



DEPARTAMENTO DE LENGUAJES Y
SISTEMAS INFORMÁTICOS E
INGENIERÍA DEL SOFTWARE

FACULTAD DE INFORMÁTICA

UNIVERSIDAD POLITÉCNICA DE MADRID

TESIS DOCTORAL

Un Proceso para la evaluación del Entrenamiento
Organizacional

AUTOR: ~~Santiago Matalonga Motta~~

DIRECTOR Tomás San Feliu Gilabert

01/07/2011

DEPARTAMENTO DE LENGUAJES Y SISTEMAS DE
INFORMÁTICOS E INGENERÍA DE SOFTWARE
FACULTAD DE INFORMÁTICA

Un Proceso para la evaluación del Entrenamiento
Organizacional

Tesis Doctoral

Santiago Matalonga Motta

01/07/2011

*"For every mistake,
you must surely be learning"*

While my guitar gently weeps.

George Harrison

Agradecimientos

Completar un volumen de estas características no es una tarea de una sola persona. Las experiencias resumidas en este trabajo de tesis han involucrado el trabajo de muchas personas que han permanecido anónimas en este volumen. A alguna de ellas he tenido la oportunidad de agradecerles en tiempo y forma, para estas y para los demás *gracias* por el apoyo y la confianza.

Quienes trabajamos en procesos sabemos que *las formas hay que respetarlas*, pero mas allá de toda formalidad, tengo la convicción que este trabajo no habría sido posible sin el apoyo, dedicación y compromiso de Tomás. Gracias por darme la libertad de explorar en mis ideas, por el consejo constante y por las intervenciones clave que resultaron en este trabajo.

Hace un par de años escuché al rector de la Universidad ORT decir que cuando un estudiante realiza el grado, quienes sufren son la familia, cuando realiza el postgrado quien sufre es la novia, y cuando se realiza el doctorado quien sufre es la esposa (en algunos casos la misma persona) y los hijos. Aunque no sigo ese patrón, me siento identificado con la imagen. A Gabriela, **no hay palabras que alcancen**. A mi familia (Mario, Beatriz y Coro), lo hago con palabras que no son mías: *“No hay presente sin historia, el futuro lo labra cada uno eligiendo de ambas lo más sabio y prudente”* Yaya 24/08/1992

Finalmente, agradecer el apoyo y la disponibilidad de Silvia Nane. No sé si algún día leerás esto, de todas maneras ya tendremos la oportunidad de vernos y agradecerte en la medida que te lo mereces. Quienes sí las han leído fueron “el tío Hugo” y “la tía Blanqui”, nuestros proof-readers, gracias a ellos, el contenido de esta tesis no se pierde en el continente.

A mis docentes y compañeros en la Universidad ORT, a Patricia por invitarme a participar del doctorado conjunto. Inés, quien me abrió las puertas a la docencia. Gastón, Pedro y todos.

A Gonzalo y su equipo en el Laboratorio de Ingeniería del Software de la Facultad de Informática de la UPM.

A Infocorp Software Factory, especialmente a Ana Inés y a Pablo, por permitirnos experimentar en su fábrica.

Santiago Matalonga

Montevideo, Julio 2011

Reconocimientos

Este trabajo fue financiado por la Universidad ORT Uruguay y la Universidad Politécnica de Madrid, dentro del marco del programa de doctorado conjunto.

Las estancias y transados a Madrid fueron financiadas por:

- Programa de Desarrollo Tecnológico Número: S/C/BE/55/17 de Diciembre 2006 a Marzo 2008
- Agencia Nacional de Investigación e Innovación BE-MOV-2009-108 en Julio de 2009
- Banco Santander Hispano-Universidad Politécnica de Madrid. De Setiembre a Diciembre de 2010.

Las asistencias a Congresos fueron parcialmente financiadas por el fondo concursable de la Universidad ORT Uruguay y fondos de la cátedra de mejora de procesos en el espacio latinoamericano de la Universidad Politécnica de Madrid.

Resumen

La industria del software es un entorno en donde los ciclos de las tecnologías son tan cortos que el entrenamiento y actualización continua de los recursos de una organización son necesarios para mantener la competitividad.

El Entrenamiento Organizacional, es una disciplina que se encuentra presente en los modelos de mejora de procesos de software. Esto indica que las organizaciones confían en el entrenamiento para:

Mantener actualizados y competitivos a sus recursos

Mantener a los recursos entrenados en sus sistemas de producción.

Como resultado, los departamentos de entrenamiento en las organizaciones de software debieran ser capaces de comunicar a la Alta gerencia su contribución a los objetivos de negocio de la organización.

La evaluación del entrenamiento es todavía un área abierta de investigación. A pesar de que el modelo dominante de evaluación del entrenamiento (el modelo de 4 niveles de Kirkpatrick) fue publicado hace más de 50 años. Todavía las organizaciones no cuentan con mecanismos reproducibles para conseguir medidas en todos los niveles del modelo.

Este trabajo presenta un proceso que permitirá a los departamentos de entrenamiento en las organizaciones de software reportar su contribución al negocio en términos de Retorno de Inversión.

El objetivo del proceso es proveer a las organizaciones de software **un mecanismo que permita a los departamentos de entrenamiento** de las organizaciones de software **justificar la inversión realizada en términos de Retorno de Inversión (ROI).**

El proceso provee una manera de identificar los contenidos de las intervenciones de entrenamiento basándose en el **análisis causal** de defectos. Los defectos de los proyectos serán interpretados como las necesidades de entrenamiento de los recursos. Permitiendo así diseñar intervenciones de entrenamiento dirigidas a estos tipos de defectos.

Este trabajo presenta la validación de este proceso mediante la experiencia en una fábrica de software. La experiencia en la fábrica fue complementada con experiencias en el laboratorio para profundizar algunos de los resultados obtenidos.

Abstract

It is accepted the software development is a fast changing industry, where technological cycles are becoming ever smaller and the need for continuous training and education is a necessity for the people pursuing jobs within the industry.

Organizational Training lies at the core of software process improvement initiatives. A software organization relies on training to:

- Update the knowledge and Skills of the workforce
- Train the workforce in new processes

As a result, like any other unit within the organization, the training department should be able to justify its existence by showing how it contributes to the bottom line results of the organization.

Training evaluation is still an open field of research in other industries. The literature on Human Resource Development is still active with research on the application of training to the workforce. The Dominant model for training evaluation is Kirkpatrick's four level of training. But in over 50 years of history of the model, there is still no reliable way of applying the model to produce results at each of the levels.

This work introduces a process for achieving Return of Investment measures for training interventions within a software Factory.

The goal is to provide **a repeatable process that will enable Training departments** within a software factory **to communicate the result of training interventions in terms of Return of Investment.**

The process provides a way of selecting training intervention by **applying causal analysis** to the defects reported by the ongoing development projects. By interpreting defects, as training needs it is possible to design training aimed at the direct removal of those defects.

This work presents the validation of the process by means of a case study in a Software Factory, and by laboratory experiences designed to complement the case study.

Índice

Agradecimientos	iv
Reconocimientos	v
Resumen	vi
Abstract	vii
Índice	viii
1 Introducción.....	1
1.1 Evaluación del entrenamiento organizacional en la industria del software.....	3
1.2 Importancia del Problema	4
1.3 Objetivo de este trabajo.....	4
1.4 Aportaciones.....	5
1.5 Publicaciones asociadas a este trabajo.....	5
2 Estado del Arte	8
2.1 Evaluación del entrenamiento organizacional.....	8
2.1.1 Evolución de los modelos de evaluación del entrenamiento.....	9
2.1.2 El modelo de evaluación de entrenamiento de cuatro niveles	10
2.1.3 El retorno de inversión en entrenamiento.....	13
2.1.4 El modelo de evaluación de Holton	14
2.1.5 Reflexiones con respecto al entrenamiento organizacional	16
2.2 Modelos de mejora de procesos.....	18
2.2.1 Modelos de la familia ISO.....	19
2.2.2 Seis Sigma.....	23
2.2.3 Modelo de Capacidad y Madurez Integrado	25
2.2.4 Reflexiones respecto a los modelos de Mejora de procesos	31
2.3 Mecanismos de Resolución.....	32
2.3.1 Procesos y formatos para escribir procesos.....	32
2.3.2 Retorno de Inversión de iniciativas de mejora de procesos en la Industria del Software	35
2.3.3 Análisis causal y clasificación de defectos	38
2.3.4 Gestión cuantitativa de procesos	41
2.3.5 Pensamiento Sistémico.....	43

2.4	Resumen del estado del arte.....	48
3	Planteamiento.....	49
3.1	Hipótesis de trabajo.....	51
4	Limitaciones	53
4.1	Capacidad de los procesos.....	53
4.2	Estabilidad de los procesos	54
4.3	Ciclo de vida de los proyectos	55
5	Resolución.....	57
5.1	Definición del proceso para la evaluación del entrenamiento organizacional	57
5.1.1	Validar Criterios de Entrada.....	60
5.1.2	Ejecutar el Proyecto	60
5.1.3	Analizar Defectos	60
5.1.4	Planificar Intervenciones.....	61
5.1.5	Ejecutar Entrenamiento.....	61
5.1.6	Evaluar el Retorno de Inversión del Entrenamiento	61
5.1.7	Comunicar Resultados	61
5.1.8	Evaluar continuidad del Proyecto.....	62
5.1.9	Validar criterios de salida.....	62
5.2	Cálculo del ROI en el proceso	62
5.2.1	Variables de Costos	63
5.2.2	Taxonomía de clasificación.....	64
5.2.3	Variables de beneficios.....	65
5.2.4	Adaptación de la fórmula de ROI.....	65
5.2.5	Interpretación de los posibles resultados	67
5.3	Plan de Implantación del Proceso Propuesto	68
5.4	Resumen de la Resolución	70
6	Experimentación	71
6.1	Breve descripción de la fábrica de software	73
6.1.1	Algunas características distintivas de esta organización.....	75
6.2	Adaptaciones realizadas al proceso para el despliegue en la fábrica	76
6.2.1	Diseño y validación de la taxonomía de Clasificación.....	77
6.2.2	Modificación del sistema de Gestión de defectos	79
6.2.3	Estudio del despliegue de la clasificación.....	80
6.3	Análisis de resultados del piloto	87

6.4	Despliegue del proceso propuesto	90
6.4.1	Selección de los proyectos	90
6.4.2	Despliegue del proceso de análisis causal	92
6.4.3	Adaptaciones al proceso de extracción de las iniciativas de entrenamiento	93
6.4.4	Entrenamiento realizado y resultados obtenidos	94
6.5	Evaluación del retorno de inversión en los proyectos del año 2008	95
6.5.1	Determinar la muestra histórica	95
6.5.2	Recolectar Datos.....	95
6.5.3	Evaluar disponibilidad de datos	96
6.5.4	Establecer nuevos límites de control.....	97
6.5.5	Calcular ROI.....	97
6.6	Evaluación del impacto de la volatilidad de los procesos en la propuesta.....	104
6.6.1	Construcción del modelo de pensamiento sistémico.....	105
6.6.2	Ejecución del modelo y aprendizaje realizado	110
6.6.3	No Conformidades como predictor de defectos del proyecto.....	113
6.7	Resumen de las experimentaciones.....	116
7	Conclusiones y futuras líneas de investigación	119
7.1	Futuras líneas de Investigación	121
8	Bibliografía.....	124

1 Introducción

Los sistemas de software se están volviendo cada vez más presentes en nuestra vida diaria[1]. El software no sólo se encuentra presente dentro de nuestro ordenador, sino que se está volviendo cada vez más omnipresente, teléfonos móviles, semáforos en la calle, marcapasos[2], todas estas objetos son parte de nuestra vida cotidiana. En unos pocos años hemos pasado de ir al banco mensualmente a pagar nuestras cuentas a realizar esto con solo pulsar el botón del ratón[3].

Ahora demandamos y esperamos más de los sistemas con los que interactuamos, motivo que los ha hecho cada vez más grandes y complejos. Pero, lamentablemente, la calidad de los mismos no ha aumentado de la misma manera. Un estudio sumamente citado al respecto es el Reporte del Caos del Standish Group[4, 5], donde año tras año investigan los principales motivos por los cuales los proyectos de software fallan. La Figura 1 muestra la evolución de resultados de este reporte. Aunque parece haber una tendencia hacia la mejora en los proyectos exitosos, es interesante observar cómo los proyectos fallidos y los comprometidos oscilan. Se puede afirmar entonces que a pesar de la mejora en herramientas y prácticas de los últimos años, todavía no hemos encontrado una “bala de plata” [6] al problema de producir software. Es decir no tenemos la solución que nos permita **construir software sin errores de manera consistente proyecto tras proyecto**.

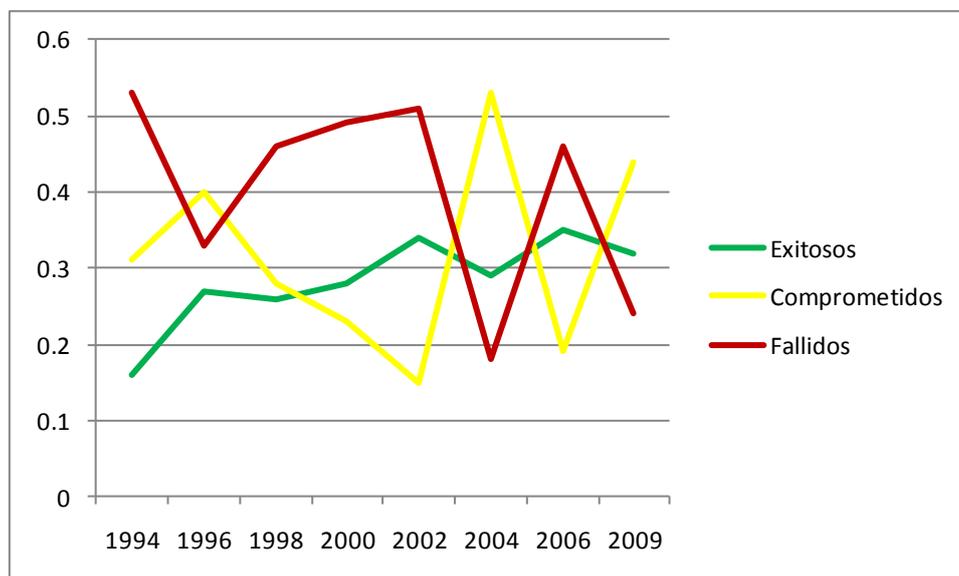


Figura 1 Evolución de resultados de proyectos de software según el Reporte del Caos

Resultante de los trabajo de Demming[7] y Juran[8] se plantea la Tríada *Procesos, Herramientas, y Personas* (ver Figura 2). Diferentes autores han atacado el problema del software por diferentes lados de la triada.

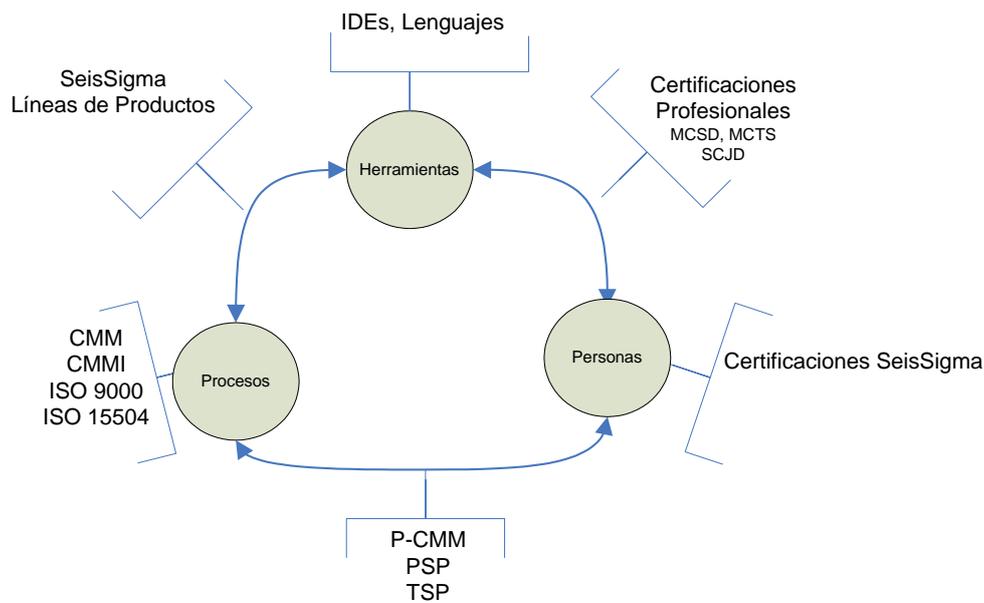


Figura 2 Triada: Procesos, Herramientas, Personas

Por un lado, las iniciativas de mejora de procesos de software [9, 10, 11, 12], intentan mejorar la producción de software a través de la mejora de los procesos utilizados para producirlos. El modelo abanderado de esta corriente proveniente del Software Engineering Institute aboga por los modelos de madurez como respuesta al problema del software. La premisa de los modelos de madurez es que organizaciones más madurez producirán Software con menor índice de defectos [13, 14].

Un modelo de mejora de procesos con más foco en el desarrollo de Recursos Humanos es People-CMM [15], donde la capacitación a los recursos se resalta como importante desde el primer nivel de madurez. Por otro lado, las iniciativas del SEI de mejorar la disciplina de las personas a través de procesos de desarrollo personales [16, 17] y los equipos [18], son propuestas para complementar los modelos de mejora de procesos de software.

Matsubara introdujo tempranamente el concepto de líneas de producción para proyectos de software en Hitachi Software Engineering [9, 19], concepto que luego es explotado por el SEI [20]. El objetivo de las líneas de producto es que la industria del software emule los resultados obtenidos por otras industrias de producción preparando a cada organización para un tipo definido de proyectos. Estas líneas de producto han convertido las “software houses” en “software factories” [21, 22], cuyo foco está en conseguir elementos de tecnologías que puedan reusarse de manera de aumentar la productividad y reducir los defectos.

Las grandes empresas de tecnología se han enfocado en proveer a los programadores de herramientas, de entornos de desarrollo más completos, con el objetivo de reducir las tareas repetitivas del programador y aumentar su productividad [21]. Pero en contraparte, esto ha determinado que el ciclo de vida de las tecnologías se reducido, la velocidad de actualización alta, y por lo tanto alta también la necesidad constante de capacitación de las personas. A modo de ejemplo, en la Figura 4, se muestra el ciclo de vida de las tecnologías Microsoft .NET desde su primera versión [23].

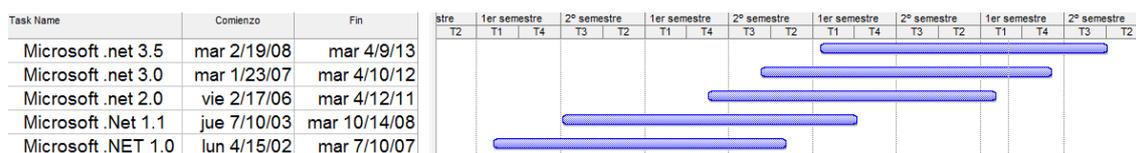


Figura 3 Ciclo de vida de las tecnologías Microsoft .NET

Como modo de contrarrestar esto, en los últimos años han surgido esquemas de certificaciones profesionales para que los desarrolladores demuestren su maestría en una tecnología. Un caso especial es Seis Sigma, en el que para acompañar la capacitación de los recursos se ha desarrollado un esquema de certificaciones profesionales para los practicantes.

Sin embargo, estas soluciones atacan sólo uno de los vértices de la tríada o uno de los lados. Nuestra opinión es que para solucionar el problema del software vamos a tener que encontrar soluciones que integren todos estos conceptos y que ataquen los tres vértices de la tríada.

1.1 Evaluación del entrenamiento organizacional en la industria del software

En la industria del software, los modelos de mejora de proceso ya mencionados [9, 10, 11, 12], han crecido en popularidad y aceptación en los últimos años[24].

Un aspecto que estas tres iniciativas tienen en común es el foco en la capacitación de las personas cómo herramienta para cambiar el comportamiento. En [25, 26] Humphrey destaca como hay que usar el entrenamiento de los desarrolladores de software para modificar su comportamiento. El Personal Software Process[17, 27] también se basa en un fuerte entrenamiento de las personas para modificar su comportamiento y mejorar sus prácticas.

El problema de la evaluación del entrenamiento consiste en obtener medidas confiables para justificar la inversión realizada en los departamentos de entrenamiento. El aspecto clave para las organizaciones está en determinar si las iniciativas de entrenamiento están contribuyendo con sus resultados de negocio. En este punto el modelo de referencia por excelencia es el de 4 niveles de Kirkpatrick[28]. Publicado por primera vez en 1959[29], este modelo se ha mantenido vigente a la hora de evaluar las iniciativas de entrenamiento. El objetivo de los cuatro niveles consiste en intentar demostrar que los dólares invertidos en entrenamiento contribuyen con los resultados de la organización. Lamentablemente, casi medio siglo después, no muchas organizaciones han conseguido establecer medidas consistentes en estos 4 niveles[30].

Los motivos de esta baja efectividad son:

1. Dificultad para establecer y mantener un sistema de mediciones. Implementar un programa de mediciones no es tarea sencilla y el 66% de los programas

implementados en organizaciones de desarrollo de software son descartados luego de 2 años de su puesta en producción[31, 32].

2. Alineación del entrenamiento con los objetivos de negocio. Para ser efectivo, es necesario invertir en intervenciones de entrenamiento que estén alineadas con los objetivos de negocio de la organización[33]. Y aunque éste es un aspecto que está aceptado por los modelos de madurez provenientes de la industria del software[9, 15], es un aspecto que tiene que tener especial foco en los departamentos de entrenamiento.

Por lo tanto, en este trabajo de tesis nos concentraremos en encontrar **un mecanismo que permita a los departamentos de entrenamiento** de las organizaciones de software **justificar la inversión realizada en términos de Retorno de Inversión (ROI)**.

1.2 Importancia del Problema

El Entrenamiento Organizacional es una disciplina que tiene más historia[33] fuera de la industria del software que dentro de ella. Las organizaciones han invertido en entrenamiento del personal para mejorar la calidad de sus productos. La inversión en entrenamiento está justificada por la premisa de que empleados altamente entrenados van a ser más productivos y por lo tanto la organización se va a beneficiar con productos de mayor calidad y con menores costos.

En estos días, las organizaciones invierten grandes sumas de dinero en programas de entrenamiento. Únicamente en los Estados Unidos, las organizaciones invierten un promedio anual de U\$S 100.000 millones [30, 34]. De esta manera, los gestores del entrenamiento se van a ver cada vez más presionados para brindar resultados tangibles a la organización con respecto a la inversión en sus departamentos. En épocas de crisis, la inversión en entrenamiento se verá reducida, pero las organizaciones deberán mantener la efectividad de sus entrenamientos. Ya sea para mantener la productividad y la competitividad de sus recursos, como para mantener la adhesión a los estándares de mejora de procesos.

1.3 Objetivo de este trabajo

Este trabajo propondrá un proceso para mejorar la eficiencia de las intervenciones de entrenamiento, vinculando la información de defectos de los proyectos, con el entrenamiento mediante el análisis causal de defectos.

En los siguientes párrafos desglosamos este objetivo de alto nivel.

- *“mejorar la eficiencia de las intervenciones de entrenamiento”*
Las intervenciones de entrenamiento necesitan estar alineadas con las necesidades de la organización. En este trabajo se presentará un mecanismo para calcular el Retorno de Inversión de las intervenciones de entrenamiento.
- *“la información de defectos de los proyectos”*

Este mecanismo interpretará los defectos de los proyectos de desarrollo como las necesidades de los proyectos. De esta manera será posible diseñar y desplegar entrenamientos orientados a disminuir los defectos, aumentando de esta manera el Retorno de Inversión de estas intervenciones.

