilizar los ya existentes. Por ejemplo, se pueden utilizar los diagramas de actividad de UML 2.0, BPMN, u otro, para representar el comportamiento del proceso, máquinas de estados para modelar el ciclo de vida de un producto de trabajo, etc.

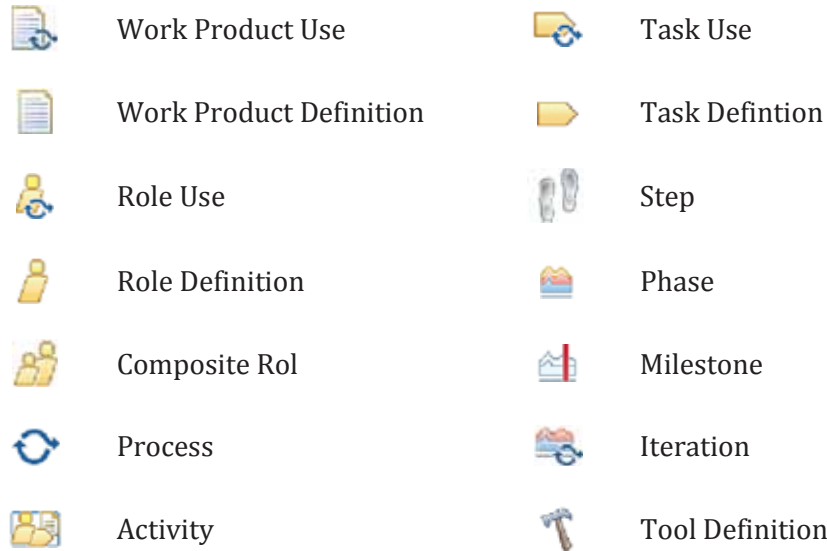


Figura A-6: Iconos de los principales elementos de modelado de SPEM.

Paquete **Managed Content:** permite agregar a los elementos del proceso anotaciones y descripciones que no pueden ser expresadas como modelos y que, por lo tanto, deben ser documentadas y gestionadas como descripciones textuales. Entre los diferentes conceptos definidos en este paquete se pueden mencionar:

- ✓ Describable Element (Elemento describable): representa una generalización abstracta para todos los elementos de SPEM que pueden ser documentados con descripciones textuales. Ejemplos de elementos descriptibles son: actividad, rol, productos de trabajo, etc.
- ✓ Content Description (Descripción de contenido): se utiliza para almacenar una descripción textual de un elemento descriptible.
- ✓ Metric (Métrica): define una medición estándar para instancias de elementos descriptibles. Por ejemplo, un ingeniero de proceso puede definir métricas para una actividad (esfuerzo estimado en horas), métricas para un producto de trabajo (errores por líneas de código), o métricas para un rol (costo por hora).

Paquete **Process With Methods:** contiene los elementos necesarios para integrar el paquete Process Structure con los conceptos y elementos del paquete Content Method.

- ✓ Task Use (Tarea en Uso): representa una tarea en el contexto de una actividad específica.

Paquete **Method Plug-In:** define los mecanismos de extensibilidad y variabilidad para contenidos de métodos y de procesos. De esta manera SPEM provee mayor flexibilidad en la definición de diferentes variantes de contenido de método y de procesos, y permite que fragmentos de contenido y de proceso puedan ser reutilizados bajo demanda y así crear, adaptar o especializar contenido

cuando éste sea requerido y pueda ser mantenido como unidades separadas.

Gracias a los diferentes paquetes mencionados, un ingeniero de procesos dispone de diferentes capacidades, conjuntos de conceptos y niveles de formalismo para documentar los procesos.