- *“vinculando la información de defectos de los proyectos, con el entrenamiento y el análisis causal”*
El análisis causal de defectos será el mecanismo para relevar las necesidades de capacitación a partir de los defectos del proyecto.
- *“Este trabajo propondrá un proceso”*
Un proceso es un conjunto ordenado de pasos que son ejecutados para conseguir un objetivo. Esta propuesta va a presentar el conjunto de pasos que es necesario ejecutar para relevar la información de defectos y calcular el retorno de inversión de la intervención de entrenamiento.

1.4 Aportaciones

La contribución principal de este trabajo está en la obtención del proceso y su validación a través de la experimentación en un entorno de producción real.

Además, como elemento de partida para calcular el ROI vinculamos la producción de software con el entrenamiento de los recursos mediante técnicas de análisis causal.

En el camino a estas validaciones se obtuvieron los siguientes resultados:

- Un proceso para el cálculo del ROI en el que contempla la variación natural de los procesos de desarrollo
- Estudio sobre el impacto del entrenamiento en la habilidad de los recursos de una fábrica para clasificar defectos.
- Estudio del impacto en la capacidad de clasificación de los equipos cuando estos cuentan con individuos no entrenados.
- Una prueba de concepto para una herramienta que procesa lenguaje natural para clasificar defectos dados una Taxonomía de clasificación. Se consiguieron resultados similares o superiores a los de clasificadores expertos.
- Un proceso de despliegue para el proceso propuesto, para que las organizaciones de software puedan reproducir nuestros resultados.
- Aplicación de las No Conformidades de Proceso como predictores de la distribución de defectos de los proyectos de desarrollo.

1.5 Publicaciones asociadas a este trabajo

Como se verá en el capítulo 6, para validar nuestra propuesta se aplicó el proceso en un entorno de producción real en una organización de software. Esta validación fue completada por experimentos realizados en el laboratorio.

La siguiente lista representa las publicaciones realizadas:

Publicaciones en revistas:

[35] Matalonga, S., and San Feliu, T. 'Calculating Return on Investment of Training using process variation', IET Software, Accepted for publication. (2011).

Publicaciones en conferencias:

[36] Matalonga, S., and San Feliu, T 'Using Process and Product Quality Assurance Audits to Measure Process Business Alignment'. Proceedings of the International Conference of Systems and Software Engineering and their Applications, 2010.

[37] Matalonga, S., San Feliu, T., and Rus, V.: 'Automatic Defect Classification: An Experience Applying Natural Language Processing'. Proc of the CLEI 2009, Pelotas, Brazil. 2009.

[38] Matalonga, S., and Feliu, T.S.: 'Using defect data to drive Organizational Training efforts'. International Conference on Software Engineering Theory and Practice, July 7-10 Year 2008 pp. 61-68

[39] Matalonga, S., and SanFeliu, T.: 'Linking Return on Training Investment with Defects Causal Analysis'. 20th Conference of Software Engineering and Knowledge Engineering, July 1-3 Year 2008. pp. 42-47

Publicaciones en congresos

[40] Matalonga, S., and San Feliu, T.: 'Using defects to account for the investment in support process areas'. In. Europe SEPG 09, Prague, Czech Republic, 10/06/2009 2009

[41] Matalonga, S., and San Feliu, T.: 'Defect Driven Organizational training'. In. Europe SEPG 2008. Munich, Germany. June 2008.

La siguiente imagen presenta como las publicaciones realizadas validan nuestra propuesta de tesis:

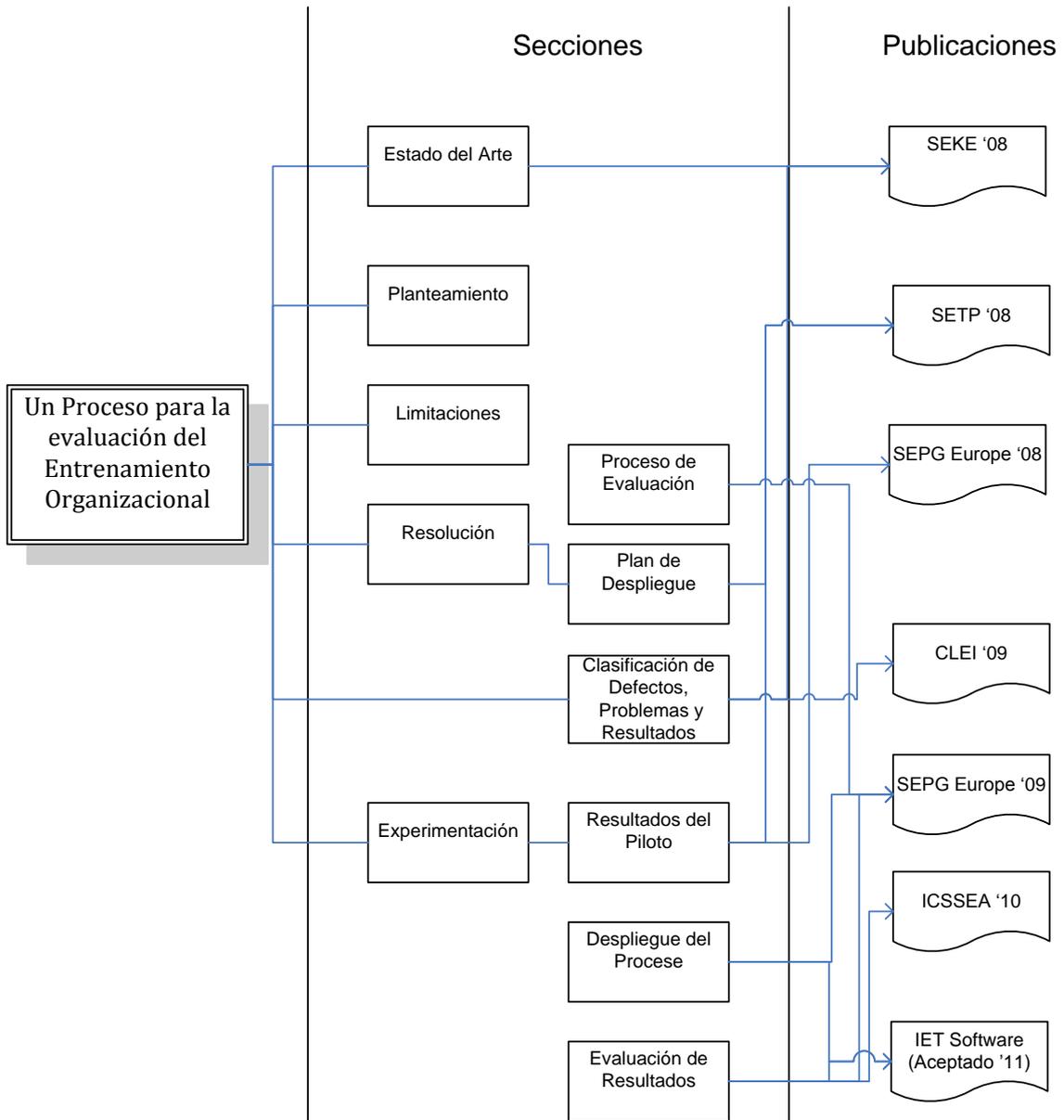


Figura 4 Gráfico de resultados publicados

2 Estado del Arte

La investigación del estado del arte se organizó alrededor de tres cuerpos de conocimiento.

En primer lugar fue necesario investigar las propuestas de evaluación de entrenamiento disponibles en la bibliografía de desarrollo de recursos humanos.

El segundo grupo de conocimiento consiste en agrupar el estado del arte de los modelos de mejora de procesos para identificar los conceptos anteriores en el desarrollo de estos modelos.

En el último grupo, se describen los mecanismos de resolución utilizados en el desarrollo de la experimentación. Este grupo incluye la investigación en análisis causal, retorno de inversión en iniciativas de mejora de procesos y gestión cuantitativa de procesos.

2.1 Evaluación del entrenamiento organizacional

Las organizaciones invierten un promedio de 100 millones de dólares anuales en entrenamiento [34]. *La inversión en entrenamiento está justificada por la premisa de que si los recursos humanos se encuentran mejor entrenados, estos van a producir un producto con menos defectos y de mayor calidad.*

La Sociedad para el entrenamiento y el desarrollo (ASTD - American Society for Training & Development¹) publica anualmente un reporte sobre el estado de la industria del entrenamiento [42, 43, 44, 45, 46, 47, 48, 49]. La siguiente gráfica muestra que, más allá de un enlentecimiento asociado a las crisis económicas mundiales del 2001, *la inversión de las organizaciones en entrenamiento medida en dólares constantes, ha aumentado en los últimos años.*

¹ www.astd.org

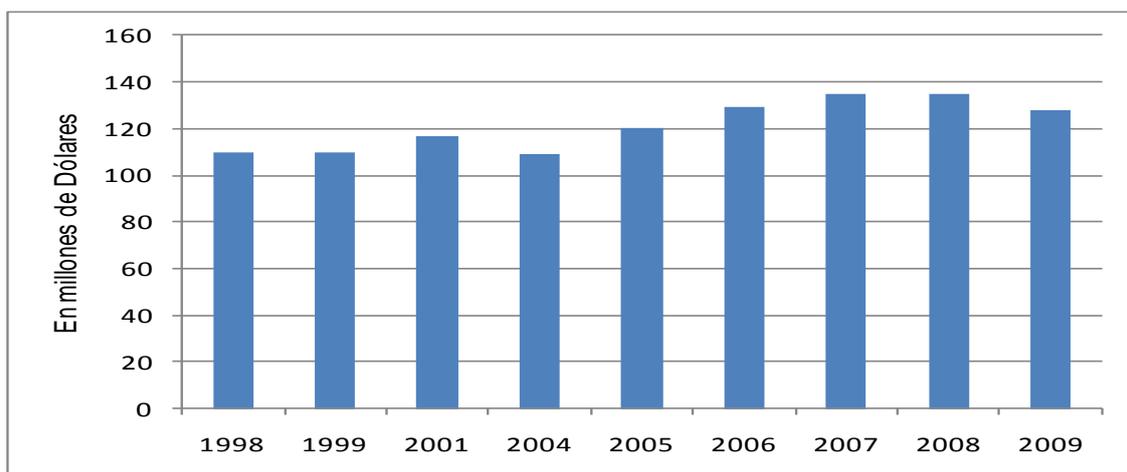


Figura 5 Inversión en entrenamiento en Millones de Dólares

El problema con el volumen de inversión en entrenamiento es *que los gestores del entrenamiento cada vez estarán más presionados por demostrar como la inversión en su sector es redituable para su organización* [43, 50, 51]. La capacidad para evaluar el entrenamiento brindado a los recursos humanos resultará clave para el éxito de los programas de entrenamiento.

En las siguientes secciones se presenta el detalle de las principales propuestas para manejar la evaluación del Entrenamiento Organizacional.

2.1.1 Evolución de los modelos de evaluación del entrenamiento

En [52, 53] se muestra un estudio de la evolución de la evaluación del entrenamiento. El mensaje de estos trabajos es *que el estado del arte está bastante más desarrollado que el estado de la práctica*. En [52] se argumenta como los avances en la teoría no van acompañados de resultados publicados por las organizaciones. Otro punto interesante de este trabajo, es que clasifica la evolución de la teoría de la evaluación del entrenamiento en tres etapas (ver Tabla 1).

La primera está fuertemente asociada al desarrollo de los modelos de evaluación del entrenamiento. Esta etapa es ampliamente dominada por el modelo de cuatro niveles de Kirkpatrick presentado en 1959[29].

La segunda etapa está determinada por el desarrollo de procesos con el objetivo de conseguir resultados de medidas concretas con respecto a la evaluación del entrenamiento. En esta etapa destaca al trabajo de Phillips [54] sobre retorno de inversión.

Finalmente, la tercera etapa está dirigida por un desarrollo de métodos de evaluación orientados a ofrecer a las organizaciones métodos alternativos para realizar evaluaciones del entrenamiento. Estos métodos se apoyan en la falta de resultados generalizados y la brecha existente entre el estado del arte y el estado de la práctica. Estos modelos proponen marcos de referencia alternativos al modelo de 4 niveles.

Etapa	Autor más significativo	Año
Desarrollo orientado a la práctica: modelo de entrenamiento de 4 niveles	Donald L Kirpatrick[28]	1959- 1980s
Desarrollo de procesos con foco en Retorno de Inversión	Jack Phillips[54]	1980s 1990s
Desarrollo orientado a la investigación Búsqueda de otros métodos	Wang and Sptizer[53] Holton [55, 56, 57]	1990s -

Tabla 1 Etapas en el desarrollo de los modelos de evaluación del entrenamiento

2.1.2 El modelo de evaluación de entrenamiento de cuatro niveles

El modelo de evaluación del entrenamiento organizacional de Kirkpatrick fue publicado por primera vez en 1959[29] y ha sido revisado por el propio autor en los últimos años[28]. Desde entonces ha sido exitoso en guiar los esfuerzos de las organizaciones para justificar los resultados de inversión de entrenamiento. En el modelo, Kirkpatrick propone 4 niveles para organizar la información de los entrenamientos realizados.

Los primeros dos niveles (Reacción y Aprendizaje) son denominados “orientados al participante”, porque el objetivo de estos niveles es obtener medidas en relación a las personas que asisten al entrenamiento. Los últimos dos niveles están “orientados a la organización”, pues el objetivo de estos niveles es conseguir medidas para que la organización pueda tomar decisiones con respecto a la inversión en entrenamiento.

- **Reacción:** El objetivo es medir la “primera impresión” de los asistentes al evento de entrenamiento.

Las herramientas de medición más comunes utilizadas en este nivel son las encuestas de satisfacción donde al asistente se le realiza una serie de preguntas con respecto al Docente, al ambiente de trabajo o a su disposición. Las mediciones en este nivel son mayormente subjetivas y dependen mucho de las actitudes del asistente con respecto al entrenamiento.

- **Aprendizaje:** El objetivo es medir el conocimiento adquirido por el participante durante el entrenamiento.

Dependiendo del entorno, puede existir la necesidad de conocer cuál es el nivel de conocimiento del sujeto antes de asistir al entrenamiento. Si éste es el caso, es necesario contar con mediciones del nivel de conocimientos antes y después de impartido el entrenamiento. Otro mecanismo sugerido es utilizar grupos de control que no reciban el entrenamiento y medir entonces la diferencia de conocimientos media entre los sujetos entrenados y los no entrenados.

En cualquiera de las dos maneras, el mecanismo para tomar las mediciones consiste en pruebas de conocimiento como las que se realizan en las Universidades

para medir si el alumno aprueba o no. La salvedad radica en que este tipo de pruebas son mejores para medir conocimientos, pero para medir adquisición de habilidades, hay que realizar otro tipo de mediciones, como simulaciones, o ejecuciones de las tareas observadas por un experto.

- **Transferencia:** El objetivo es medir cuánto del “Aprendizaje” adquirido en la etapa anterior pudo ser “transferido” a la organización.

A diferencia de las medidas anteriores, transferencia no puede ser medida inmediatamente después de realizado el entrenamiento. El problema radica en que existe un tiempo prudencial entre que el individuo recibe el nuevo conocimiento/habilidad, y el momento en que está preparado para aplicarlo en su entorno de trabajo.

Cuando se mide transferencia también, hay que tener en cuenta que la transferencia puede ser afectada por otros factores. Por ejemplo, el individuo puede no estar motivado para aplicar el nuevo conocimiento en la organización, o bien la organización no le ha brindado las oportunidades para aplicar este nuevo conocimiento o él puede no haber identificado las oportunidades para aplicarlo.

Por lo tanto, la mejor manera sugerida [28] es hacer evaluaciones 360, donde se diseñan una serie de preguntas orientadas a evaluar las variables relevantes que puedan afectar la transferencia del entrenamiento en la organización.

- **Resultado:** El objetivo de contribuir con los objetivos económicos de la organización.

El desafío aquí es poder aislar la contribución del entrenamiento a los objetivos de negocio de la organización, de las otras iniciativas de la organización. El autor argumenta que conseguir esto puede no ser económicamente viable, por lo que se debiera aspirar a demostrar “más allá de toda duda razonable”[28] la forma en que el entrenamiento ha contribuido con los resultados de la organización. Para Kirkpatrick la manera de conseguir esto es conseguir transparencia en todas las suposiciones tomadas, de manera que los cálculos no sean cuestionados, y las suposiciones cambiadas según las necesidades de la alta gerencia.

La sencillez de su presentación, ha conseguido que el modelo de 4 niveles continúe siendo el modelo de referencia a la hora de hablar de evaluación del Entrenamiento Organizacional.

2.1.2.1 Críticas y trabajos sobre el modelo de cuatro niveles

En los primeros años desde la publicación del modelo de 4 niveles, los resultados publicados se concentraron en la aplicación del modelo a distintas áreas de conocimiento. El desarrollo de la teoría de evaluación del entrenamiento tiene un empuje fuerte a partir de la década del 1980 donde muchos autores comienzan a publicar trabajos buscando maneras de mejorar el modelo de cuatro niveles.

Uno de estos autores que ha trabajado bastante en el desarrollo del modelo de Kirkpatrick es Alliger. En 1989, él argumenta que conseguir medidas en todos los cuatro niveles puede no resultar eficiente desde el punto de vista de costos [34]. El propio Kirkpatrick reconoce esto mismo en un artículo publicado en 1997[58]. Para ahorrar el paso de tener que

encontrar medidas en todos los niveles, Alliger intenta establecer relaciones causales entre ellos, concluyendo que:

- 1) Los niveles están organizados ascendentemente con respecto a las necesidades de información de la organización.
- 2) Existe una relación causal entre los niveles de Kirkpatrick.
- 3) Existe una relación de temporalidad entre los primeros dos niveles, y los últimos.
- 4) Medidas positivas en un nivel no necesariamente están relacionadas con medidas positivas del nivel anterior.

Sin embargo, el último resultado es refutado tanto por [59] como por el propio Alliger que en 1997[30]. El objetivo de ambos trabajos es encontrar una relación estadística entre los resultados de los niveles inferiores para “ahorrarse” el tener que obtener medidas de todos los niveles para llegar al nivel de “Resultados”. De todas maneras, aunque sus trabajos concluyen que en ciertos casos es posible encontrar relaciones entre las medidas de los niveles inferiores con los resultados de los niveles superiores, se mantiene constante el hecho de que son pocos los ejemplos de organizaciones que realmente hayan conseguido obtener medidas en los cuatro niveles. De hecho los principales ejemplos que se citan corresponden a los mismos casos de estudio que son presentados en el libro[28]. En [60] se estudia la relación entre el nivel de reacción y el nivel de transferencia y se discute la aplicación de las medidas de reacción como predictivas del resultado del entrenamiento – y de esta forma se le da soporte al modelo de Holton (ver 2.1.4) que explícitamente ha suprimido el nivel de reacción como uno de los resultados a medir.

Por otro lado, en [50] los autores estudian las condiciones en que a una organización le convendría obtener medidas en nivel 4. En este artículo se plantea que no todas las organizaciones debieran apuntar a obtener medidas de nivel 4, y que puede ser que organizaciones en las que sea rutina tomar datos de esfuerzo y defectos tengan más fácil el camino para obtener medidas en este nivel. En [61] retoman el resultado de que los objetivos de evaluación del entrenamiento tienen que estar asociados con las necesidades de información de las organizaciones y argumentan que no todas las organizaciones necesitan medidas en los cuatro niveles y por lo tanto puede ser éste el motivo por el que no se cuente con suficientes casos de éxito de la aplicación completa del modelo.

Quizás las críticas más importantes al modelo de Kirkpatrick vengan desde el punto de vista de [34, 55, 62] donde los autores argumentan que más que un modelo de evaluación del entrenamiento, es una taxonomía de clasificación de los resultados posibles de las evaluaciones de entrenamiento. Su argumento se basa en que Kirkpatrick no propone técnicas, ni herramientas para apoyar a que los practicantes consigan obtener medidas en ninguno de los tres niveles.

Finalmente en el 2004 [63] realiza un estudio de la evolución de la investigación sobre el modelo de Kirkpatrick, concluyendo que el motivo por el cual este modelo se ha mantenido activo radica en la simpleza con la que logra transmitir los objetivos de la evaluación del entrenamiento, la misma conclusión con la que [61] cerraban su trabajo.

Por lo tanto, [63] dice que la sencillez del modelo de Kirkpatrick es útil para ser utilizada como heurística, pero que las organizaciones deben adaptar los niveles según sea apropiado para su evaluación.

Por otro lado, Phillips [54] propone agregar un quinto nivel al modelo de Kirkpatrick denominado Retorno de Inversión en entrenamiento. El objetivo de Phillips es conseguir un nivel de “resultados” en el cual el gestor del Entrenamiento Organizacional comunique su contribución a los resultados de la organización en términos numéricos.

2.1.3 El retorno de inversión en entrenamiento

El retorno de inversión en entrenamiento puede considerarse relacionado al nivel de resultados de Kirkpatrick, por más que este último explícitamente especifique que no considera que el Retorno de Inversión no sea parte del nivel de resultados. Quizás es por esta afirmación que Philips [54] identifica al retorno de inversión como un quinto y nuevo nivel.

Según Philips, calcular el retorno de inversión en entrenamiento puede no ser una tarea fácil, por lo que en su trabajo[54] incluye un proceso de cuatro etapas para calcular el retorno de inversión.

1. **Recolección de Datos:** Para Phillips es importante que la recolección de datos se realice en todas los niveles de Kirkpatrick. El plan de recolección de datos debe desarrollar las actividades necesarias para conseguir este objetivo. El objetivo fundamental de esta etapa es conseguir, almacenar y dejar disponibles la información necesaria para ser utilizada en las próximas etapas del proceso de cálculo del ROI.
2. **Aislar las variables que pueden afectar al entrenamiento:** Con el objetivo de calcular el ROI eficientemente se deben aislar las variables que pueden tener influencia sobre el entrenamiento. En cada uno de los 4 niveles, pueden existir variables externas que puedan afectar el resultado de la evaluación. Por ejemplo a nivel 2, los conocimientos previos de los participantes pueden afectar las medidas de aprendizaje. Phillips propone algunas técnicas para aislar las variables, la más común de ellas es el uso de grupos de control, aunque argumenta que no en todas las oportunidades se pueden utilizar los. Principalmente por razones de costo.
3. **Convertir las variables en datos numéricos:** Según Phillips, existen dos tipos de variables que pueden afectar el cálculo del ROI. Las variables que generan datos “Hard”, y las que generan datos “Soft”. “Hard data”, son aquellas que son salidas tangibles de los procesos productivos de las organizaciones, y por lo tanto es sencillo convertir su valor a datos numéricos. Las variables “Soft”, son efectos secundarios de los procesos productivos. En esta categoría se encuentran medidas como el índice de rotación de los empleados de la organización, satisfacción del empleado con el trabajo, o compromiso con la organización. Para poder utilizar las variables “Soft” en el cálculo del ROI, se debe convertir su impacto en datos numéricos. Para Phillips, es fundamental que esta conversión se realice de la forma lo más transparente posible. El autor recomienda la utilización de escalas en tablas, y la aprobación de estas tablas por parte de la gerencia, o los clientes que van a estudiar los resultados del ROI.
4. El último paso del proceso es la aplicación de la fórmula del cálculo del ROI.

$$(1) \quad ROI = \frac{(Beneficios - Costos)}{Costos} * 100$$

Por otro lado, [64] aplica la fórmula clásica del ROI tanto para estimar los retornos de una intervención como para calcular el ROI real luego de 6 meses de realizada la intervención. El aporte de estos autores radica en el uso del ROI como elemento de decisión para continuar con el entrenamiento.

2.1.3.1 Críticas y trabajos sobre el modelo de retorno de inversión

El modelo de Phillips ha sido criticado por la literatura por poner demasiado énfasis en los métodos numéricos[65], aunque en el método de Phillips dicta que los resultados no numéricos sean traducidos a valores numéricos para calcular el ROI, existen oportunidades en que esto no puede realizarse. Asociado a esto, en [62] se fundamenta que la dificultad de aplicar técnicas de Retorno de inversión en entrenamiento se encuentra en la subjetividad de la estimación de los Beneficios. Por ejemplo cuando se entrena para mantener la rotación o para mantener conocimiento en la organización, pueden ser casos en que el sistema de métricas de la organización no esté preparada para establecer este tipo de equivalencias y tenga que recurrir a métodos subjetivos de quien realiza el cálculo del ROI. De todas maneras tanto Phillips[54] como Kirkpatrick[28] argumentan que todas las decisiones subjetivas tomadas por el gestor del entrenamiento debe estar documentadas para que puedan ser evaluadas por la alta gerencia.

En [62, 66] se presentan métodos alternativos a la evaluación del entrenamiento que no están basados en métodos numéricos como el de Phillips.

Más allá de criticar al modelo de Phillips en sí, lo que usualmente es criticado es la aplicabilidad del cálculo de Retorno de Inversión en las intervenciones de entrenamiento[51, 62, 67] dada su tendencia a sostener métodos completamente numéricos. En [64], se intenta limitar esta debilidad por medio del estudio de estimación. Sin embargo, no se contemplan los costos de este estudio en el posterior cálculo del ROI.

Por otra parte, a pesar de que Kirkpatrick específicamente excluye al cálculo del Retorno de Inversión como parte de su nivel 4[28]. En esta propuesta se considerará al cálculo del ROI como el nivel 4 de Kirkpatrick. La intención es tratar de mantener la sencillez y la transparencia en la comunicación de los resultados. Al fin y al cabo, el objetivo de esta propuesta está en aportar mecanismos para comunicar los resultados del entrenamiento en términos de datos tangibles y cuantificables.

Por último, con respecto a la formula del ROI (1), apoyamos el punto de vista de [68, 69] que es fundamental enmarcar los cálculos de ROI en un marco temporal para poder comparar la inversión en entrenamiento con otras inversiones que podrá hacer la organización.

2.1.4 El modelo de evaluación de Holton

En un trabajo ya mencionado, [53] identifican el último periodo de desarrollo de la teoría de evaluación del entrenamiento como un momento en que los investigadores son guiados por el objetivo de desarrollar nuevos modelos de evaluación de entrenamiento. Ellos afirman que la sencillez del Modelo de 4 niveles significó su mayor éxito y al mismo tiempo el motivo por el cual la teoría de Evaluación del Entrenamiento no evolucionó más allá de él. Y únicamente las presiones económicas de final del milenio pudieron empujar al nuevo desarrollo de la teoría.

El más significativo modelo alternativo de evaluación de entrenamiento es el propuesto por Holton [55, 56, 57]. El modelo de Holton se distancia del modelo de evaluación del entrenamiento de Kirkpatrick. A diferencia de las otras propuestas de modelos que construían sobre la base teórica de los 4 niveles. Holton propone un esquema en el que no se distinguen los 4 niveles clásicos de Kirkpatrick. Holton entiende que el modelo de 4 niveles no es una metodología para la evaluación de entrenamiento sino una taxonomía de los posibles tipos de evaluación. Para Holton, una metodología debiera permitir realizar predicciones del estado futuro a partir de sus unidades básicas de construcción. Y esto no es posible con el modelo de cuatro niveles no es posible.

Por lo tanto, el modelo de Holton, intenta proponer un nuevo paradigma que permita sacar conclusiones a partir de sus elementos básicos de construcción. A estos elementos Holton los llama "resultados". Para Holton, existen 3 tipos de resultados por los cuales se debiera evaluar el entrenamiento:

- Resultados de Aprendizaje
- Resultados Individuales
- Resultados de la Organización.

Además de estos 3 niveles de resultados, el modelo de Holton reconoce que existen variables que pueden favorecer o desmerecer la intervención del entrenamiento, y que deben ser medidas para poder aislar los resultados del entrenamiento en el individuo.

- Habilidad y diseño del entrenamiento, la habilidad del entrenador, así como el diseño de la intervención pueden afectar los resultados de aprendizaje.
- Capacidad de transferencia: Implica obtener medidas con respecto a las facilidades que la organización brinda para que el individuo pueda aplicar el conocimiento aprendido en el entorno de la organización. Y también, las capacidades del individuo para transferir ese conocimiento dentro de la organización.
- Relación con los objetivos de negocio: La relación con los objetivos de negocio de la organización pueden afectar cualquiera de los tres resultados principales.

Otras variables a tener en cuenta son las que Holton identifica como Variables de Ambiente a las variables que pueden influir en el ambiente de la organización para favorecer o desmerecer la efectividad del entrenamiento. A estas las agrupa en dos:

- Motivación para aprender: Apunta a identificar la actitud del sujeto a ser entrenado en el contexto de la organización. Por ejemplo, actitud del individuo, capacidad del individuo, actitud del entorno con respecto al entrenamiento, son

algunas de las variables que podrían medirse para intentar valorar la motivación para aprender.

- Entorno de la organización: Apunta a identificar las características del entorno en la organización que puedan afectar los resultados del entrenamiento. Ejemplo de estas variables podrían ser: Relación de la intervención con los objetivos de negocio, utilidad esperada, aplicación de los conocimientos.

2.1.4.1.1 Críticas al modelo de Holton

En [70], Holton desarrolla un mecanismo para medir las variables que afectan a la transferencia del entrenamiento en las organizaciones. Este estudio valida la porción del modelo que afecta a la transferencia del entrenamiento a la organización. Pero no se ocupa de validar el modelo en forma completa.

Por otro lado [56], el autor compara su modelo con el de Kirkpatrick, y reconoce que no tiene la sencillez del anterior. Pero el objetivo del modelo de Holton es intentar medir las variables externas que influyen en la medición de los resultados del entrenamiento. Especialmente a nivel organizacional.

Con respecto a la aplicabilidad de su modelo, Holton argumenta que la falta de herramientas que acompañen al modelo de Kirkpatrick es la razón por la que no existan en la academia registros contundentes de aplicaciones de ese modelo. Cada organización termina implementando su sistema de medidas como mejor puede en función de los 4 niveles. El objetivo del modelo de Holton, está en desarrollar la teoría de la evaluación del entrenamiento brindando una metodología que permita desarrollar a los practicantes, obtener medidas y generar conclusiones a partir de los elementos de su teoría.

El Modelo de Holton no ha sido validado completamente. En el mismo artículo [56] el autor presenta estudios de la literatura que soportan partes de su construcción pero no el modelo completo.

2.1.5 Reflexiones con respecto al entrenamiento organizacional

En las secciones anteriores se mostró como las empresas consideran que mantener a los recursos entrenados es importante para conseguir obtener productos de mejor calidad; al mismo tiempo se presentó como la inversión en entrenamiento ha ido en aumento en los últimos años. Sin embargo, el problema que se plantea, y en especial en tiempo de crisis, es cómo demostrar que la inversión en entrenamiento está reeditando o contribuyendo a los resultados de la organización. A pesar de que la historia y el desarrollo de los modelos de inversión en entrenamiento llevan casi medio siglo de desarrollo, éste es todavía un problema abierto para la mayoría de las organizaciones.

En esta tesis se abarca este problema desde el punto de vista de una organización productora de software. Las organizaciones de software presentan los mismos problemas con respecto a la evaluación del entrenamiento organizacional (ver Área de proceso de Entrenamiento Organizacional), sin embargo en esta tesis se propone que al estar una

organización preparada para producir software se cuenta con una serie de ventajas que pueden ser aprovechadas por los gestores de entrenamiento.

En el proceso propuesto por esta tesis, se muestra un mecanismo sencillo y reproducible cuyo objetivo es conseguir medidas del nivel de “Resultados” en el modelo de cuatro niveles. Se brindará un mecanismo para que las organizaciones de software que hayan alcanzado niveles de capacidad adecuados para sus iniciativas de medidas, puedan ser capaces de extender su sistema de métricas para dar servicio a sus departamentos de entrenamiento. Mediante la aplicación del proceso de esta propuesta, las organizaciones podrán obtener medidas en “nivel de resultados”, desafiando así las conclusiones de Alliger de que es necesario pasar por cada uno de los niveles antes de obtener datos en el siguiente.

De todas formas, según se mostrará en la sección Área de proceso de Entrenamiento Organizacional, al establecer Limitaciones con respecto a la capacidad del proceso de entrenamiento, es esperable que estas organizaciones hayan conseguido ya la capacidad de obtener medidas en alguno de los niveles inferiores de Kirkpatrick.

Por otro lado, parte de la dificultad de que no se hayan conseguido resultados consistentes con el nivel de “Resultados”, está en que el alcance de los programas de entrenamiento muchas veces puede ser muy amplio para poder realizar cálculos de Retorno de Inversión. Por eso nuestra propuesta se centra en mostrar Retorno de Inversión de intervenciones de entrenamiento específicas para una unidad productiva (un proyecto de desarrollo de una fábrica de software).

La tabla siguiente resume y compara los modelos de entrenamiento revisados por la literatura.

Autor	Nombre del Modelo	Modelo Valido	Reporta contribución a la Inversión de la Organización	Es simple o complejo	Utiliza la Industria de Software	Aplicado en Organizaciones CMMI Nivel 3 o inferior	Existe guía de pasos para seguir el proceso
Kirkpatrick	Modelo de Evaluación de 4 Niveles	Si	Si	Simple	Si	No	No
Phillips	Retorno de Inversión	Si	Si	Simple	No	No	Si
Holton	Evaluación del Entrenamiento	No	Si	Complejo	No	No	No

Tabla 2 Comparación de los modelos de entrenamiento

2.2 Modelos de mejora de procesos

La mejora de procesos se concentra en mejorar la calidad de los productos a través de la mejora de los procesos utilizados para crear estos productos. La mejora de procesos de software tiene su origen en los modelos de calidad de otras industrias. Por lo pronto, todos los modelos de mejora estudiados siguen de una manera u otra el ciclo de mejora de procesos propuesto por Shewarth[71] y Demming [7]. El ciclo de mejora de procesos se puede describir en estos 5 pasos:

1. Establecer el marco actual de procesos
2. Identificar los defectos de producción.
3. Realizar un análisis causal para identificar el origen de los defectos
4. Probar una modificación al proceso
5. Reiniciar el Ciclo

En la industria del software, los primeros intentos de establecer modelos de mejora de procesos se realizaron en la década del 1980[72]. Durante esa se funda en Estados Unidos el Software Engineering Institute que se iba a convertir en el creador del modelo de Mejora de procesos llamado CMM. La primera versión de este modelo se publica en 1993[10].

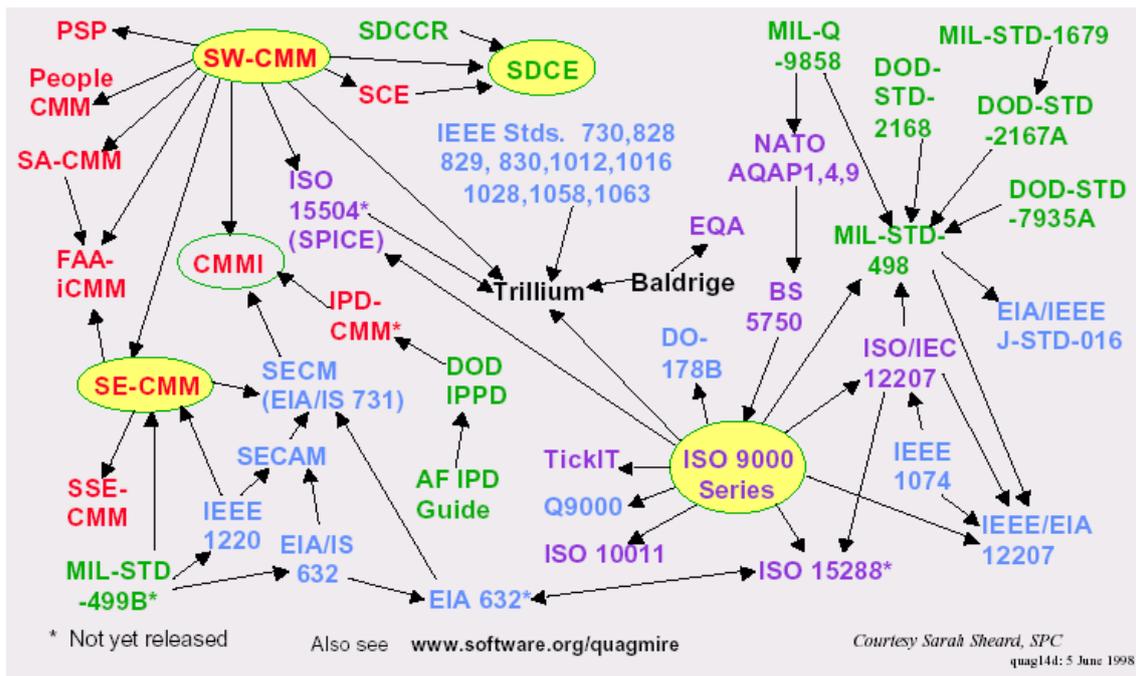


Figura 6 Modelos de Mejora de Procesos

En los siguientes 16 años, otros modelos se van creando en función de las debilidades del modelo CMM. En la Figura 6 se puede ver la diversidad de modelos existentes. En esta sección se presentarán 3 modelos. En primer lugar, los modelos relacionados a la serie 9000 y 15504 de ISO. Aunque no aparece en la figura, estudiar Seis Sigma y su relación con la capacidad de los procesos será importante para la resolución de la propuesta. Finalmente se estudiará el modelo de Capacidad y Madurez Integrado (CMMI).

2.2.1 Modelos de la familia ISO

La International Standards Organization también integra modelos de mejora de procesos. Una de las certificaciones ISO más conocidas es la que se basa en la norma 9000. Cuya primera versión se publicó en 1994. La norma 9000 es una norma de propósito general no específicamente dirigida a la producción de Software. Es una evolución de estándares militares del gobierno británico. Por este motivo es que el estándar es dividido en 3 recomendaciones que abarcan entornos de producción (9001); entornos de desarrollo (9002) y entornos de inspecciones o testing (9003). En los años 1998 y 2000 se realizaron dos revisiones importantes de la norma. En el 1998 se extiende el concepto de ISO de control de calidad a través de revisiones de conformidad a mejora de calidad a través de la implementación de acciones preventivas. La versión del 2000[11] agrupa los tres modelos en uno solo. Esta versión introduce también en concepto del sistema de procesos de la organización, favoreciendo la orientación al proceso por parte de las organizaciones en lugar de orientaciones funcionales.

La última versión del Estándar ISO 9000 es la versión aceptada el 13 de Noviembre de 2008[73]. Al igual que las últimas versiones, se subdivide en 8 grandes capítulos llamados Cláusulas, y estas Cláusulas se encuentran divididas en Sub cláusulas. Las primeras 3

Cláusulas de la norma describen el alcance, los términos y las definiciones que se van a utilizar en el resto del cuerpo. Las otras cláusulas describen los requerimientos que un sistema de gestión que cumple con ISO9000:2008. Los grupos de Cláusulas son:

4. Sistema de Gestión de Calidad. Este capítulo describe los requerimientos de manuales, documentos y control de cambios.
5. Responsabilidad Gerencial. Este capítulo identifica el rol de la alta gerencia en los procesos del Sistema de Gestión de Calidad.
6. Gestión de Recursos. Este capítulo identifica los requerimientos necesarios para asegurar que el Sistema de Calidad cuente con los recursos necesarios para su funcionamiento.
7. Gestión de Producción. Este capítulo identifica los requerimientos de los procesos productivos del sistema de gestión de calidad.
8. Medición, Análisis y Mejora. Este capítulo especifica los requerimientos de verificación del producto producido, y de recolectar la información necesaria para alimentar la mejora de procesos.
9. Las normas TickIt, que son guías para la interpretación de las normas ISO 9000 para la industria del software[74].

Para interpretar las normas ISO 9000 a la industria de software se cuenta por un lado con los guías TickIt[75]. Estas guías presentan interpretaciones para cada una de las clausulas y tienen recomendaciones para la identificación de medidas y métricas para el seguimiento de procesos de software.

Por otro lado, en 1993 se comenzó a trabajar en la norma 15504, originalmente denominada SPICE (Software Process Improvement and Capability detErmination). La primera versión fue publicada en 1995. ISO 15504 es un modelo de mejora de procesos orientado al desarrollo de software, que en la revisión del 2004[12] fue extendido para incluir todas las áreas del ciclo de vida del desarrollo de software, incluyendo por ejemplo: Gestión de Configuración, Gestión de Proyecto.

ISO 15504 está dividido en un conjunto de 9 documentos:

1. Describe la organización general del modelo, y explica los conceptos
2. Contiene las guías y los requerimientos para la realización de evaluaciones de procesos.
3. Describe los requerimientos mínimos necesarios para la realización de una evaluación de procesos basada en la parte 2.
4. Describe cómo interpretar y utilizar el modelo para la mejora continua de procesos.
5. Presenta por medio de un caso de estudio el diseño de un modelo de evaluación de procesos basado en la parte 2.
6. Presenta por medio de un caso de estudio el diseño de un modelo de evaluación de procesos para entornos de producción de sistemas basado en los requerimientos de la parte 2.

7. Describe las condiciones necesarias para realizar evaluaciones de madurez en las organizaciones basadas en los perfiles de las capacidades de los procesos según se describen en la parte 2.
8. La parte 8 tiene el objetivo de interpretar la norma para el uso en la selección de proveedores. Esta parte está en desarrollo.
9. Contiene un glosario de los términos utilizados en los anteriores documentos.

Los requerimientos que los procesos de una organización tiene que cumplir se describen en el Documento 2 de la norma[76]. Existen 5 categorías de procesos que la organización tiene que cumplir: Cliente-Proveedor; Ingeniería, Soporte, Organización y Proyecto.

ISO 15504 determina dos dimensiones de los procesos. La dimensión del proceso, está asociada con las 5 categorías anteriores y se determina mediante el cumplimiento de las sentencias descritas bajo cada una de las 5 categorías de proceso, similar a los Objetivos Genéricos del modelo CMMI.

La dimensión de la capacidad del proceso se determina a partir del cumplimiento de cada uno de estas 5 categorías de procesos. Existen 9 atributos de procesos que se asocian estos grupos de procesos. El cumplimiento de cada uno de estos 5 atributos determinará la capacidad del proceso.

2.2.1.1 Análisis Causal en los modelos de la Familia ISO

Los modelos de la familia ISO referenciados en el apartado anterior especifican requerimientos de análisis causal. Sin embargo, como será mencionado en la sección siguiente, la norma ISO 9000 al ser de propósito general no es educativa con respecto a ninguna de sus cláusulas.

En esta norma, los requerimientos de análisis causal son referenciados dentro de la cláusula 8 (Medición, Análisis y Mejora). Específicamente, las secciones 8.4 Análisis de Datos y 8.5 Mejora. Aunque la referencia a utilización de técnicas de análisis causal no es explícita. La norma permite inferir que para “mejorar continuamente” los procesos la organización debe verificar la aplicabilidad del sistema de gestión de calidad y mediante el “análisis de datos”.

Con respecto a ISO 15504, al ser esta norma una aplicación del modelo de calidad ISO 9000 a entornos productivos de software, era esperable que ISO15504 contuviera requerimientos de análisis causal. Al igual que en la norma ISO 9000, la referencia a la necesidad de análisis causal no es explícita ni educativa y está comprendida en la cláusula 5.6 de la parte 2 de la norma, donde se detallan los requerimientos que un proceso optimizado debe cumplir, entre ellos se destaca la necesidad de que las “data del proceso es analizada para identificar las causas comunes de variación”[76].

2.2.1.2 Entrenamiento en los modelos de la familia ISO

El entrenamiento organizacional también está referenciado dentro de los modelos de la familia ISO como uno de los puntos centrales para asegurar que los procesos sean ejecutados de la manera que estos están definidos.

En ISO 9001:2008 la sección en cláusula 6.2 es la que se refiere a los recursos humanos. En esta cláusula la norma dispone los requerimientos para los recursos humanos de la organización. En especial en la sub cláusula 6.2.2 se determina que la organización es responsable de “determinar las competencias básicas necesarias para realizar las actividades de trabajo que afecten la calidad del producto”[11]. Dentro de este punto también se identifican las necesidades a cubrir por un prospecto departamento de capacitación que deberá ser responsable de mantener los registros de capacitación de los recursos involucrados con las tareas productivas.

En ISO 15504, la referencia a la necesidad de mantener al equipo entrenado es uno de los requisitos para que un proceso alcance nivel de capacidad tres. En la cláusula 5.4.1 se especifica que se deben “identificar las competencias básicas para realizar el proceso”. Y en la cláusula 5.4.2, se especifica que la asignación del personal a las tareas del proceso debe tener en cuenta las competencias y la educación de cada uno de los individuos.

Ambos modelos presentan requerimientos de capacitación pero desde diferentes puntos de vista. ISO 9000, presenta los requerimientos desde un punto de vista organizacional, al tiempo que ISO 15504 los presenta en función de la adopción y capacidad de los procesos.

En relación a la propuesta de este trabajo, ninguno de los dos presenta requerimientos ni da guías de cómo construir o desarrollar un departamento o área de entrenamiento que tenga bajo su responsabilidad la planificación y ejecución del entrenamiento. Por lo tanto, no existen expectativas de evaluación de intervenciones de entrenamiento más allá de los requerimientos de mantener los registros del personal.

2.2.1.3 Críticas a los modelos de la familia ISO

En primer lugar, el problema con ISO 9000, es que no está orientado a Software, sino a entornos de producción. En [77] realizaron un estudio entre las primeras empresas de software alemanas en conseguir una certificación ISO 9000, descubriendo que sólo el 40% había introducido mejoras en sus prácticas de desarrollo, pero que sin embargo en cerca del 97% la iniciativa de implementación ISO había cumplido con sus expectativas. Por otro lado los encuestados destacaban que el mayor valor se encuentra en la filosofía de implementar un sistema de calidad como habilitador del cambio organizacional.

Otro problema de ISO 9000, que ISO 15504 comparte en menor medida, es la distribución de los estándares y normas. Por ejemplo, aunque ISO 9000 es el que define los requerimientos del sistema de calidad, es ISO 9003 el que explica cuales son las mejores maneras de desplegar un sistema de gestión de calidad. Además, cada estándar tiene un costo asociado, por lo que en mi opinión esto eleva las barreras de entrada. En [77] los autores refuerzan este comentario en función de la norma 9000.

Con respecto al estándar ISO 15504, parece claro que SPICE no ha sido tan adoptado como CMM/CMMI. Quizá por la diferencia de años entre las primeras versiones de cada modelo, o quizá porque CMMI incorpora las mejoras que SPICE propone a la versión original de CMM (como se ve en la Figura 6 Modelos de Mejora de Procesos). En [78], uno de los autores del modelo CMM original analiza las diferencias entre los dos diferentes

estándares y destaca el “ciclo continuo” como una de las diferencias fundamentales.² Es importante notar los 6 años de diferencia entre este artículo y la fecha de publicación del modelo CMM, para fundamentar las diferencias de experiencias entre uno y otro modelo. En relación a la adopción de SPICE [24], presenta un modelo para estimar el grado de adopción de ISO 15504. Sin embargo concluye que los datos utilizados no son concluyentes con un grado adecuado de certeza.

2.2.2 Seis Sigma

Seis Sigma es al mismo tiempo una filosofía y una combinación de metodología, de medidas y de herramientas. El objetivo de Seis Sigma se centra en satisfacer a los clientes a través de la reducción y la prevención de los defectos de producción y de esta forma mejorar los objetivos de negocio. Para Seis Sigma, un defecto es cualquier variación, en el producto o en el proceso que no satisfaga al cliente o que agregue costos de producción. El nombre “Seis Sigma” viene de la tolerancia que se puede esperar de estos procesos de producción, “sólo 6 desviaciones estándar de la media”.

Cada nivel de UN SIGMA, se entiende como la capacidad de los procesos productivos de la organización. A un nivel de capacidad de 3 sigmas es esperable que se incurra entre 6200 y 67000 defectos de producción cada 1 millón de oportunidades, mientras que un proceso que opera a Seis Sigma es esperable que incurra en 3,4 defectos de producción cada 1 millón de oportunidades.

Seis Sigma está basado en los siguientes tres axiomas[79]:

- Todo puede ser descrito como un proceso
- Todo proceso tiene variación
- Es necesario contar con datos para entender la variación y para tomar decisiones que mejoren el proceso.

Esta filosofía es independiente del entorno productivo en el que se aplique. No depende del dominio, sino de aceptar la definición de los defectos y en respetar los tres axiomas de la filosofía. El problema con adoptar Seis Sigma en la industria del Software, es la falta de madurez inherente a la industria[80].

En el centro de la metodología de Seis Sigma se encuentran el pensamiento estadístico que guía los dos ciclos de Mejora. Por un lado el Ciclo DAMIC (nombrado por sus siglas en inglés de Definir, Analizar, Medir, Mejorar y Controlar), se centra en la mejora de procesos a través del estudio y el análisis causal de los defectos que no son parte de la variación natural del proceso (ver 2.3.4).

El otro ciclo DFSS (por sus siglas en inglés para Diseñar Para Seis Sigma) tiene por objetivo diseñar productos o procesos para cumplir con los objetivos de Seis Sigma. A diferencia del ciclo anterior, DFSS no tiene un ciclo de mejora tan elegante y sencillo. Su

² En la Sección Modelo de Capacidad y Madurez Integrado veremos que este tipo de representación ha sido adoptado por el modelo CMMI

implementación se ha visto dirigida por los diferentes procesos que distintas organizaciones han implementado para conseguir el objetivo.

Un término usualmente asociado con el despliegue de Seis Sigma son los llamados “Eventos Kaizen”. Los eventos Kaizen, son en realidad las sesiones de innovación utilizadas para mejorar los procesos. En los eventos Kaizen se integra a usuarios del proceso en sesiones donde el proceso es analizado para mejorarlo. Técnicamente, ésta es una práctica de implementación que Seis Sigma toma de Lean Manufacturing[81]. Lean es una iniciativa derivada de la industria automotriz Japonesa[82] cuyo objetivo principal es la reducción de las prácticas del proceso que no generan valor para el cliente. Y por esta razón, cuando la definición de defecto de Seis Sigma se asocia con el punto de vista del cliente, la sinergia entre Lean y Seis Sigma permite que ambas iniciativas sean desplegadas en conjunto.

2.2.2.1 Críticas al modelo Seis Sigma

Como se mencionó en la sección anterior, Seis Sigma es independiente del entorno productivo. En este caso, no significa lo mismo que en el caso de ISO 9000. La diferencia está en que Seis Sigma es más general en sus enseñanzas, no es una norma a cumplir, sino una filosofía a adoptar. Los métodos y herramientas de Seis Sigma pueden ser implementados en cualquier entorno, siempre y cuando se respeten los 3 axiomas básicos de la filosofía.

En relación al desarrollo de software, Seis Sigma puede ser utilizado, pero lo primero que hay que hacer es definir que es un defecto desde el punto de vista del análisis de Seis Sigma.

Los modelos de mejora de procesos pueden ser interpretados también en el contexto de la filosofía Seis Sigma. Por ejemplo en [83, 84] el autor dice que el despliegue de cada área de proceso CMMI (ver Modelo de Capacidad y Madurez Integrado) puede ser interpretado como un proceso DFSS de Seis Sigma. Y al mismo tiempo, dado un proceso controlado cuantitativamente, el ciclo DAMIC puede ser utilizado como la base para la interpretación del área de proceso de Gestión Cuantitativa de Proyectos[84] de CMMI.

Con respecto a la adopción de Seis Sigma en la industria de Software, se han realizado estudios. Por ejemplo en [85], los autores realizan un estudio del resultado de la adopción de Seis Sigma en empresas que hayan llevado adelante al menos un proyecto de adopción. El resultado muestra que la gran mayoría, 95%, es capaz de reportar ahorros en costo. Pero al mismo tiempo sólo el 40% de los encuestados es capaz de responder con su costo de calidad, dato que el autor interpreta como falta de madurez en que estas empresas, a pesar del éxito relativo de la adopción de Seis Sigma, continúan con la cultura de finalizar los proyectos en primer lugar. Este estudio sin embargo no es significativo con respecto al número de organizaciones de software que hayan adoptado Seis Sigma en general. Por otro lado, [86] argumenta que el relativo bajo grado de adopción de Seis Sigma – al igual que el anterior no presenta datos concretos con respecto al grado de adopción – se debe a la falta de una metodología clara para seguir, por lo que propone una amalgama de metodologías y herramientas para facilitar la adopción de Seis Sigma por parte de empresas de Software.

2.2.3 Modelo de Capacidad y Madurez Integrado

El modelo de Capacidad y Madurez integrado es una evolución del modelo de Capacidad y Madurez presentado por el SEI en 1993[10]. Actualmente, la versión oficial es la número 1.3 que fue liberada en Noviembre de 2010[87]. La Figura 6 Modelos de Mejora de Procesos, muestra como la versión actual del modelo no sólo se ha beneficiado de la experiencia del SEI con el modelo anterior, sino que ha adoptado conceptos de otros modelos. Un ejemplo de esto es la representación continua; en [78] (publicado antes de la liberación de la primera versión del CMMI) el autor del CMM indica cómo “un modelo de procesos continuo” podía prestar más flexibilidad a las organizaciones que adoptan la mejora de procesos basada en CMMI.

El modelo de Capacidad y Madurez de Software integrado está organizado alrededor de 22 Áreas de Proceso. Un área de proceso es un conjunto relacionado de prácticas que, al implementarse en conjunto satisfacen un conjunto de objetivos considerados importantes en esa área. Estas prácticas construyen el mensaje del modelo CMMI y les indican a los practicantes de la mejora de proceso QUE es lo que hay que hacer para cumplir con sus requerimientos. El COMO realizar estos objetivos es dejado a libertad de los usuarios del modelo para que puedan adaptarlo a las realidades de su organización. *De esta manera el modelo es directivo en qué hay que hacer pero los mecanismos de cómo realizar los requerimientos del modelo son dejados para los practicantes.*

El modelo CMMI tiene dos formas de ser leído según como se agrupan las áreas de proceso; estas formas se denominan *Representaciones*. La representación escalonada, es la representación donde las organizaciones van incrementando su nivel de madurez, y donde cada área de proceso está asociada a un nivel de madurez específico. En la representación continua, concepto que CMMI toma de ISO 15504, el practicante puede seleccionar qué áreas de proceso son críticas para su negocio y mejorarlas individualmente.

En la representación escalonada, las áreas de proceso se agrupan en 5 niveles de madurez. Cada nivel describe las características esperadas de una organización que ha alcanzado el grado de madurez requerido para encontrarse a ese nivel. El nivel de madurez inicial tipifica a organizaciones que son esencialmente reactivas y que no tienen procesos institucionalizados. En el nivel de madurez Manejado, los procesos son definidos a nivel de cada uno de los proyectos de la organización. En el siguiente nivel, Definido, aparece por primera vez el concepto de Organización como una entidad que se preocupa por que sus procesos estén definidos a través de sus proyectos de desarrollo. Los niveles 4 y 5 son denominados niveles de madurez superiores o de alta madurez. En el nivel 4, la organización ha seleccionado sus áreas de mejora y los procesos pertenecientes a esas áreas son gestionados estadísticamente. El nivel Optimizado, es el nivel que pone en práctica los conceptos de mejora continua, análisis causal e innovación, en la organización.

En la representación continua, los niveles no se denominan de madurez, sino de capacidad. CMMI define a un proceso como “las actividades que pueden ser mapeadas a la implementación de una o más prácticas de las áreas de proceso del modelo CMMI, y que pueden ser utilizadas con fines de mejora de procesos o con fines de evaluación”[9]. De esta definición se desprende que mientras que la madurez es una medida a nivel de la

organización, la capacidad es una medida a nivel de los procesos. Los niveles de capacidad de los procesos de una organización son 6 en el modelo CMMI. Para ilustrar la relación que existe entre los niveles de madurez y de capacidad, se afirma que si una organización se dice que ha alcanzado un nivel de madurez 3, significa que ha alcanzado un nivel de capacidad 3 en cada uno de los procesos asociados con las 18 áreas de proceso que el modelo ha identificado se encuentran a nivel de madurez 3.

Cada área de proceso, se encuentra a su vez dividida en objetivos específicos y objetivos genéricos. Los objetivos genéricos son comunes a todas las áreas de proceso, y su satisfacción determina la capacidad de los procesos. Los objetivos específicos determinan los requerimientos de procesos para esa área de proceso. Son los objetivos específicos los que nos comunican el QUE hay que hacer para cumplir con los requerimientos del modelo CMMI. El cumplimiento de todos los objetivos específicos, dentro de todas las áreas de procesos pertenecientes a un nivel determinado, determina ese nivel de madurez.

2.2.3.1 Grupos de Áreas de Proceso

El modelo CMMI también organiza las áreas de proceso en función del área de conocimiento a las que éstas refieren. Existen 4 grupos de áreas de proceso en el modelo CMMI: Gestión de Proyecto, Ingeniería; Soporte y Organizacional.

El grupo de Gestión de Proyecto se encarga de establecer las actividades relacionadas con la planificación, el seguimiento y el control de los proyectos de desarrollo.

Las áreas dentro del grupo de Ingeniería se ocupan de mostrar las mejores prácticas para mejorar la gestión del desarrollo.

Soporte, implica las áreas de conocimiento que no están directamente relacionadas con la gestión ni con el desarrollo del producto pero que es importante tener en cuenta para el desarrollo de los proyectos.

Finalmente el grupo de áreas de proceso Organizacional se encarga de establecer y mantener la infraestructura necesaria para establecer los procesos de mejora de procesos para cada una de las áreas de proceso anteriores.

A continuación se presentarán las cuatro áreas de proceso en las que aplican al proceso propuesto en este trabajo.

2.2.3.1.1 Área de proceso de Verificación

El área de proceso de Verificación, es un área de proceso que pertenece a la categoría de Ingeniería y se encuentra a nivel de madurez 3. El propósito del área de proceso de verificación, es establecer las prácticas necesarias para asegurarse que el producto cumpla con sus especificaciones en el entorno de producción[9]. El área de proceso de Verificación incluye actividades relacionadas con el Testing de Software (primer objetivo específico) y con las revisiones de pares (segundo objetivo específico).

El tercer objetivo específico se enfoca en analizar las actividades realizadas en los dos primeros para asegurarse que los objetivos del proyecto con respecto a las actividades de Verificación se han cubierto.

SG1 Preparar la verificación	
SP 1.1	Seleccionar los productos para la verificación
SP 1.2	Establecer el entorno de verificación
SP 1.3	Establecer los procedimientos y los criterios de verificación
SG2 Realizar Revisiones de Pares	
SP 2.1	Prepararse para las revisiones de pares
SP 2.2	Realizar las revisiones de pares
SP 2.3	Analizar los datos de las revisiones de pares
SG3 Verificar los productos seleccionados	
SP 3.1	Realizar la verificación
SP 3.2	Analizar los resultados de la verificación

Tabla 3 Área de Proceso de Verificación

2.2.3.1.2 Área de proceso de Análisis Causal y Resoluciones

El área de proceso de Análisis causal y Resoluciones (CAR) es un área de proceso del grupo de Soporte y se encuentra a nivel de madurez 5. El objetivo de esta área de proceso es “identificar las causas de los defectos y de otros problemas y establecer acciones correctivas para resolverlos”.

Esta área de proceso tiene dos objetivos específicos, “Determinar las causas de los defectos” e “Implementar Acciones para corregirlos”. Es en esta área de proceso que el modelo termina de cerrar el ciclo de mejora de procesos propuesto por Demming (PDCA) dentro del modelo CMMI. La tabla presenta los objetivos específicos de esta área de proceso.

SG1 Determinar las causas de los defectos	
SP 1.1	Seleccionar Datos de Defectos para el análisis
SP 1.2	Analizar las causas
SG2 Tratar las causas de los defectos	
SP 2.1	Implementar acciones propuestas

SP 2.2	Evaluar los efectos del cambio
SP 2.3	Registrar los datos

En la sección Análisis causal y clasificación de defectos, se estudiará la aplicación y las herramientas utilizadas por la industria ya sea dentro o fuera del contexto CMMI.

2.2.3.1.3 Área de proceso de Aseguramiento de Calidad de Proceso y Producto

El objetivo del área de proceso de Aseguramiento de Calidad de Proceso y Producto (PPQA) es proveer al staff técnico y a la gerencia con información objetiva sobre el estado de los procesos. El área de proceso PPQA se encuentra a nivel de madurez 2 del modelo CMMI.

El área de Proceso de PPQA tiene dos objetivos específicos. El primer objetivo usualmente se centra en establecer y mantener las capacidades de la organización de efectuar auditorías. Una auditoría es una verificación objetiva de los procesos y/o productos en función de un estándar establecido. En las auditorías de producto y proceso, el objetivo se encuentra en encontrar desviaciones, que se denominan “No Conformidades”. Las desviaciones encontradas en los procesos deberán ser reportadas, y gestionadas para asegurar que los proyectos sean ejecutados según los procesos de la organización. La Tabla 4 muestra los objetivos específicos de esta área de proceso.

SG1	Evaluar objetivamente los procesos y los productos
SP 1.1	Evaluar objetivamente los procesos
SP 1.2	Evaluar objetivamente los productos y los servicios
SG2	Proporcionar ideas objetivas
SP 2.1	Comunicar y asegurar la resolución de los no cumplimientos
SP 2.2	Establecer registros

Tabla 4 Área de Proceso de PPQA

El área de Proceso de PPQA también está relacionada con dos prácticas genéricas. El objetivo genérico 2.9 indica que un proceso definido debe ser auditado para asegurar que esté siendo ejecutado según su definición. Y, según la implementación en cada organización, con la práctica genérica 2.10, que requiere que la alta gerencia este informada sobre el estado de los procesos. Las actividades de PPQA pueden brindar la información necesaria para ofrecer esta visibilidad.

El hecho que el modelo incluya el requerimiento de auditoría desde el nivel de capacidad 2, intenta asegurar que los objetivos que se van consiguiendo a medida que los procesos evolucionan no se pierdan. En [88] se presenta una visión de que PPQA es como la red que evita que la organización caiga en su nivel de madurez. Cuanto más ajustadas sean las auditorías de PPQA a los procesos, más confianza tendrá la organización en que sus procesos están siendo ejecutados según están definidos.

2.2.3.1.4 Área de proceso de Entrenamiento Organizacional

El área de proceso de entrenamiento organizacional (OT) es un área del grupo Organizacional, y se encuentra a nivel de madurez 3 en el modelo. Como se menciona en la sección anterior, es en este nivel de madurez donde el concepto de organización toma relevancia. El objetivo de esta organización es institucionalizar sus procesos a través sus diferentes proyectos.

El área de procesos de entrenamiento organizacional cuenta con 2 objetivos específicos. El foco del primero de ellos es alinear los objetivos de entrenamiento con los objetivos de negocio de la organización, estableciendo de esta forma las “Capacidades de entrenamiento” de la organización. Usualmente, esto se traduce en tener un mecanismo para identificar las necesidades de entrenamiento desde la alta gerencia y en establecer un departamento de entrenamiento organizacional que se encargue de mantener registros de los cursos, de los participantes, y de los tipos de entrenamiento que la organización puede disponer.

El segundo objetivo se centra en realizar los entrenamientos planeados y en evaluar su efectividad. Con respecto a la evaluación de la efectividad del entrenamiento, CMMI no es muy directivo en cuanto a las posibilidades que existen para realizar evaluaciones de entrenamiento. Simplemente se limita a decir que “debe existir un proceso para determinar la efectividad del entrenamiento”[9]. Leyendo un poco más entre líneas, se infiere que los registros de entrenamiento y las medidas definidas en el plan de métricas de la organización debieran contribuir a decir si el entrenamiento ha sido efectivo en cubrir los objetivos de entrenamiento definidos en los procesos asociados con el primer objetivo específico. Sin embargo, el modelo no es directivo en este sentido.

Estos objetivos específicos se presentan en la tabla

SG1	Establecer la Capacidad en Formación en la Organización
SP1.1	Establecer las necesidades estratégicas de formación
SP1.2	Determinar qué necesidades formativas son responsabilidad de la Organización
SP1.3	Establecer un Plan Táctico de formación en la organización
SP1.4	Establecer las posibilidades / capacidades de formación

SG2 Proporcionar la formación necesaria	
SP2.1	Proporcionar formación
SP2.2	Establecer registros de formación
SP 2.3	Evaluar la efectividad de la formación

Tabla 5 Área de Proceso de Entrenamiento Organizacional

El área de Entrenamiento Organizacional, no es el único lugar del modelo CMMI donde pueden encontrarse referencias a la necesidad y la importancia del entrenamiento para institucionalizar los procesos.

- GP 2.5. Para que un proceso pueda operar a nivel de capacidad 2, este proceso tiene que tener resuelto cómo va a entrenarse a los practicantes de ese proceso en las habilidades y conocimientos necesarios para ejecutarlo. La práctica genérica 2.5, es denominada Entrenar a las personas, y su objetivo es que el entrenamiento apropiado sea provisto a las personas que van a ejecutar los procesos. Esta sub práctica tiene la particularidad, en comparación con las otras 9 sub prácticas dentro del objetivo genérico 2, de que es la única, quizás junto con la GP 2.1 *Definir Políticas*, que no puede ser asociada con un área de proceso de nivel de madurez 2. Esto es para complementar la necesidad de establecer las capacidades de entrenamiento desde el comienzo de las iniciativas de mejora de proceso. El entrenamiento es útil para oficiar de palanca y hacer que las personas cambien sus hábitos de trabajo.
- PP 2.5 En la sub práctica específica 2.5 del área de proceso de Planificación de Proyecto. Esta sub práctica está enmarcada dentro del objetivo específico 2 “Desarrollar el Plan de Proyecto”, y específicamente dice que una vez que se cuenta con los recursos humanos del proyecto identificados (PP SP 2.4) es necesario evaluar si es necesario complementar las habilidades y conocimientos de los recursos del proyecto. Lo interesante de esta sub práctica es que el modelo no tiene guías de cómo debe hacerse para conseguir identificar el GAP entre los conocimientos disponibles y los requeridos.

En este trabajo se propone un proceso que interpretará los defectos de los proyectos como las necesidades de capacitación de los mismos.

2.2.3.2 Críticas al modelo de Capacidad y Madurez Integrado

Las principales críticas al modelo CMMI vienen del lado de los autores que argumentan que el modelo requiere un grado de formalidad, y documentación en los procesos de la organización que adopta al modelo, y que esta explosión documental genera una burocracia que no beneficia a las organizaciones[89, 90]. Aunque no sea la intención del modelo requerir documentación para que las organizaciones cumplan con sus prácticas, las malas interpretaciones[89], tanto del modelo como del método de evaluación[91]

hacen que las implementaciones tiendan a generar más documentación de la estrictamente necesaria. De todas maneras en los últimos años se ha visto un esfuerzo de los investigadores y practicantes para amalgamar las metodologías ágiles con el modelo CMMI[90, 92, 93, 94]. El mensaje de estos trabajos es que los mejores resultados se obtienen cuando las organizaciones consiguen aplicar la disciplina del modelo CMMI con la flexibilidad de las metodologías ágiles de desarrollo.

Más allá de estos argumentos filosóficos en cuanto a la interpretación del modelo CMMI y de si éste es apto o no para ser adaptado a las metodologías ágiles de desarrollo, el modelo ha sido criticado en cuanto a que su implementación es cara para las organizaciones de software[95, 96]. Esto es especialmente crítico en las pequeñas y medianas empresas[97, 98] donde los recursos son aún más escasos.

El modelo también es criticado en cuanto a su estructura. A pesar de que una de las intenciones explícitas de los autores en el modelo es que éste debe ser un conjunto de buenas y efectivas prácticas, existen autores[98] que argumentan que la falta de una metodología concreta para adoptar CMMI hace que el conocimiento concretado en el modelo no se esté replicando en las distintas organizaciones.

Quizá es por esto que la mayor crítica con respecto al modelo CMMI viene a partir de su falta de foco en la mejora continua demostrable a través de datos tangibles. En [99] se argumenta que las principales razones por las que no es posible asociar los resultados de evaluaciones de alta madurez con alta productividad son inherentes a la forma en que el modelo está construido, y especialmente cómo el método estándar de evaluación (SCAMPI [100]) selecciona los proyectos de una organización para ser evaluados. En [89] se realiza una muy buena crítica, no con respecto a las intenciones del modelo sino a las maneras “comunes” en que éste ha sido interpretado. Se argumenta que el modelo ha sido adoptado por la industria convirtiéndolo casi en un estándar de facto. Pero que este crecimiento no ha sido acompañado por el SEI en la oferta de capacitaciones para conseguir una homogenización en las interpretaciones. Si a esto se suma las lagunas en los métodos de evaluación, el resultado es un gran número de evaluaciones de dudoso potencial para predecir resultados futuros de performance. Quizás podría ser ésta la explicación de por qué en 2001 una encuesta realizada por Software Industry Benchmarking 2001 publicó resultados tan bajos como que sólo el 21% de las organizaciones involucradas con iniciativas de mejora de procesos consiguen resultados tangibles[101].

2.2.4 Reflexiones respecto a los modelos de Mejora de procesos

En las secciones anteriores se presentaron tres modelos de mejora de procesos. Y se dejó en evidencia que existe una convicción de que la mejora de procesos puede ayudar a obtener mejores productos de mejor calidad y con mayor productividad.

Es nuestra opinión que más allá de las críticas a los modelos presentados, existe una falta de certeza en cuanto a la efectividad del enfoque en mejora de procesos. Tanto nuestra investigación como nuestra experiencia en la industria del software nos incitan a pensar que no es tanto un problema del modelo que se seleccione, sino de las interpretaciones

que se les han dado. Más allá de las críticas que hemos presentado en función de su estructura y contenido, consideramos cierta la frase acreditada a George Box “Todos los modelos están equivocados, pero algunos pueden ser de utilidad”.

Creemos que en la práctica, la selección del modelo de mejora de procesos a ser adoptado por la empresa debe ser resultado de la evaluación de los objetivos de negocio, debiendo cada uno seleccionar el que tenga sentido en el contexto de negocio y del entorno de la organización.

A modo de resumen de esta sección, en la tabla siguiente presentamos las grandes diferencias entre los modelos estudiados. En este trabajo basaremos nuestras premisas en el modelo CMMI. La decisión se basa en función de que lo consideramos el modelo más completo en términos de vocabulario y material de apoyo, de que fue diseñado explícitamente para ayudar a las organizaciones de software a mejorar sus procesos y de que cuenta con más de 15 años de experiencia y evolución.

Modelo	Distribución Libre	Adoptado por la Industria	Orientado a Software	Foco en análisis causal	Foco en Entrenamiento
ISO 9000	No (versiones Draft disponibles)	Alto	No	Medio	Medio
ISO 15504	No (versiones Draft disponibles)	Medio/Bajo	Si	Bajo	Medio
Seis Sigma	No	Medio	No	Alto	Alta
CMMI	Si	Si	Si	Bajo	Medio

Tabla 6 Comparación de los modelos de mejora de procesos

2.3 Mecanismos de Resolución

En esta sección se agrupan las líneas de investigación complementarias a los dos cuerpos principales. Los autores y grupos de conocimientos relevados en esta sección han sido utilizados como base para establecer la resolución del problema o para evaluar la efectividad de la propuesta de tesis.

2.3.1 Procesos y formatos para escribir procesos

El primer punto dentro de este capítulo releva las alternativas disponibles para documentar un proceso productivo. En [102] se presenta una revisión de las formas de

modelar un proceso. En esta sección se presentan algunas de las alternativas disponibles para documentar un proceso de producción de software.

En primer lugar, con respecto a CMMI es un modelo que no es directivo, no hay una parte del modelo donde se diga cuáles son las características o el formato de un proceso productivo, sin embargo sí define las características que un proceso debiera tener (ver sección 2.2.3). El área de procesos de que se ocupa de establecer la capacidad de la organización de definir procesos es OPD (por las siglas en inglés de foco en la Definición de Procesos Organizacionales). En esta área de proceso en el objetivo específico (SG 1, SP 1.1)[103] se describe que se deben establecer y mantener las definiciones de procesos que son necesarios para soportar los procesos productivos de la organización. También se describe que debe establecerse un estándar para la definición de procesos. En cuanto a las características que el proceso debe cumplir, la definición depende del nivel de capacidad de este proceso. Por ejemplo a nivel de capacidad 3, el proceso debe ser definido. Y según el glosario de CMMI un proceso definido es aquel que “es customizado a partir del conjunto de procesos estándar de la organización, cuya descripción está siendo mantenida, que contribuye con información de mejora a la organización. Este proceso es ejecutado de acuerdo a una política organizacional por recursos entrenados, es monitoreado, controlado y su ejecución es objetivamente evaluada”[9].

Este nivel de abstracción de la definición de lo que es un proceso, hace que muchas organizaciones tengan que salir a buscar guías para poder escribir un proceso.

Para continuar en la escuela del SEI, Humphrey en su Proceso de Desarrollo Personal[17] (PSP), presenta los procesos que los desarrolladores deben seguir. El formato seleccionado por Humphrey es de una tabla con las columnas: Paso, nombre, y actividades, y donde la primera fila representa los criterios de entrada y la última fila representan los criterios de salida.

Además, como herramienta de soporte, en este mismo libro se dedica todo un capítulo a establecer una definición de un proceso. En este capítulo el autor propone un proceso de 8 pasos:

1. Determinar las necesidades y prioridades de procesos, donde un proceso de desarrollo de software se construye para producir productos de software. El proceso personal de desarrollador debe construirse teniendo en cuenta los criterios de calidad y los tipos de productos de software que se desarrollarán con ese proceso.
2. Determinar los Objetivos del proceso y los criterios de calidad. Estos objetivos y criterios deben salir de las necesidades del proceso, sin embargo el autor recomienda comenzar por mapear los criterios de calidad y objetivos del PSP.
3. Caracterizar el proceso actual. Implica establecer una descripción de los objetivos y características que describen al proceso actual de manera de poder identificar los puntos de éste que serán mejorados.
4. Caracterizar el proceso objetivo. ¿Cuál es el objetivo al que se quiere llegar? Se necesita saber cuál es la iniciativa de mejora antes de comenzar a mejorar.

5. Establecer una estrategia para el desarrollo del proceso. Implica establecer el plan de implantación que llevará desde el proceso actual al proceso objetivo.
6. Definir el proceso. Para el PSP el proceso de escritura de proceso es iterativo Top-Down donde se debe comenzar por una descripción de alto nivel del proceso y las tareas, y luego ir refinando cada tarea como un proceso independiente hasta que el nivel de abstracción es “razonable” con las necesidades y conocimiento de cada uno. El formato a utilizar es el de la tabla y columnas ya descrito en esta sección.
7. Validar el proceso. Implica probarlo con ejercicios y de ser posible comparar los resultados contra los datos históricos de cada uno.
8. Mejorar el proceso. Cambiarlo, mejorar o agregarle elementos y puntos de control.

A nivel organizacional, el problema con el formato de definición de proceso es que éste está orientado a los procesos personales de software. Elementos como Roles y Responsabilidades, no están contemplados por que no tienen sentido dentro de su contexto. Este modelo puede ser tomado como base para la definición de procesos, pero debe ser adaptado al contexto de una organización con varios involucrados en la ejecución de un proceso.

Por otro lado, en Six Sigma se trata con procesos de negocio, y la herramienta propuesta para modelar estos procesos de negocio es la llamada SIPOC[79]:

- Suppliers (Proveedores)
- Inputs (Entradas)
- Process (Proceso)
- Outputs (Salidas)
- Customer (Cliente)

Los modelos SIPOC se representan usualmente como diagramas de secuencia con 5 líneas de flotación que representan cada uno de estos elementos. Existe un proceso recomendado para identificar y escribir diagramas de procesos SIPOC. La recomendación es comenzar con entrevistas a los practicantes del proceso para “caracterizar el proceso actual”. Las entrevistas se deben conducir con el objetivo de identificar las salidas, las entradas y el proceso de transformación entre las primeras y las últimas. El diagrama se construye y valida en estas sesiones.

Los modelos presentados hasta este punto, representan modelos de descripción de procesos Funcionales u Organizacionales[102, 104], porque se concentran en encontrar definiciones de las tareas y actividades de los procesos productivos. Los modelos anteriores son también modelos estáticos, ya que no permiten la simulación del comportamiento del proceso[104]. Otra manera de representar a los procesos puede ser mediante modelos dinámicos. Estos modelos dinámicos como el pensamiento sistémico (ver sección 2.3.5) se utilizan para entender el comportamiento y la incidencia de las variables que influyen sobre las salidas de los procesos.

2.3.2 Retorno de Inversión de iniciativas de mejora de procesos en la Industria del Software

El objetivo final de la mejora de procesos basado en el modelo CMMI es conseguir la satisfacción del cliente controlando el número de defectos y los desvíos en las estimaciones en los proyectos de desarrollo de software[10]. Como se presentó en la sección 2.1.3, el Retorno de Inversión es una técnica utilizada para evaluar los resultados de las inversiones en distintas iniciativas. En la sección mencionada los autores lo utilizaban para medir el valor del dinero invertido en iniciativas de entrenamiento. En los trabajos relevados en esta sección, los autores utilizarán el Retorno de Inversión para justificar la inversión en iniciativas de mejora de procesos.

Las dos siguientes sub secciones dividen el estudio en dos grandes grupos de autores. Quienes se concentran en medir la capacidad del proceso, lo hacen respetando una premisa de los modelos de mejora de procesos de software, es que la Calidad de un producto está basada en la calidad del proceso utilizado para crearlo[7]. Así la inversión en mejora de proceso es justificada como mecanismo para lograr procesos más predecibles.

El segundo grupo de autores utiliza medidas de reducción de defectos y costo de defectos observados como entradas para calcular los beneficios en la fórmula de Retorno de Inversión - ver fórmula (1).

2.3.2.1 Técnicas basadas en la medición de la capacidad del proceso

Esta medición se realiza, usualmente, asociada a evaluaciones formales de conformidad. Por ejemplo una evaluación SCAMPI[100] o una certificación ISO determinan si los procesos de una organización cumplen con los estándares requeridos. En el contexto del modelo CMMI, solo una evaluación SCAMPI del tipo A puede dar niveles de madurez a las organizaciones de software.

Por lo tanto, varios autores han estudiado el problema de determinar el nivel de madurez, o la capacidad de los procesos intentando independizarse de las evaluaciones formales.

El principal mecanismo utilizado es el uso de cuestionarios orientados a identificar las prácticas del modelo que se están cumpliendo en los proyectos. Este es el enfoque seguido por [14, 105, 106, 107]. En [106] utiliza un cuestionario a partir de los objetivos de las áreas clave de proceso del modelo CMM. Los resultados de este trabajo soportan el concepto de que aquellos procesos que más estrechamente siguen los objetivos de las áreas clave de proceso de CMM tienden a tener menos defectos. El trabajo realizado por [14] en Intel utiliza formularios de estas características y reporta beneficios en términos de reducción de defectos en los niveles de madurez superiores. Esta es también la estrategia utilizada por el SEI, quienes utilizan también el sistema de encuestas para establecer el nivel de madurez de los encuestados[107].

En estos trabajos, el cálculo del retorno de Inversión se realiza para toda la iniciativa de mejora de procesos, y los beneficios se calculan en función de la observación de mejora en una métrica particular (por ejemplo reducción de defectos, desvío de cronograma, etc). En [108] presentan un modelo para calcular el Retorno de Inversión de implementar mejoras de proceso en distintos puntos del ciclo de vida. Los autores concluyen que la mejora de procesos da resultados positivos, pero que a medida que se llegan a los niveles de madurez superiores la relevancia de las mejoras se va achatando. En ese mismo año, [109] argumenta que calcular el Retorno de Inversión para toda la iniciativa de mejora de procesos abarcaría demasiadas variables que no necesariamente están vinculadas con los resultados de la mejora de procesos de software. Por lo que propone un modelo de estimación basado en COCOMOII[110] para aislar los efectos de la mejora de procesos de otras iniciativas de la organización. Utilizando métricas de esfuerzo, el autor concluye que aumentar un nivel de madurez equivale a una reducción del desvío de esfuerzo de entre un 15%-21%.

Un estudio interesante debido al volumen de datos que manejan es la experiencia reportada en General Dynamics[111]. Los autores cuentan con datos de esfuerzo, retrabajo, productividad y densidad de defectos. En esta organización cada proyecto realiza periódicamente un autoanálisis de su nivel de madurez, basado en cuestionarios, y en función de los resultados de este análisis de madurez, los autores muestran cómo en términos generales las métricas analizadas van mejorando.

2.3.2.1.1 Críticas a las técnicas basadas en la medición de la capacidad del proceso

De los resultados explicados anteriormente, se entiende que basarse únicamente en las mediciones formales de madurez o capacidad limita a los investigadores la disponibilidad de datos para realizar sus conclusiones[106]. En un trabajo ya mencionado[99], el autor critica la validez de los resultados de las evaluaciones formales de CMMI ya que representan una muestra de la organización y no el nivel medio de capacidad de la organización.

Con respecto a los autores cuyo mecanismo para medir la capacidad de los procesos estaba basado en encuestas, el principal problema identificado está en la capacitación y el entendimiento de quienes contestan las encuestas. Estos factores afectan la confiabilidad de la información. En la sección 2.2.3.2 ya se estableció la falta de capacitación y comprensión de las intenciones del modelo CMMI en un número importante de los usuarios del modelo en la industria, por lo que a cada estudio basado en encuestas debiera estar acompañado de evidencias de que esta debilidad está resuelta entre los entrevistados.

Por otro lado, como se establecerá en la sección 6.5.5.3.2, las medidas de conformancia de esta propuesta no se centran en el modelo de referencia, sino en la implementación del área de Proceso de PPQA de la organización objetivo.

Nuestra propuesta no se concentrará en el nivel de madurez de la organización, aunque si se establecerán requerimientos a las capacidades de sus procesos (ver sección 4.1). Por el

contrario, el foco estará en estudiar las desviaciones de los procesos bajo estudio con respecto a los procesos de la fábrica.

2.3.2.2 Técnicas basadas en la reducción de defectos

Esta sección presenta el trabajo de autores que han utilizado técnicas de reducción de defectos para justificar la inversión en iniciativas de mejora de procesos.

En primer lugar Watts Humphrey en su propuesta del modelo de desarrollo de software de equipos (TSP [25]) utiliza la reducción de defectos observada luego de la implantación de TSP como mecanismo para fundamentar su inversión en reducción de defectos.

Más recientemente, [112] realiza un análisis cualitativo del impacto de la reducción de defectos para justificar la inversión en técnicas de testing automático de software como mejora al proceso de Verificación.

Finalmente en el contexto de la mejora de procesos de software, en [113] se aplican modelos genéricos de Retorno de Inversión a distintos programas de Mejora de Procesos de Software. Más concretamente, en [13, 114, 115] se reportan beneficios en términos de costos o de Reducción de Variación tras la implantación de modelos de mejora de procesos.

En [116] el autor aplica la fórmula de ROI interpretando los ahorros en cambios en los procesos como beneficios, aplicando la fórmula tanto para ahorro en costos como para ahorro en cronograma.

Por último, los autores que se concentran en estudiar el Costo de la Calidad [116, 117, 118, 119], utilizan la reducción de defectos para justificar la inversión en mejoras al proceso de control de calidad. En [117, 119] se presentan sendos modelos para calcular el costo de la calidad que tienen en cuenta tanto elementos directos como indirectos, utilizando como información de beneficio la reducción de defectos que observaron luego de haber aplicado su modelo. En [118], para la industria de servicios, se intenta llegar a una fórmula para determinar el punto donde la inversión en calidad supera el costo que el cliente está dispuesto a pagar.

2.3.2.2.1 Críticas a las técnicas basadas en reducción de defectos

Calcular en ROI en función de la reducción observada de defectos parece a priori un acercamiento razonable para calcular la efectividad de las iniciativas de mejora de procesos. La principal crítica que puede aquí hacerse es que para poder transformar beneficios en unidades de moneda es necesario que la organización sea capaz de identificar cuál es el costo de un defecto. De hecho cuando Humphrey realiza el cálculo de beneficio de la implantación de TSP [25] reconoce que éste es el principal problema de su enfoque y el mismo utiliza datos históricos de otros estudios.

En el modelo propuesto en este trabajo, también utilizaremos el mecanismo de justificar el retorno de inversión en entrenamiento a partir de la reducción de defectos observada luego de la implantación del proceso propuesto.

2.3.3 Análisis causal y clasificación de defectos

En la sección 2.2 se estableció que el *análisis causal forma parte de los ciclos de mejora de procesos*[7, 120, 121, 122]. El análisis causal se concentra en entender las relaciones causa-efecto dentro de un sistema con eventos y condiciones que provocaron los resultados observados[123], ya sea negativos o positivos. El objetivo de aplicar análisis causal de defectos es identificar las causas e implementar mejoras que eviten la aparición de estos defectos en el futuro[120].

En la industria de software, el principal uso que se le da a las técnicas de análisis causal está en el estudio de los defectos producidos por los equipos de desarrollo de software. Ya en 1975, [124] luego de realizar un estudio de los defectos incurridos en el desarrollo del sistema operativo MS-DOS, el autor concluye que a pesar de haber realizado ese trabajo en forma post mortem, el análisis de los defectos puede resultar provechoso para mejorar la performance de los proyectos de desarrollo futuros.

En la década de los 80, en IBM se reportan resultados exitosos de aplicar análisis causal de defectos para identificar oportunidades de mejora de procesos[125, 126]. A este ciclo de mejora lo llamaron Orthogonal Defect Classification (ver IBM Clasificación Ortogonal de Defectos) donde los defectos fueron clasificados en varias – en IBM se utilizaban dos - categorías ortogonales para luego ser agrupados y sus causas analizadas. Las conclusiones eran tratadas como oportunidades de mejora de procesos. Este ciclo de mejora fue reproducido en otras organizaciones como por ejemplo en [127].

En todos los casos, los mecanismos de análisis causales utilizados tienen en común el uso de taxonomías para clasificar y agrupar los defectos registrados.

2.3.3.1 Taxonomías de clasificación de defectos

Las taxonomías para realizar la clasificación de defectos son útiles para agrupar defectos a la hora de estudiar las causas y los motivos de la introducción de los mismos en las líneas de producción. Muchas organizaciones de software clasifican los defectos de alguna manera básica, por ejemplo en Alto, Medio, Bajo[128, 129], para priorizar los esfuerzos de estabilización del software. Sin embargo cuando la comprensión de las metodologías de desarrollo se completa y cuando la intención de mejorar los ciclos de desarrollo se hace patente, clasificar los defectos en categorías más elaboradas facilitará el acceso a la información contenida en ellos.

2.3.3.1.1 IBM Clasificación Ortogonal de Defectos

En IBM [125] se utiliza una clasificación ortogonal de defectos para obtener propuestas de mejora de procesos para la organización. En este contexto, los desarrolladores clasifican los defectos en una de las siguientes cuatro categorías.

- *Educación*, los defectos en esta categoría son aquellos en que el desarrollador no entendió algún aspecto del proceso que tenía que ejecutar, o del producto que tenía que desarrollar.
- *Comunicación*, en esta categoría se identifican los defectos en que el desarrollador no tenía la información necesaria para realizar su tarea.

- *Desatención*, en esta categoría se identifican los defectos en que el desarrollador olvidó tener en cuenta todos los casos de borde.
- *Transcripción*, en esta categoría el desarrollador sabía qué tenía que hacer, entendía el problema completamente, pero simplemente cometió un error.

2.3.3.1.2 Otras clasificaciones propuestas

Existen otras taxonomías de clasificación de defectos además de la de IBM. Una de las primeras, y que todavía se mantiene activa es la propuesta realizada por [130]. La taxonomía de Beizer resulta útil para analizar los datos en forma automática, ya que presenta las categorías en forma numérica. En la taxonomía de Beizer las categorías están identificadas por números enteros separados en 1000 unidades, lo que permite la extensión y adaptación de la misma a las necesidades de la organización. Aunque los trabajos de Beizer tienen casi 20 años de antigüedad, muchos de sus resultados se mantienen activos, por ejemplo es utilizada en los formularios del Personal Software Process [27].

Grady[131] presenta un estudio de la taxonomía de clasificación de defectos utilizada en Hewlett-Packard. Al igual que en el caso de IBM, en HP utilizan las categorías para asignar las causas de defectos y de disparadores que representan lo que los desarrolladores piensan que ha generado los defectos.

En [132] los autores presentan un reporte de las distintas clasificaciones utilizadas al momento de la publicación y de cómo se utilizaban. Entre los relevamientos se encuentran los trabajos de IBM y de HP destacados anteriormente.

2.3.3.2 Estudio de la aplicación de técnicas de análisis causal en entornos de baja madurez

Como el área de proceso de análisis causal (CAR) es un área de proceso que pertenece al nivel 5 de madurez del modelo CMMI, no es esperable que una organización nivel de madurez 3 haya implementado esta área de proceso[89]. En [38] nosotros presentamos una revisión sistemática de los autores que han trabajado en técnicas de análisis causal de defectos en busca de experiencias de aplicación en entornos de baja madurez del modelo CMMI. La discusión que se presenta en esta sección provee una actualización a los resultados presentados en esa conferencia.

Las primeras cuatro columnas de la tabla representan los objetivos con que el autor utiliza análisis causal dentro de la organización bajo estudio. Las otras 3 columnas representan las herramientas que los autores utilizan en las sesiones de análisis causal dentro de cada una de las organizaciones. Estos resultados serán relevantes para desplegar el proceso de análisis causal en el momento de la experimentación requerido por el proceso propuesto.

Autor	Objetivo de la aplicación				Herramientas Utilizadas		
	Desplie- gue en Baja Madurez	Mejora de Proce- sos	Reduc- ción de Defectos	Capacita- ción	Diagra- mas Ishikawa[133]	Mindma ps	Herramien- tas Automá- ticas
Card D. [126]	No	Si	Si	No	-	-	-
Bahandari [134]	No	Si	-	No	-	-	-
Card D. [127]	No	No	Si	No	-	-	-
Peled [135]	Si	Si	No	No	Si	No	No
Fredericks [136]	No	Si	No	No	-	-	-
Norman E.F [137]	No	No	No	No	No	No	Si
Lezak M [138]	No	Si	No	No	-	-	-
Jacobs J.C. [139]	No	Si	No	No	No	No	Si
Bibi S. [140]	Si	Si	No	No	No	No	Si
Buglione L Et all [121]	Si	-	No	No	Si	No	No
Matalonga [39]	Si	No	No	Si	Si	No	No
Buglione L [120]	Si	Si	No	No	No	Si	No
Kalinowski et all [122]	-	Si	Si	No	Si	-	Si

Los trabajos más importantes para fortalecer las bases del proceso propuesto son los de [120, 121, 135]; estos tres trabajos representan experiencias de la aplicación de técnicas de análisis causal en organizaciones de bajo nivel de madurez. *Aportan por lo tanto un nivel de validación al proceso propuesto.*

2.3.3.3 Reflexiones sobre las técnicas de análisis causal

El análisis causal es una técnica básica para implementar acciones de mejora de procesos. Aunque para el modelo CMMI el área de proceso relacionada (CAR) está a nivel de madurez 5 (ver sección 2.2.3.1.2), la aplicación de análisis causal no es privativa de las organizaciones de alta madurez.

Los trabajos de [120, 121, 135] demuestran que en organizaciones que guían sus procesos de mejora a partir del modelo CMMI, también es posible desplegar procesos de mejora basados en técnicas de análisis causal. Inclusive en [103] se comenta como el SEI ha evaluado introducir requerimientos de análisis causal en niveles inferiores de madurez.

De todas formas, aunque estos resultados muestren que es posible desplegar técnicas de análisis causal a niveles de madurez inferiores, el análisis causal de defectos tiene que estar basado en un conocimiento cuantitativo del proceso de producción. Y aunque en el modelo CMMI el área de métricas está a nivel de madurez 2, las áreas de gestión cuantitativa de procesos se encuentran a nivel 4.

Por este motivo, en este volumen se han cuidado los términos referidos a la institucionalización de un proceso de análisis causal. Los requerimientos en este trabajo se concentran en establecer que es posible aplicar técnicas de análisis causal en entorno de baja madurez.

Con respecto a las clasificaciones de defectos estudiadas, es posible concluir que el principal uso es para alimentar los ciclos de análisis causal que guiarán la mejora de procesos. En [119] presenta los motivos por los cuales es importante que cada organización desarrolle su taxonomía de clasificación de acuerdo a sus necesidades.

En esta propuesta se aplicarán los resultados de [119] para justificar el esfuerzo que se requiere en construir una taxonomía de clasificación que sea propia de la organización (ver sección 6.2.1).

2.3.4 Gestión cuantitativa de procesos

La gestión cuantitativa de procesos implica aceptar que el proceso de producción puede ser entendido en función de sus salidas. En el modelo CMMI, las áreas de proceso que exponen requerimientos cuantitativos de proceso se encuentran a nivel de madurez 4. En [141], se describe que las recomendaciones del modelo CMMI con respecto a la gestión cuantitativa de procesos tiene su origen principal en los trabajos de Shewarth [71], quien trabajando para los Laboratorios Bell presentó las técnicas fundamentales para la gestión cuantitativa o estadística (SPC) de un sistema de producción. Los métodos de Shewarth fueron popularizados por Deming[7].

Actualmente, los métodos de control estadístico son aceptados en las industrias manufactureras, pero son de poca penetración en la industria del software[142]. Una encuesta realizada por el SEI [143], las organizaciones con evaluaciones de alta madurez solo representan el 10% del total de organizaciones con evaluaciones CMMI. Además, un estudio reciente muestra que la mayoría de estas organizaciones no están realmente aplicando las técnicas de SPC correctamente[144].

Los gráficos de control representan las salidas del proceso en una serie temporal. Son útiles para diferenciar la variación natural de la variación especial. Wheeler[145] explica que la variación natural, también llamada “la voz del proceso” es la que podemos esperar de cualquier proceso de producción y responde a causas intrínsecas al proceso. La variación excepcional es la que puede asignarse a causas excepcionales.

En un proceso predecible, las variaciones excepcionales deben ser estudiadas, idealmente mediante técnicas de análisis causal, para encontrar y eliminar sus causas. Se define que un proceso está “en control” cuando las salidas del proceso no exhiben comportamiento de variación especial. Una característica utilizada de los procesos “en control” es que el comportamiento pasado puede utilizarse para predecir comportamiento futuro[146].

Para construir gráficos de control se necesitan identificar el tipo de datos y los límites de control del gráfico. Del tipo de datos dependerá el tipo de gráfico de control a utilizar ([146] describe los criterios de decisión por tipo de datos para seleccionar el gráfico de control adecuado).

Tradicionalmente, Shewarth [71], ubicaba los límites de control a tres desviaciones estándares de la media. Wheeler[145] introduce los gráficos de variación (moving range chart), estos gráficos describen la variación entre dos puntos consecutivos de la serie, permitiendo identificar causas especiales de variación sin necesidad de conocer todas las heurísticas presentadas por Shewarth.

2.3.4.1 Procesos Predecibles

Como mencionamos se dice que un proceso es predecible cuando su variación está dentro de los límites naturales del proceso cada vez que éste es ejecutado[145]. La ventaja de contar con un proceso predecible es que se pueden fijar expectativas a sus salidas. Si éstas no se ajustan a los requerimientos de negocio para ese proceso, se deberá cambiar el proceso, o revisar los objetivos de negocio.

En la sección anterior se presentó que la identificación de variación natural y especial está determinada por los límites de control del gráfico de control del proceso, y por los límites del gráfico de variación.

Es de destacar que tanto los límites de tres sigmas de Shewarth (junto con sus heurísticas) como los criterios de Wheeler para discriminar causas naturales, son justamente heurísticas. Como el objetivo es identificar problemas para asignar causas y soluciones, distintos autores han presentado sus heurísticas para su determinación. El problema está en el costo de buscar explicaciones cuando la variación es natural. Deming lo llama “un ejercicio de inutilidad”. Recientemente, en software se han criticado los criterios utilizados para establecer estos límites[147], donde los autores proponen un modelo de costos para

establecer los límites de control, que en su opinión se ajusta más al proceso de desarrollo de software.

En nuestra opinión, el tema se resuelve a un balance del costo beneficio. La propuesta de Shewarth, a tres sigma, es lo suficientemente sencilla de calcular, y a él le resulta efectiva. La propuesta de Wheeler, aunque un poco más complicada en su matemática, es efectiva porque evita el conocimiento de las heurísticas de Shewart. En este trabajo utilizaremos los criterios para construir gráficas de control seguidos por Wheeler y Florac. Nuestra propuesta utilizará los gráficos de control para determinar la línea base de productividad y luego de transcurrida la intervención de entrenamiento establecer una medida de retorno de inversión basada en la observación de los desplazamientos de los límites de control.

2.3.5 Pensamiento Sistémico

El ciclo básico de mejora de procesos (ver sección 2.2) está íntimamente asociado a la gestión cuantitativa de proceso vista en la sección anterior. En un entorno ideal, se evaluarían los puntos del proceso que exhiben causas naturales de variación, se realizaría el análisis causal para encontrar los motivos y se modificaría el proceso.

En la práctica, se evidencia que existen otros factores que pueden afectar la producción de un proceso de software. Las herramientas de pensamiento sistémico [148, 149] permiten entender el comportamiento dinámico de un sistema. El Pensamiento sistémico está compuesto por los siguientes conceptos interrelacionados[149]:

- Entender que la causa y el efecto pueden ser recíprocos. El pensamiento sistémico incorpora el pensamiento circular para permitir la identificación de círculos virtuosos o viciosos.
- El sistema como la causa. El comportamiento, y por lo tanto la performance, dependen principalmente de sistema. Entender al sistema como la causa del problema permite concentrarse en mejorarlo.
- Pensamiento dinámico sustituye al pensamiento estadístico. Las relaciones entre las variables se entienden en función de su evolución en el tiempo.

Para ilustrar el pensamiento sistémico usualmente se recurre a la siguiente analogía. Desde la educación inicial está grabado el concepto de causa-efecto como parte intrínseca de nuestro aprendizaje. Los chicos aprenden a base de prueba y error. Y mientras más chica sea la temporalidad entre la causa y el efecto, mejor preparados están para asimilar este aprendizaje. Un ejemplo claro de esto es como un recién nacido “aprende” que cuando tira su juguete (causa), mamá está disponible para levantarlo, devolvérselo y hacer alguna monería con él (efecto). Momento en que el vuelve a tirarlo. Lentamente el bebé aprende que para llamar la atención de mamá alcanza con tirar el juguete. Lo que el niño no puede ver, es como por cada iteración mamá se empieza a aburrir del juego hasta que en un momento “sin previo aviso” lo termina [150]. Este es un ejemplo de un sistema con un flujo de “feedback”, la acción del niño tiene un efecto no deseado (el aburrimiento de la madre), que eventualmente llegará a un cierto límite que romperá con la relación causa-efecto que él había aprendido.

El problema radica en que las relaciones causa efecto de las organizaciones donde trabajamos, tienen temporalidades que nos inhiben la asociación de la causa con el efecto. Y para entender las influencias de las variables de estos sistemas, se tiende a descomponer el problema para intentar entenderlo, perdiendo así la visión del conjunto.

2.3.5.1 Pensamiento sistémico en la industria del software

Los procesos de desarrollo son buenos lugares para aplicar el pensamiento sistémico ya que presentan entornos volátiles y no lineales [150, 151].

En la industria del software, el primer trabajo que encontramos en la aplicación de pensamiento sistémico es el de Abdel-Hamid[152]. Sus esfuerzos se basan en obtener sistemas para la toma de decisiones en los proyectos. El libro presenta modelos de sistemas cuyo objetivo es enseñar a los gerentes de proyecto las consecuencias cíclicas de sus acciones. Algunos de estos modelos fueron luego tomados por Pfahl, para desarrollar modelos de simulación con el objetivo de entrenar a jefes de proyecto[153]. En ese trabajo el autor encontró que aquellos jefes de proyecto que consiguen desarrollar habilidades de pensamiento sistémico tienden a reaccionar proactivamente ante los cambios en sus proyectos.

Más recientemente, los trabajos de Madachy [149] se concentraron en las relaciones dinámicas entre las diferentes etapas de producción de un producto de software. Su objetivo expreso es proveer a los recursos involucrados en la producción de software de modelos de pensamiento sistémico que les permitan entender el comportamiento de sus proyectos.

En el campo de la ingeniería de software experimental, los modelos de simulación, y entre ellos los modelos de simulación basados en el pensamiento sistémico, han sido propuestos como una herramienta de bajo costo para realizar experiencias sobre métodos y teorías asociadas a la producción de software[154, 155].

2.3.5.2 Elementos de modelado de un sistema dinámico

Los modelos de pensamiento sistémico son herramientas que se utilizan para asistirnos en la comprensión de sistemas causales complejos. La intención que llevamos al construir estos modelos no es obtener simulaciones exactas de la realidad, sino componer un modelo que nos permita realizar inferencias con respecto al comportamiento del sistema bajo estudio[156].

En la experimentación (sección 6.6) se presentará un modelo de pensamiento sistémico que será utilizado para comprender los resultados obtenidos en la experimentación. En esta sección se presentan los átomos de construcción que son utilizados para construir un modelo sistémico.

- Stocks; son los elementos fundamentales de agrupación. En estos elementos las unidades del flujo del sistema se van coleccionando. Aunque el software permite varios tipos de Stocks, el de uso más común es el Reservorio. Los reservorios se utilizan cuando se quiere acumular las unidades de flujo del sistema y no importa

el orden en que las unidades salen del reservorio. En Ithink, el software de pensamiento sistémico utilizado, los Reservorios se representan por un cuadrado.



Figura 7 Ejemplo de reservorio

- Flujos; representan las comunicaciones entre los reservorios. Los flujos indican cómo las unidades se mueven en el sistema. En el software utilizado son representados como una llave de paso. Por convención los flujos se nombran como acciones. Cuando un flujo apunta hacia un Stock, se dice que el flujo carga o llena el stock. La Figura 8 representa un flujo que carga al stock. El ejemplo de esta figura podría leerse de la siguiente manera: “Codificar (flujo) inyecta Defectos en el Código (Stock)”.

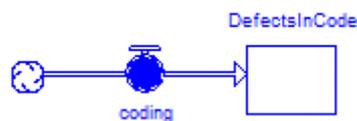


Figura 8 Ejemplo de flujo

- Conectores; los conectores controlan los ciclos en los que las unidades se mueven en el sistema, y se representan como una flecha. Los conectores pueden representar las relaciones causales entre los distintos Stocks y Flujos del Sistema. En la Figura 9 se muestra un “conector de acción”, estos conectores completan el movimiento del flujo en el sistema. El ejemplo de la Figura 9 se lee: “La acción de retrabajo implica codificación, que tiende a generar nuevos defectos en el código”. Como se ve en este ejemplo, los modelos de pensamiento sistémico, simplifican el lenguaje utilizado para describir ciclos causales. La oración utilizada para describir la Figura 9 es más complicada que la conjunción de estas 4 formas.

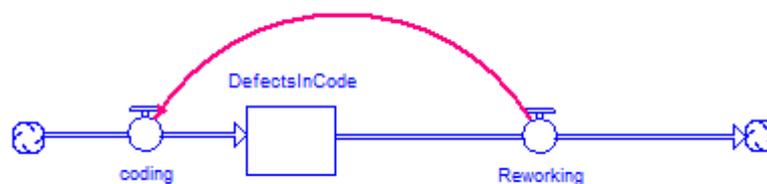


Figura 9 Ejemplo de Conector

- Convertidores; Son utilizados para inicializar numéricamente el sistema. Los convertidores pueden utilizarse para sembrar el sistema a partir de los datos históricos observados.

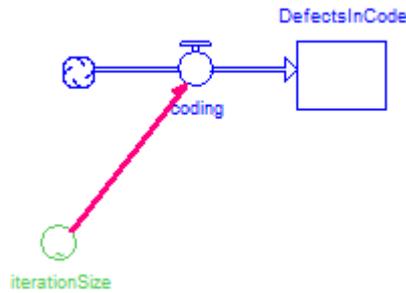


Figura 10 Ejemplo de Convertidor

Sin entrar en mayores detalles con respecto a los tipos de Convertidores que el software provee, en esta propuesta los convertidores serán utilizados para asegurar la coherencia de unidades del modelo. Por eso, el modelo de la Figura 10 tiene que ser completado por el convertidor (“DefectIntroductionRate” ratio de inyección de defectos) en la Figura 11. En el ejemplo de la Figura 11 las unidades están representadas por:

- DefectsInCode: Unidades (1 unidad representa un defecto)
- IterationSize: Hora
- DefectIntroductionRate: Defectos/Hora

De esta forma se asegura la consistencia de unidades entre los convertidores y el reservorio, y las unidades del flujo quedan determinadas por la ecuación:

$$1.2 \text{ Hora} * \frac{\text{Defecto}}{\text{Hora}} = \text{Defecto}$$

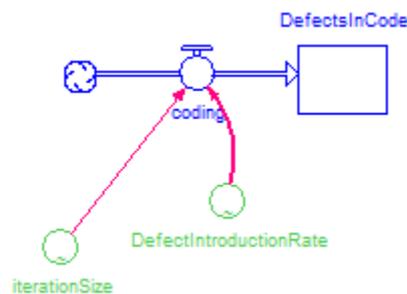


Figura 11 Consistencia de Unidades

Asegurar la consistencia de unidades es útil para establecer una validación primaria del sistema, y aumentar la confianza en los resultados de la simulación. Y es parte del proceso recomendado por los autores del software para construir modelos de pensamiento sistémico.

2.3.5.3 Proceso de modelado de un sistema dinámico

No parece existir una metodología aceptada para el diseño y construcción de modelos de simulación de un sistema dinámico. Madachy [149], recomienda un proceso iterativo para determinar el alcance y los elementos del sistema. Al tiempo que Pfhal[151] incorpora el

concepto de comportamiento estadístico. Esto es, establecer los valores iniciales de las variables para asegurar que el sistema se comporte de forma que reproduzca el comportamiento de la realidad bajo estudio. En esta sección utilizaremos el proceso seguido por el fabricante del software de simulación que estaremos utilizando [156].

A la hora de construir modelos de pensamiento sistémico, una de las primeras decisiones que hay que tomar está en establecer el nivel de abstracción del sistema. Según Forrester[157], el sistema de una persona puede ser el sub sistema de otra. En el caso del pensamiento sistémico, el problema está en conseguir el balance entre el nivel del detalle del sistema y la complejidad de construirlo en una herramienta de análisis. En general se recomienda que los sistemas sean abarcativos (Breadth) y generales (Shallow), sin que estas condiciones dejen de respetar las formalidades requeridas para validar las simulaciones que se pueden conseguir con el software[149, 156].

El modelo propuesto para escribir un proceso es un esquema en V. El objetivo en la base de la V está en identificar el sistema a modelar/entender, y en los niveles sucesivos se va creando el modelo del sistema[156].

Definir el sistema a Modelar	
1) Desarrollar la hipótesis	1) Identificar la audiencia Objetivo
2) Probar la Hipótesis	2) Definir los objetivos de aprendizaje
3) Extraer Conclusiones	3) Definir la estrategia de aprendizaje
4) Validar conclusiones	4) Implementar la estrategia de aprendizaje

Tabla 7 Proceso de construcción de un modelo de sistemas

En el primer nivel se crean las bases del modelo que se va a estudiar. Ambas tareas se realizan en paralelo. Por un lado se tiene que identificar la audiencia para quien se va a construir este modelo, al tiempo que la definición de la hipótesis debe complementar la definición del sistema a modelar. La hipótesis representa el conocimiento actual del comportamiento del sistema.

En el siguiente nivel se refinan los objetivos de aprendizaje. En este nivel se comienza a construir e identificar los Stocks, y flujos que incluirá el modelo. Los objetivos de aprendizaje se irán refinando e irán guiando qué entidades estarán dentro y fuera del sistema. Si se cuenta con datos históricos para modelar el sistema, se deben comenzar a incluir en el modelo los convertidores que representen estas entradas.

La validación del sistema se realiza ajustando el comportamiento observado del sistema con los datos históricos. Los autores [156] recomiendan aquí ajustar un solo convertidor a la vez para conseguir un “sistema estable”, es decir, un modelo de sistema cuyo comportamiento sea similar a la realidad observada.

Los últimos dos niveles representan los niveles de aprendizaje, el modelo permitirá modificar los valores de los convertidores para evaluar cómo el sistema se comporta frente a diferentes presiones de las variables que en él influyen. La estrategia de

aprendizaje en los modelos sistémicos de IThink es siempre la misma “aprender mediante el manejo dinámico de las variables”[156].

La última etapa representa el reconocimiento de los autores de este proceso de la frase ya citada de George Box (“Todos los modelos están equivocados pero algunos son útiles” – ver sección 2.2.4), es decir evaluar bajo qué circunstancias el modelo creado no representa la realidad modelada. Esto requiere de realizar un análisis honesto de las limitaciones del modelo creado.

2.4 Resumen del estado del arte

En este capítulo hemos descrito el estado del arte de las tres áreas de conocimiento en las que se basa nuestra propuesta de tesis.

En primer lugar, la sección 2.1, mostró cómo el problema de evaluar el entrenamiento impartido a los recursos de la organización es un problema abierto en cualquier tipo de organización. El modelo de 4 niveles aparece como el referente en la literatura de evaluación del entrenamiento. Phillips construye su modelo del retorno de Inversión en entrenamiento basándose fuertemente en el modelo de Kirkpatrick. Pero además, Phillips destaca que es necesario proveer métodos cuantitativos y técnicas para que las organizaciones puedan alcanzar los niveles de cálculos de ROI que él propone. Para nosotros, una debilidad que tiene esta formulación del Retorno de Inversión es que no da cuenta de la temporalidad en que se realiza la inversión. ¿Si un banco ofreciera intereses del 10% anual lo elijaríamos sobre un competidor que ofrece un 5% mensual? Los resultados del retorno de inversión deben estar acompañados por la explicación de la temporalidad en que se realizaron los cálculos. En la sección 6.5.5.3, se presenta el cálculo de ROI resultante de la experimentación estableciendo los rangos de fecha que fueron tomados como base para el cálculo de datos de Costos y Beneficios.

En la sección 2.2 presentamos los modelos de mejora de procesos más adoptados por la industria de Software. La sección 2.2.4, rescata el modelo de Capacidad y Madurez integrado como el modelo de referencia para este trabajo. El modelo CMMI cumple con los requerimientos para ser a la vez un modelo educativo, en relación al material que el volumen brinda, y un modelo aceptado por la industria como referencia para evaluar la madurez de las organizaciones productoras de software.

Finalmente, en la sección 2.3 presentamos los mecanismos que serán utilizados para resolver el planteamiento de esta tesis. El foco de estos está en la aplicación de técnicas de análisis causal y pensamiento sistémico al problema del control cuantitativo de procesos. Las secciones Análisis causal y clasificación de defectos 2.3.3 y 2.3.5 presentan dos técnicas distintas para establecer el comportamiento esperado de los procesos.

La sección 2.3.1, se concentra en el estudio de la aplicación de cálculos de Retorno de Inversión a las iniciativas de mejora de procesos. Esta sección valida la originalidad de nuestro aporte, y al mismo tiempo valida la adaptación de las técnicas de Retorno de Inversión en entrenamiento a las iniciativas de mejora de procesos.

3 Planteamiento

En el capítulo anterior se han resaltado en *itálicas* las principales ideas que han llevado a esta propuesta. Este capítulo comienza por detallar y revisar estas 5 ideas para luego presentar la construcción de la propuesta de este trabajo de investigación.

1. *“La inversión en entrenamiento, medida en dólares constantes, ha aumentado en los últimos años”*. Este aumento llevará a que *“los gestores del entrenamiento cada vez estarán más presionados por demostrar cómo la inversión en su sector es redituable para su organización”*.

En la sección 2.1 se presentó el resultado del relevamiento de la inversión en entrenamiento por parte de diferentes fuentes. En esta sección se desprende que todavía es un problema abierto el obtener métodos para que las organizaciones puedan justificar la inversión que realizan en el entrenamiento de sus recursos. Este es además, un problema que pertenece a las organizaciones en general, y no solamente a las organizaciones de software.

2. *“La inversión en entrenamiento está justificada por la premisa de que si los recursos humanos se encuentran mejor entrenados, estos van a producir un producto con menos defectos y de mayor calidad”*. Por otro lado, en la Tabla 6, se mostró que todos los modelos de mejora de proceso estudiados reivindican al entrenamiento como elemento para potenciar el cambio, y la mejora de proceso.

3. En el área de evaluación del entrenamiento organizacional, *“el estado del arte está bastante más desarrollado que el estado de la práctica”*.

Se presentaron los modelos principales de referencia de evaluación de entrenamiento, ver secciones 2.1.2, 2.1.3 y 2.1.4. La oración extraída en este punto destaca un punto en común de estos tres modelos: no presentan una metodología para que los practicantes en evaluación del entrenamiento reproduzcan en sus entornos. Los tres modelos principales no han presentado ejemplos de organizaciones que hayan consistentemente reproducido los modelos.

4. Con respecto al modelo CMMI, este *“ha sido adoptado por la industria convirtiéndolo casi en un estándar de facto”*

En la sección 2.2.3 se presentó un resumen de la organización del modelo CMMI y se identificaron las referencias al entrenamiento de los integrantes de las organizaciones productoras de software dentro del modelo.

El modelo CMMI no es *“directivo (en lo que a sus expectativas de proceso se refiere)”*, porque *“los mecanismos de cómo realizar los requerimientos del modelo son dejados para los practicantes.”*

El entrenamiento en CMMI está presente desde los primeros niveles de madurez, y representado fuertemente por el área de proceso que lleva su nombre (sección 2.2.3.1.4). El entrenamiento es para CMMI la palanca mediante la cual conseguir modificar la forma de trabajo de las personas.

5. El *“análisis causal forma parte de los ciclos de mejora de procesos.”*

Sin embargo, no todos los defectos se deben a problemas del proceso.

En la sección 2.3.3 se presentó un detalle de las técnicas y las herramientas publicadas con respecto al análisis causal de defectos. Se presento como la mayoría

de los autores estaban aplicando el análisis causal de defectos para elicitar iniciativas de mejoras de procesos.

Con estas 5 ideas, en esta propuesta se *vinculará al análisis causal de defectos con la capacitación de los recursos*.

Se entiende que existe un tipo de defecto que para removerlo será más eficiente invertir en entrenamiento.

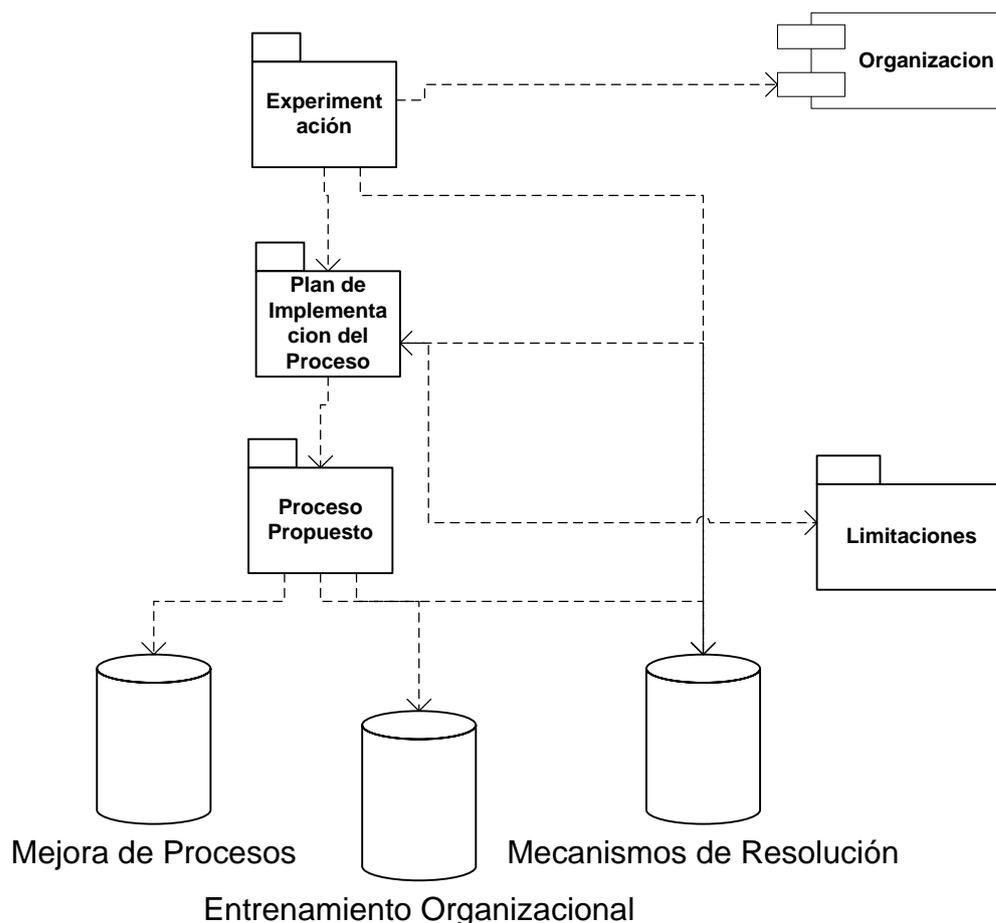


Figura 12 Componentes de la Propuesta

La Figura 12 ilustra los componentes de la propuesta utilizando un diagrama de paquetes UML 2.0[158]. En este diagrama se ve como la capa de “Proceso Propuesto” se construyó en base a los modelos revisados en el Capítulo del Estado del Arte. Las afirmaciones enumeradas anteriormente en esta sección representan los puntos en base a los cuales está edificada la propuesta. Las primeras dos afirmaciones corresponden a la figura de la base de datos “Entrenamiento Organizacional” de la figura. La base de datos “Mejora de Procesos” representa las fuentes que corresponden a las afirmaciones 2 y 3. Finalmente, la última afirmación está representada por la Base de Datos de “Mecanismos de Resolución”.

A partir de estos componentes se ha construido el planteamiento de solución de esta tesis.

El proceso propuesto, representado por el paquete que más dependencia con las bases de datos tiene, consta de dos grandes partes. Por un lado se propondrá una definición de proceso que pueda ser seguida por las organizaciones productoras de software. Por otro lado, el proceso propuesto también incluye en su definición el método para calcular su efectividad a través de la presentación de resultados mediante la utilización de técnicas tomadas de la literatura de Retorno de Inversión en entrenamiento.

Con el objetivo de que este propuesto pueda ser implementado por organizaciones de software, se incluyó en la propuesta un proceso de despliegue. Como se ve en la figura, este proceso de despliegue de procesos asume que la organización se encuentra en las condiciones iniciales que serán descritas en la sección 4.

Finalmente en la experimentación, se analizará el despliegue del plan de implantación, del proceso propuesto y del mecanismo para el cálculo del ROI en una organización productora de software. La experimentación realizada en la fábrica de software será soportada por experimentaciones en laboratorios como se describirá en el capítulo 6.

3.1 Hipótesis de trabajo

Con el objetivo de bajar el nivel de abstracción de este planteamiento para llevarlo a la práctica, el proceso propuesto contestará las siguientes preguntas:

1. ¿Es posible desplegar un proceso de análisis causal en una organización que no ha alcanzado niveles de madurez o capacidad superiores en sus procesos?
2. ¿Es posible establecer un mecanismo reproducible para medir el retorno de inversión de las intervenciones de entrenamiento? (En organizaciones cuya capacidad esté dentro de las Limitaciones ya descritas)
3. ¿El entrenamiento de los recursos tiene impacto sobre los defectos incurridos en los proyectos? Y ¿Existe algún tipo de defectos en los que el entrenamiento tenga mayor impacto?

A modo de resumen, en la siguiente tabla se establece el mapeo entre las afirmaciones y la hipótesis:

Afirmación	Hipótesis
<p>Afirmación 1: “La inversión en entrenamiento, medida en dólares constantes, ha aumentado en los últimos años” y “los gestores del entrenamiento cada vez estarán más presionados por demostrar cómo la inversión en su sector es redituable para su organización”</p>	<p>Hipótesis 2: ¿Es posible establecer un mecanismo reproducible para medir el retorno de inversión de las intervenciones de entrenamiento? (En organizaciones cuya capacidad esté dentro de las Limitaciones ya descritas)</p>
<p>Afirmación 2: “el estado del arte está bastante más desarrollado que el estado de la práctica”</p>	
<p>Afirmación 3: “La inversión en entrenamiento está justificada por la premisa de que si los recursos humanos se encuentran mejor entrenados, ellos van a producir un producto con menos defectos y de mayor calidad”</p>	<p>Hipótesis 3: ¿El entrenamiento de los recursos tiene impacto sobre los defectos incurridos en los proyectos? Y ¿Existe algún tipo de defectos en los que el entrenamiento tenga mayor impacto?</p>
<p>Afirmación 4: CMMI, “<i>ha sido adoptado por la industria convirtiéndolo casi en un estándar de facto</i>” y CMMI no es “<i>directivo (en lo que a sus expectativas de proceso se refiere)</i>”, porque “<i>los mecanismos de cómo realizar los requerimientos del modelo son dejados para los practicantes.</i>”</p>	<p>Hipótesis 1: ¿Es posible desplegar un proceso de análisis causal en una organización que no ha alcanzado niveles de madurez o capacidad superiores en sus procesos?</p>
<p>Afirmación 5: “<i>análisis causal forma parte de los ciclos de mejora de procesos.</i>” Sin embargo, no todos los defectos se deben a problemas del proceso.</p>	

Tabla 8 Hipótesis de trabajo

4 Limitaciones

En el capítulo anterior se presentaron las hipótesis sobre las cuales se construyó la propuesta. La Figura 13, muestra un diagrama de espina de pescado[133] que fue aplicado para establecer la relación de los tres apartados de esta sección con la construcción lógica de la propuesta presentada en el capítulo anterior.

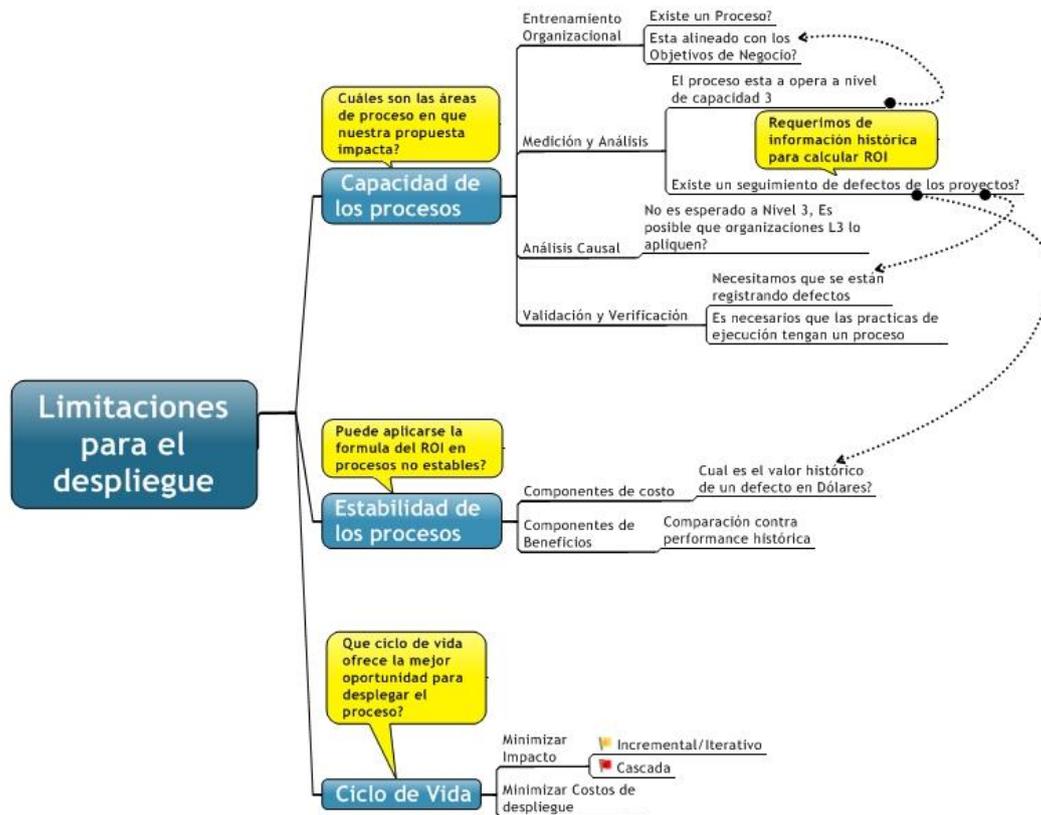


Figura 13 Limitaciones de la propuesta

4.1 Capacidad de los procesos

Para fijar un punto de partida, este capítulo establece los requerimientos que se deben cumplir para poder aplicar la propuesta de este trabajo.

Para poder desplegar el proceso, y asegurar su efectividad es necesario que la organización haya alcanzado determinadas capacidades sobre los procesos asociados a las siguientes áreas de proceso. Para definir estas capacidades se utilizará el modelo CMMI como marco de referencia.

- Entrenamiento Organizacional

- Los procesos que soportan al entrenamiento organizacional en la organización están a nivel de capacidad tres (sección 2.2.3).
- En el organigrama de la organización se puede identificar un grupo o área responsable de planificar y brindar el entrenamiento a los recursos de la organización.
- El entrenamiento brindado está alineado con los objetivos de negocio de la organización.
De manera rutinaria el grupo de entrenamiento define las necesidades de capacitación de la alta gerencia de la organización
- **Medición y Análisis**
 - Los procesos asociados al área de medición y análisis están a nivel de capacidad tres.
 - El sistema de métricas ha conseguido una medida confiable para el Costo de un Defecto (necesario para poder calcular el Retorno de Inversión).
Si no se ha alcanzado debe ser introducido como un objetivo de medición.
 - El sistema de métricas de la organización contiene datos históricos de los defectos incurridos en los proyectos.
- **Organizational Process Focus**
 - La organización tiene previstos y documentados los mecanismos para asegurar la capacidad de despliegue de nuevos procesos (OPF SG2 y OPF SG3).
- **Validación y Verificación**
 - Se están ejecutando al menos una de las sub prácticas de las áreas de proceso de Validación o Verificación (por ejemplo Testing, revisiones de pares o inspecciones).
 - Los defectos identificados en las prácticas de Validación y/o Verificación están siendo registrados en un sistema de seguimiento de defectos.

El primer punto a destacar de este conjunto de restricciones, es que no están imponiendo una limitación sobre el nivel de madurez de la organización. Los primeros dos puntos de esta sección limitan a que los procesos de Medición y Análisis y de Entrenamiento Organizacional operen a **capacidad 3**. Esta limitación es mucho menos restrictiva que forzar un nivel de madurez 3 en toda la organización. Por ejemplo, el último punto de este apartado, lo único que requiere es que se esté realizando testing. Una organización que sólo ejecuta testing tendría un rating para el área de proceso de Verificación de **0 – incompleto**, y si cumple las otras limitaciones podría implantar el proceso propuesto.. Sin embargo, de haber exigido como limitación que la organización estuviera operando a nivel de madurez 3, si se hubiesen exigido a **17** áreas de proceso operar a nivel de capacidad **3 - Definido** (entre ellas al área de Verificación), se estaría requiriendo además de Testing, Revisiones de pares, por lo que la misma organización no cumpliría esta limitación.

4.2 Estabilidad de los procesos

Un proceso predecible es aquel cuya variación natural es conocida, y por lo tanto podemos realizar expectativas en cuanto a su performance futura (sección 2.3.4.1).

Cuando un proceso es predecible, las salidas de este proceso se encuentran dentro de su rango de variación natural. Cuando se realizan modificaciones sobre un proceso, se debiera observar un cambio en el rango de variación natural del proceso, y las causas de ese cambio pueden, razonablemente, ser asignadas a las modificaciones realizadas sobre el proceso.

Cuando un proceso no cumple con estas condiciones, no es posible realizar este tipo de inferencias. Y por lo tanto se hace sumamente difícil el evaluar las modificaciones sobre el proceso.

En síntesis, para asegurar la efectividad del proceso es necesario que los procesos de producción de la fábrica exhiban comportamientos estables.

La dificultad de agregar esta restricción es que a nivel de madurez 3 (o capacidad 3) este tipo de gestión cuantitativa de los procesos no es requerida por el modelo CMMI, y por lo tanto evaluar esta restricción será una tarea de los investigadores.

4.3 Ciclo de vida de los proyectos

El pensamiento sistémico enseña que el aprendizaje se ve favorecido cuanto más corto sean los ciclos de reforzamiento (sección 2.3.5), es decir cuando la acción y la reacción estén más cerca temporalmente. Esto implica:

- Existe un ciclo de reforzamiento como el de la Figura 14.
- Cuanto más cercano en el tiempo estén las entradas de las salidas, mas fácil resultará comprenderlo.

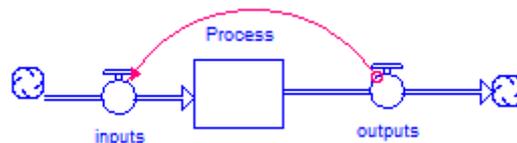


Figura 14 Bucle de reforzamiento

Un ciclo de vida en cascada, permitirá la inclusión del análisis causal únicamente al final del proyecto. Peor aún, el aprendizaje tendría que ser transferido a la organización y luego evaluado en un próximo proyecto para asegurar la transferencia a los recursos.

En un ciclo de vida iterativo, existirán más oportunidades de establecer sesiones de análisis causal, y que el resultado de ellas sea volcado en el propio proyecto que las generó.

Por esta razón se establece que los proyectos que se están ejecutando utilizando ciclos de vida iterativos serán los que podrán utilizar el proceso propuesto.

Además de las ventajas con respecto a las oportunidades de aprendizaje, es esperable que en un equipo que se organice jerárquicamente los programadores verán disminuida su

carga de trabajo sobre el final de las iteraciones. En proyectos con organizaciones menos jerárquicas, por ejemplo en proyectos SCRUM[159], las sesiones de análisis causal podrán ser agregadas en las sesiones de retrospectiva.

5 Resolución

En el presente capítulo se presenta el proceso que integra los esfuerzos del departamento de entrenamiento de una organización de software con los defectos de los proyectos de desarrollo. Para conseguir esto, el departamento de entrenamiento planificará las intervenciones de entrenamiento en los proyectos a partir de las salidas de las sesiones de análisis causal de los defectos de los proyectos.

En la sección 5.1 se presenta al proceso propuesto, detallando sus actividades y los responsables.

En la sección 5.2 se presenta el mecanismo que el proceso propuesto utiliza para calcular el retorno de inversión de las intervenciones de entrenamiento.

Luego, en la sección 5.3, se describe el plan de implantación del proceso propuesto. Este plan de implantación está diseñado para una organización que cumpla con las limitaciones expuestas en el capítulo anterior.

Finalmente en la sección 5.4 se presenta el resumen de este capítulo.

5.1 Definición del proceso para la evaluación del entrenamiento organizacional

Esta sección presenta el detalle de proceso propuesto; para definir este proceso, se utilizarán los pasos propuestos de Humphrey[17] (sección 2.3.1):

1. Determinar las necesidades de procesos y criterios de calidad
En el punto **1. Del Planteamiento**, la necesidad del proceso propuesto surge de la presión que los gestores de entrenamiento están teniendo y van a tener para justificar la inversión en sus departamentos.
2. Determinar los Objetivos del proceso
El primer *driver* del *Planteamiento* también hace énfasis en la necesidad de que los gestores de entrenamiento sean capaces de demostrar la contribución de la inversión del entrenamiento a los resultados de la organización, por lo que el segundo objetivo para el proceso es que *presente un mecanismo para el cálculo del retorno de Inversión* de las intervenciones de entrenamiento.
De acuerdo con la ecuación (1) presentada en la sección 2.1.3, existen dos factores para tener en cuenta.
 - Costos
 - Beneficios

Los **costos**, son usualmente conocidos por todos los gestores, por lo tanto el objetivo estará en proveer un mecanismo para vincular los **beneficios** del entrenamiento con la *reducción de defectos*.

3. Caracterizar el proceso actual

En la sección *Limitaciones*, se establecieron las condiciones iniciales que una organización de software debiera cumplir para poder desplegar el proceso. En la definición del proceso propuesto se referenciará al modelo CMMI para indicar los puntos en los que la propuesta se relaciona con el modelo.

4. Caracterizar el proceso objetivo

En la sección 2.1, se determinó que muy pocas empresas han llegado a conseguir medidas en el nivel 4 de Kirkpatrick. En las organizaciones de software, donde las capacidades de medición estén desarrolladas, es posible aprovechar las mismas para aplicar técnicas de retorno de inversión. Además, si el esfuerzo de análisis causal de defectos está dirigido a entender las necesidades de capacitación de los equipos de desarrollo se podrán diseñar intervenciones de entrenamiento que sean más efectivas al estar mejor alineadas con las necesidades de los proyectos.

La meta final es conseguir que los departamentos de entrenamiento se comuniquen con la alta gerencia en términos de Retorno de Inversión, como mecanismo para demostrar la efectividad de las inversiones de entrenamiento.

5. Establecer una estrategia para el desarrollo del proceso

La estrategia de definición del proceso y la propuesta fue definida en *Planteamiento*, en la Figura 12.

6. Definir el proceso

En esta sección se presenta la definición del proceso con los componentes que recomienda el autor.

7. Validar el proceso

La validación de este proceso, y del plan de despliegue que lo acompaña en esta propuesta, es presentada en la sección 6.

8. Mejorar el proceso

Este último se encuentra fuera del alcance de este volumen de tesis. Este punto corresponde a la filosofía de mejora continua que es común a todas las iniciativas de mejora de proceso (sección 2.2).

La Tabla 9, presenta al proceso propuesto para vincular análisis causal de defectos con el retorno de inversión de intervenciones de entrenamiento en organizaciones productoras de software. El formato utilizado está basado en el que utiliza [16, 17] para documentar los procesos del Personal Software Process. Para adaptarlo a las necesidades de la propuesta, se han incluido las columnas *Responsable* y *Referencias*. La primera indica que

rol dentro de la organización está asociado con ese paso, y la segunda indica la relación del proceso con el modelo CMMI o con la sección correspondiente del estado del arte.

Paso	Nombre	Responsable	Herramientas	Referencias
0	Validar Criterios de Entrada	Investigador o Grupo de Mejora de Procesos de la Organización		
1	Ejecutar el Proyecto	Equipo de Desarrollo	Sistema de Gestión de Defectos Taxonomía de clasificación de defectos (Sección 5.2.2)	CMMI VER SG1 y SG2c CMMI VAL SG2
2	Analizar Defectos	Equipo de Desarrollo	Sistema de Gestión de Defectos Diagramas de Espina	CMMI CAR SG 1 Técnicas de Análisis Causal (Sección 2.3.3)
3	Planificar Intervenciones	Equipo de Entrenamiento	Sistema de Gestión de Defectos	CMMI PP SP 2.5
4	Ejecutar Entrenamiento	Equipo de Entrenamiento Equipo de Desarrollo		CMMI OT SP 2.1
5	Evaluar el Retorno de Inversión del Entrenamiento	Equipo de Entrenamiento	Control estadístico de procesos (Sección 5.2)	CMMI OT SP 2.3 Sección 2.1
6	Comunicar Resultados	Equipo de Entrenamiento		CMMI GP 2.5 y GP 2.8
7	Evaluar continuidad del Proyecto	Si el proyecto está en ejecución retomar en paso 3		
8	Validar Criterios de Salida			

Tabla 9 Proceso Propuesto

5.1.1 Validar Criterios de Entrada

Los criterios de Entrada del paso 0, representan la verificación de las sentencias del capítulo de Limitaciones. Estas Limitaciones se han consolidado en 3 afirmaciones verificables que una organización de software puede revisar.

- El sistema de gestión de defectos incorpora la taxonomía definida. Implica que se ha desarrollado una taxonomía de clasificación de defectos y que el sistema de gestión de defectos ha sido modificado para incorporarla.
- El grupo de capacitación tiene acceso a los datos del sistema de gestión de defectos. CMMI no especifica que el equipo de entrenamiento deba tener acceso a los contenidos del sistema de gestión de defectos. Por lo tanto es importante hacer este requerimiento explícito.
- El proceso está planificado, tanto para mantener la coherencia con los niveles de capacidad requeridos, como para asegurar los recursos necesarios para cumplir con los objetivos del proceso; es fundamental que las instancias del proceso están planificadas con los planes de proyecto.

5.1.2 Ejecutar el Proyecto

El primer paso del proceso implica que el proyecto debe primero generar los datos para que puedan realizarse las sesiones de análisis causal. Existen en este paso dos actividades a destacar:

- Ejecutar actividades de Verificación y Validación
- Registrar y clasificar defectos en el sistema de gestión de defectos

El objetivo en este paso del proceso es no ser directivos en el tipo de actividades que la organización realiza. En principio, en relación a las prácticas de Ingeniería de Software anteriores al despliegue de este proceso, la única modificación que la organización debiera realizar es la clasificación de defectos encontrados en esas actividades.

En la columna *Referencias* se detalla también esta generalidad al incluir tanto a las actividades de Verificación como Validación. El objetivo final es que todos los defectos registrados en el sistema de gestión de defectos sean clasificados según la taxonomía de clasificación de la organización.

5.1.3 Analizar Defectos

En el paso 2 se realizan las sesiones de análisis causal de defectos. Como el objetivo es intervenir lo menos posible en los procesos normales de desarrollo de una fábrica de software, en el paso 2, se establece el criterio de entrada:

- El proyecto se encuentra en etapa de puesta en producción/Implementación.

Esta restricción representa la discusión sobre el ciclo de vida del proyecto mencionada en la sección 4.3. La actividad fundamental de este paso es que los desarrolladores realicen

sesiones de análisis causal de defectos. Las salidas de estas sesiones serán registradas también en el sistema de gestión de defectos de la organización.

5.1.4 Planificar Intervenciones

En el paso 3, interviene por primera vez el equipo de entrenamiento de la organización. La tarea es que el equipo de entrenamiento releve las necesidades de capacitación del proyecto a partir de las propuestas de mejora de las sesiones de análisis causal. El equipo de entrenamiento debe tener acceso al sistema de seguimiento de defectos, para utilizar los defectos registrados para priorizar las intervenciones de capacitación. Así, la intervención de capacitación seleccionada no dependerá únicamente de la sesiones de análisis causal, sino también podrá ser priorizada según el volumen de defectos.

Para asegurar la secuencialidad como criterio de entrada a este paso establece que:

- El proyecto ha realizado sesiones de análisis causal

El objetivo es establecer un punto de entrada para las intervenciones de entrenamiento, fijando así las iteraciones que abarcan los defectos utilizados por el equipo de entrenamiento para planificar la intervención. Esto va a ser importante en el momento de establecer el marco de tiempo en el que se calculará el retorno de Inversión en entrenamiento, que es éste el objetivo de esta propuesta (sección 5.2.4).

5.1.5 Ejecutar Entrenamiento

Tanto en la planificación como en la ejecución de la intervención de capacitación se realizará según los procesos de la organización destino. A los efectos del cálculo del ROI, la organización deberá registrar el costo de cada intervención de entrenamiento. Aunque esto último no está explícito en el modelo CMMI, es esperable que un departamento de entrenamiento operando a nivel de capacidad 3 lo esté haciendo.

5.1.6 Evaluar el Retorno de Inversión del Entrenamiento

El paso 5 corresponde a las tareas necesarias para realizar el cálculo del retorno de Inversión. El detalle de los pasos para realizar el cálculo ROI en el proceso se detalla en la sección 5.2. A modo de resumen, para calcular el ROI se comparan los defectos incurridos en las iteraciones siguientes a que se realizó la intervención del entrenamiento con la proyección de defectos calculada en función del histórico de defectos de la organización. Es de esperar que la disminución de defectos observada sea suficiente para cubrir los costos de desplegar este proceso.

5.1.7 Comunicar Resultados

Este punto representa el objetivo final de esta propuesta que era proveerle al departamento de entrenamiento la capacidad de justificar la inversión utilizando datos tangibles, en especial a través de un reporte que identifique el retorno de inversión en entrenamiento de las intervenciones de capacitación de los proyectos.

5.1.8 Evaluar continuidad del Proyecto

Este último paso representa nuevamente la intención de que este proceso sea desplegado en proyectos con ciclos de vida iterativos, que darán más oportunidades de aplicar el proceso y de esperar Retornos de Inversión positivos.

5.1.9 Validar criterios de salida

Para terminar, se estableció un último paso con los criterios que una organización puede seguir para terminar con la ejecución del proyecto.

- Se han completado y comunicado los Informes con el Retorno de inversión de las intervenciones de entrenamiento. Realizar los informes y comunicarlos, implicaría que se ha podido seguir el proceso y el mecanismo para calcular el ROI de las intervenciones.
- Se han realizado las sesiones de análisis causal y registrado sus salidas. Aunque sin éstas, no sería posible realizar las intervenciones, es esperable que las sesiones de análisis causal generen más propuestas de capacitación que las que el equipo de capacitación pueda abarcar. Por lo tanto, es importante verificar que todas se encuentran registradas en el sistema.
- Están los defectos clasificados y registrados en el sistema de gestión de defectos. De manera similar, los defectos clasificados, representarán una fuente de conocimiento a la que la organización puede acceder mas allá del uso que este proceso les da.
- Se ha terminado el proyecto. Representa el punto de control lógico para no seguir ejecutando intervenciones de entrenamiento sobre los recursos de un proyecto terminado.

Como es común en las representaciones de un proceso, los pasos aquí descritos pueden ser ejecutados en paralelo y no es necesario respetar la secuencialidad con que fueron explicados. Con este fin, se incluyeron las descripciones de los criterios de entrada en los pasos que correspondía estableciendo así mecanismos de control sobre el flujo de ejecución del proceso.

5.2 Cálculo del ROI en el proceso

Para calcular el ROI de la aplicación de la propuesta de tesis se utilizará una modificación (ver sección 5.2.4) a la fórmula del ROI (1).

La adaptación propuesta resulta de aceptar la variación inherente al proceso de construcción de software. Mediante la aplicación de la formula (1), se podría establecer un cálculo del ROI en función del comportamiento “promedio” del proceso. Pero esto no se adapta a la realidad de una fábrica de software, por lo que se aporta aquí una adaptación a la fórmula que contemple las variaciones naturales de los procesos.

A diferencia de [64], donde se estiman los beneficios mínimos y máximos previo a la intervención del entrenamiento, para luego establecer el cálculo en función de los

beneficios obtenidos. En esta propuesta se aplican los límites de control para realizar el cálculo del ROI de la intervención de entrenamiento.

En la secciones siguientes (5.2.1 y 5.2.3) se presentarán las variables que se deberán tener en cuenta para calcular el retorno de inversión. El cálculo de beneficios, dependerá del tipo de defectos a concentrarse en las intervenciones de entrenamiento, por lo que en la sección 5.2.2, se presenta una discusión del tipo de defectos que esta propuesta pretende abarcar. Y finalmente, en la sección 5.2.4, se presenta la modificación a la forma de calcular el Retorno de Inversión que tendrá en cuenta la variación natural de los procesos.

5.2.1 Variables de Costos

La Tabla 10 describe las variables de costos identificadas para el despliegue de este proceso. Los costos de la implantación del proceso se pueden dividir en dos grandes tipos:

- costos de Establecer el proceso,
- costos de Ejecutar el proceso.

Tipo de Costo	Descripción del Costo	Costo Fijo/Costo Variable
Costos de Establecer el Proceso	Desarrollar taxonomía de clasificación	Fijo
	Modificar sistema de gestión de defectos	Fijo
	Desarrollar capacidad de análisis causal	Fijo
Costos de Ejecutar el Proceso	Esfuerzo en reuniones de Análisis Causal	Variable
	Esfuerzo en Preparación de la intervención entrenamiento	Variable
	Ejecución de la Intervención de Entrenamiento	Variable

Tabla 10 Variables de Costos identificados Proceso

Los Costos de Establecer el proceso, implican las tareas que es necesario completar para llevar a una organización que se encuentre dentro de las condiciones expresadas en la sección Limitaciones. Estas tareas están asociadas a evaluar el costo de las primeras dos etapas del proceso de despliegue descrito en la sección 5.3. Estos son:

- El costo de desarrollar la taxonomía de clasificación de defectos. Bajo esta descripción se agrupan todas las tareas que la organización realice con el objetivo de generar una taxonomía de clasificación que se adecue a sus necesidades de información y objetivos de negocio (ver sección 5.2.2).
- El costo de modificar el sistema de clasificación de defectos. La modificación del sistema de gestión de defectos para que éste se adecue a la Taxonomía desarrollada, podrá o no implicar un proyecto de desarrollo en sí mismo.
- El costo de desarrollar la capacidad de análisis causal en los recursos y procesos de la fábrica. Implica la ejecución por parte de la organización de sus procesos de

despliegue de procesos para que se diseñe y despliegue un proceso de análisis causal de defectos para que pueda ser seguido durante la ejecución del proceso propuesto.

Estos ítems de costo también traen consigo necesidades de entrenamiento, y el costo de éstas será imputado a cada una de estas categorías. Por ejemplo, tener confianza en el análisis causal de diferentes proyectos, la organización deberá asegurar que todos sus integrantes serán capaces de clasificar defectos de manera consistente a través de la organización (ver sección 6.2.2). Este costo de capacitación también será tenido en cuenta.

Este grupo de costos se pueden clasificar como *costos Fijos*, ya que son costos en los que la organización debiera incurrir por única vez durante los preparativos para desplegar el proceso. Por lo tanto, se verán amortizados a sucesivas iteraciones del proceso.

Los *Costos de Ejecutar el Proceso*, implican las tareas resultantes de la ejecución del proceso. De las 6 tareas que componen el proceso que se presentó en la sección 5.1 serán incluidos los costos asociados a estas tres actividades:

- Esfuerzo en reuniones de análisis causal. Será tenido en cuenta el tiempo y costo de la asistencia de los recursos a las sesiones de análisis causal.
- Esfuerzo en preparación de la intervención de entrenamiento. Dependiendo del sistema de métricas, el departamento de entrenamiento podría ya estar llevando esta información para cada uno de sus entrenamientos. La identificación e inclusión de este punto entre las variables de cálculo, realiza la vinculación entre la inversión en entrenamiento, las sesiones de análisis causal y el retorno de inversión.
- Esfuerzo y Recursos comprometidos en la intervención del entrenamiento. Implica que se tendrán en cuenta los costos de ejecutar el entrenamiento. Incluyendo los costos de:
 - asistencia de los miembros del equipo
 - costo del docente
 - y cualquier costo de materiales que sea necesario para ejecutar la intervención del entrenamiento

5.2.2 Taxonomía de clasificación

Como se repasó en la sección 2.3.3.1 en la literatura se han propuesto varias taxonomías de clasificación de defectos. Siguiendo los criterios de Slaughter[119] donde los mejores resultados se obtienen cuando las organizaciones desarrollan sus taxonomías de defectos basadas en sus necesidades de información. Y el desarrollo de una taxonomía de clasificación propia de la organización objetivo será uno de los puntos incluidos en el plan de implantación propuesto en la sección 5.3

A modo de referencia, la propuesta recomienda la taxonomía de clasificación de ODC descrita en 2.3.3.1.1. Los motivos para seleccionar a esta taxonomía son los siguientes:

- ODC está probada y ha sido utilizada por la industria.

- ODC es adaptable y puede utilizarse como abstracción de la taxonomía de la organización, siendo este un elemento que nos permitirá publicar los resultados obtenidos sin comprometer la información de defectos de la organización.
- una de las categorías de ODC explícitamente hace referencia a los defectos de *Capacitación*. Aplicando el mecanismo propuesto, se podrán identificar y ejecutar intervenciones de entrenamiento que tengan por objetivo reducir los defectos de entrenamiento.

5.2.3 Variables de beneficios

El cálculo de Beneficios se realizará mediante la comparación de los defectos esperados, calculados a partir de los datos históricos de los proyectos de la organización con los defectos detectados en el proyecto objetivo.

Se espera que la intervención de entrenamiento desarrollada a partir del análisis causal de defectos reduzca el volumen de defectos del proyecto.

Las ecuaciones (3) y (4) detallan el cálculo del factor de beneficios.

$$(3) \text{Beneficios} = \Delta \text{Defectos} * \text{CostoDefecto}_{\text{MediaHistorica}}$$

donde:

$$(4) \Delta \text{Defectos} = \text{Defectos}_{\text{Esperados}} - \text{Defectos}_{\text{Observados}}$$

- $\text{CostoDefecto}_{\text{MediaHistorica}}$, representa la medida confiable que la organización ha conseguido para el costo de sus defectos (según se estableció en la sección 4.1)
- $\text{Defectos}_{\text{Esperados}}$, serán calculados estableciendo un gráfico de control de proceso basado en la información histórica de los proyectos de la fábrica, como se explicará en la sección siguiente.
- $\text{Defectos}_{\text{Observados}}$, será la medición de los defectos

5.2.4 Adaptación de la fórmula de ROI

Para completar la propuesta de cálculo, falta proveer un mecanismo para establecer cuál es el nivel esperado de defectos. Para esto se aplicará el mecanismo de gestión cuantitativa de procesos descrito en la sección 2.3.4.

En la sección 4.2 se expuso como una limitación que los procesos de la organización fueran predecibles. Esto significa que mediante la aplicación de las técnicas mencionadas en la sección 2.3.4 será posible determinar el rango de variación natural del proceso. En este caso, significa que será posible establecer un rango de defectos esperados en los proyectos de la organización.

A partir de este rango de variaciones es posible calcular tres resultados del Retorno de Inversión de la propuesta según se explica en la Figura 15.

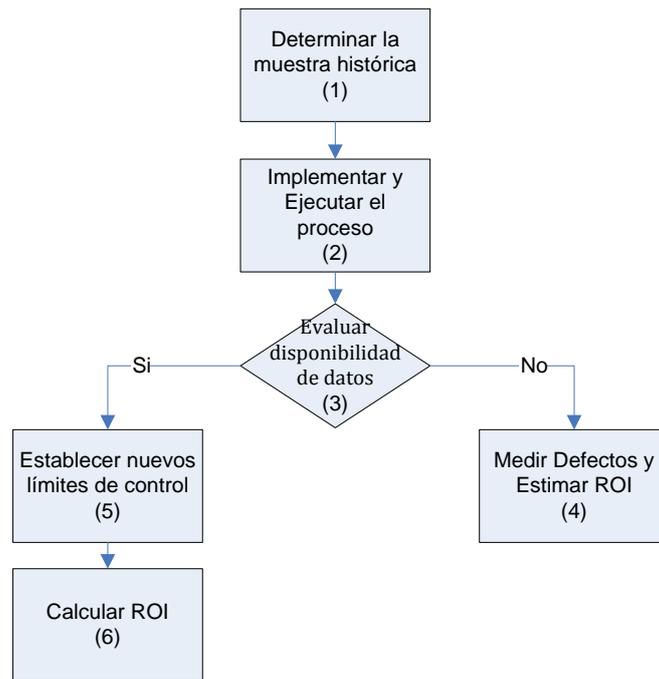


Figura 15 Mecanismo para calcular el ROI

1. Determinar la muestra histórica.
Luego de seleccionar los proyectos de la organización que determinarán la muestra histórica, aplicar las técnicas de control de proceso de la sección 2.3.4, y calcular los límites de control de defectos de los proyectos.
Se recomienda utilizar los siguientes indicadores:
 - Porcentaje de Defectos de Educación
 - Porcentaje de Defectos de Educación Normalizados
 - Normalizados por:
 - Tamaño del Proyecto
 - Esfuerzo
 - Esfuerzo del equipo de testing
2. Implementar el proceso y ejecutar el proyecto.
Determinar en qué proyectos se va a implantar y ejecutar el proceso propuesto, ejecutarlo y recolectar los datos detallados en este capítulo.
3. Evaluar disponibilidad de datos
Para establecer nuevos límites de control se necesitan por lo menos **tres** nuevos puntos (iteraciones).
4. Medir Defectos y Estimar ROI
Al no contar con suficientes datos como para calcular los nuevos límites de control, el cálculo del ROI se debe realizar proyecto a proyecto, comparando de la siguiente manera:
 - **Peor Caso**, Límite de control superior de la muestra histórica, comparado con los defectos observados en el proyecto.
 - **Caso Observado**, media histórica, comparado con los defectos observados en el proyecto.
 - **Mejor Caso**, Límite de control inferior de la muestra histórica, comparado con los defectos observados en el proyecto.
5. Establecer nuevos límites de control

Aplicar nuevamente las técnicas de control estadísticos de procesos para los proyectos donde se haya aplicado el proceso propuesto. Se obtienen así nuevos límites de control.

6. Calcular ROI

Con los límites de control de los proyectos históricos, y los nuevos límites de control calcular el ROI de la siguiente manera (según ejemplifica la siguiente figura):

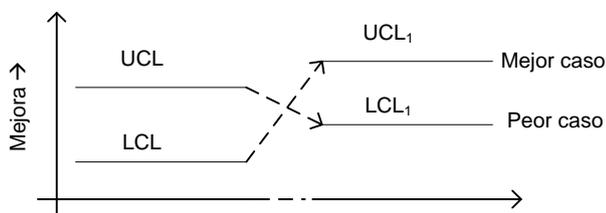


Figura 16 Interpretación de los límites de control

- **Peor Caso.**
 - $Defectos_{Esperados} = UCL_{Historico}$
 - $Defectos_{Observados} = LCL_{Nuevos\Pr oyectos}$
- **Caso Observado.**
 - $Defectos_{Esperados} = Pr omedio_{Historico}$
 - $Defectos_{Observados} = Pr omedio_{Nuevos\Pr oyectos}$
- **Mejor Caso,**
 - $Defectos_{Esperados} = LCL_{Historico}$
 - $Defectos_{Observados} = UCL_{Nuevos\Pr oyectos}$

Al comunicar resultados de Retorno de Inversión, los valores deben ponerse en contexto adjuntando un período de tiempo en los que el retorno de inversión se obtiene. Es decir, se deberá tener en cuenta la fecha en que se incurre en el primer costo imputado a alguna de las variables de costo, y la fecha en que se congela el proyecto para medir los defectos y calcular el ROI.

5.2.5 Interpretación de los posibles resultados

La interpretación de los resultados de la propuesta no son tan directos como interpretar el cálculo de ROI utilizando los valores promedio. Al aplicar control estadístico del proceso al cálculo del ROI, los gestores de entrenamiento deberán tener en cuenta los límites de control para sacar conclusiones de los resultados de la intervención. La Figura 17, presenta una explicación de los posibles resultados que pueden obtenerse mediante la aplicación de la propuesta. En todos los caso se asume que la dirección de “mejora” para el negocio esta dada por un incremento en el eje vertical.

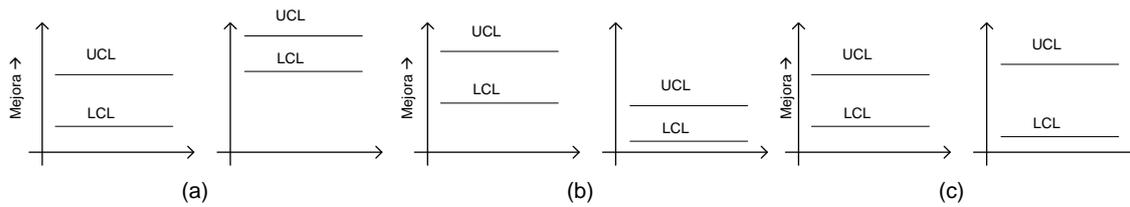


Figura 17 Interpretación de resultados

- (a) En este caso, la mejora es observable a nivel de la gráfica. Se observa una reducción en la variación, a la vez que el rango de control se ha desplazado en la dirección de los objetivos de negocio.
- (b) Este caso representa el opuesto al caso anterior. Se observa una clara pérdida de performance.
- (c) Este caso es el más difícil de evaluar, principalmente por que se observa un aumento en la variación del proceso. En estos casos, el criterio de mejora será subjetivo ya que cuantitativamente no es posible hablar de mejora en términos de control estadístico.

5.3 Plan de Implantación del Proceso Propuesto

En esta sección se describe el plan de trabajo propuesto para desplegar el proceso descrito en la sección 5.1. El plan de implantación está dividido en 3 grandes actividades (ver Tabla 11).

El objetivo de la primera actividad del plan de despliegue involucra la generación de las capacidades de clasificación por parte de la organización. Identificando las siguientes 3 tareas:

- Diseñar y Validar la taxonomía de clasificación: Esta tarea implica que la organización debe desarrollar su propia taxonomía de clasificación. El desarrollo de esta taxonomía debiera incluir el rotulo de cada categoría y los criterios que los defectos deben cumplir para integrar esa categoría. La organización deberá asignar los recursos necesarios (humanos y materiales) para que la taxonomía de clasificación respete los objetivos de negocio de la organización. La tarea de validación implica que el equipo de trabajo debe asegurarse de que la taxonomía de clasificación sea completa, es decir, que todos los defectos en los que la organización puede incurrir estén contemplados en las categorías de la taxonomía. Como herramienta de validación, el equipo de trabajo también cuenta con la base de datos histórica del sistema de gestión de defectos.
- Modificar el Sistema de Gestión de Defectos: Con la taxonomía desarrollada y validada, el objetivo de esta tarea es modificar el sistema de gestión de defectos para que la taxonomía quede integrada en él.
- Desplegar el esquema de clasificación en la organización: El foco de esta tarea consiste en entrenar a los recursos de la organización en el uso de la taxonomía. Para que la organización puede realizar un análisis de los defectos registrados es necesario que la misma tenga confianza en que los defectos son clasificados en forma consistente a lo largo de la organización.

Antes de pasar a la siguiente etapa, la organización debiera ser capaz de contestar afirmativamente estas 3 preguntas a modo de *Lista de Verificación*:

- Todos los defectos reportados en el sistema de gestión de defectos (históricos), pueden ser clasificados en alguna de las categorías de la taxonomía desarrollada.
- La taxonomía desarrollada está integrada con el sistema de gestión de defectos.
- Todos los recursos de la organización están entrenados en el uso de la taxonomía con el sistema de gestión de defectos.

Etapa	Actividades	Responsable
Establecer Capacidad de Clasificación en la organización	Diseñar y Validar la taxonomía de clasificación	Analistas de software de la organización
	Modificar el Sistema de Gestión de defectos	Responsables de mantenimiento de herramientas
	Desplegar el esquema de clasificación en la organización.	Grupo de Mejora de Procesos
Desplegar Proceso de Análisis Causal	Establecer y Mantener el proceso de Análisis causal	Grupo de Mejora de Procesos
	Entrenar a la organización y desplegar el proceso	Área de Entrenamiento
Seleccionar e implementar el proceso completo en un Piloto	Seleccionar el proyecto piloto	Gerencia de la Organización
	Ejecutar el proceso (clasificación de defectos, análisis causal e intervención de entrenamiento)	Proyecto Piloto
Analizar resultados	Analizar resultados y estudiar despliegue	Investigadores y Gerencia de la Organización

Tabla 11 Plan de Despliegue

El objetivo de la segunda actividad es desplegar el proceso de análisis causal de defectos en la organización. En la sección 2.2.3.1.2 se detalló que no es esperable que una organización CMMI nivel 3 de madurez haya implementado prácticas de análisis causal. Por lo tanto el despliegue de las capacidades de análisis causal debe ser un punto a incluir en este plan de implantación. En la sección 2.3.3 se presentó el estudio de las diferentes alternativas para implementar análisis causal de defectos, cada organización deberá implementarlo siguiendo las restricciones de su cultura y sus capacidades. Las tareas relacionadas con esta actividad son las siguientes:

- Establecer y Mantener el proceso de Análisis causal. Esta tarea implica que la organización debe seguir sus procesos para definir el proceso de análisis causal dentro de la organización.

- Entrenar a la organización y desplegar el proceso. Esta tarea implica realizar las actividades de capacitación y de despliegue que la organización considere necesaria para que el proceso de análisis causal quede institucionalizado en la organización.

A modo de lista de verificación, una vez completada esta actividad, la organización debiera poder responder afirmativamente estas preguntas:

- Los nuevos proyectos tienen planificadas las sesiones de análisis causal
- Los recursos de la organización están capacitados en las tareas que el nuevo proceso de análisis causal implica

La tercera actividad del plan de implementación se centra en la selección de el/los proyectos piloto para desplegar la propuesta de procesos. Esta selección dependerá de las prioridades y los tipos de los proyectos de la organización. Con el proyecto piloto seleccionado, la organización deberá proceder a aplicar sus procesos de despliegue de procesos para desplegar en el piloto el proceso propuesto.

Finalmente, el objetivo de la etapa de Análisis de resultados será el de realizar el cálculo de ROI que será presentado a la alta gerencia de la organización para que ésta pueda evaluar si continúa el despliegue del proceso en el resto de la organización.

5.4 Resumen de la Resolución

El proceso propuesto pretende vincular el análisis causal de defectos con los resultados del entrenamiento organizacional. La intención es que mediante el análisis causal de los defectos reportados por los equipos de desarrollo, los departamentos de entrenamiento tengan más herramientas para planificar y ejecutar intervenciones de entrenamiento que aporten un valor agregado a los proyectos de desarrollo de software en ejecución.

Adjunto al proceso propuesto se plantea un método para calcular el Retorno de Inversión de estas intervenciones de entrenamiento. El método de cálculo de retorno de inversión propuesto construye a partir de los métodos relevados en las secciones 2.1.3 y 2.3.2. Nuestra forma de calcular el ROI tiene en cuenta la variación natural de los procesos de desarrollo. De esta forma nuestra propuesta consigue vincular el análisis causal de defectos con los resultados del entrenamiento organizacional dentro de una organización productora de software. Se consigue además un mecanismo reproducible que es capaz de resolver un problema abierto en la literatura de entrenamiento organizacional (ver sección 2.1).

Además, para facilitar la adopción por parte de las empresas de software de nuestro proceso, se ha propuesto un plan de implantación. El plan de implantación tiene por objetivo crear en la organización de software objetivo las herramientas y las capacidades de los procesos que serán necesarias para la implantación del proceso propuesto. Este plan de implantación se construye a partir de las capacidades esperables de una organización de software en que los procesos cumplan con las Limitaciones planteadas.

6 Experimentación

En este capítulo se detallan las experiencias de validación a las que fueron sometidos los elementos de la propuesta. El objetivo de este trabajo de investigación es establecer un proceso que vincule el análisis causal de defectos con el retorno de inversión en entrenamiento. En el capítulo anterior se presentaron los elementos de la propuesta:

- El proceso propuesto,
 - Su mecanismo para el cálculo del Retorno de Inversión
- El Plan de despliegue del proceso.

Estos tres elementos se diseñaron para que una organización productora de software que se encuentre en las condiciones descritas en la sección Limitaciones, pueda implementar el proceso y presentar resultados de Retorno de Inversión de las intervenciones de entrenamiento. Por lo tanto, para realizar la experimentación de este conjunto, fue necesario encontrar una fábrica de software que cumpla con las Limitaciones planteadas. Una vez verificadas, se procedió a ejecutar los pasos indicados en el plan de implantación del proceso, para finalmente analizar los resultados.

Este capítulo de validaciones se ha escrito siguiendo el orden temporal de ejecución de las experimentaciones:

1. Se detalla el plan de implantación
2. Se describe la ejecución del proceso propuesto
3. Se presenta el análisis de resultados.

Con el objetivo de reafirmar los resultados obtenidos en la experimentación en la fábrica, se realizaron también experimentos fuera del entorno de la fábrica.

La Tabla 12 describe los puntos de esta propuesta que fueron experimentados, y el mecanismo que se utilizó para experimentarlos.

Elementos de la Propuesta	Etapas	Resumen del Tipo de Experimentación
Plan de Despliegue	Limitaciones y Pre requisitos	Se estudió cómo la fábrica de Software seleccionada cumplía con las limitaciones del trabajo (sección 6.1).
	Establecer capacidad de clasificación de la organización	<p>En la sección 6.2 se describe el Plan de despliegue utilizado en la fábrica de software. Se realizaron los siguientes pasos para verificar las actividades mostradas:</p> <p>La sección 6.2.1 describe como se desarrolló y validó la taxonomía de clasificación de la fábrica de software</p> <p>La sección 6.2.2 describe el sistema de gestión de defectos y la modificación realizada</p> <p>La sección 6.2.3 describe el despliegue de la taxonomía en la fábrica de software, incluyendo un experimento realizado en conjunto con la fábrica y con miembros del Laboratorio de Lenguajes y Sistemas de Información de la UPM (sección 6.2.3.4)</p> <p>La sección 6.2.3.5 presenta un experimento realizado en el laboratorio para estudiar la viabilidad de aplicar técnicas automáticas de clasificación en las organizaciones</p>
	Desplegar Proceso de Análisis causal	Se delegó en la fábrica de software la responsabilidad de ejecutar esta etapa, la sección 6.4.2 resume los pasos tomados por la fábrica
	Seleccionar e Implementar el piloto	En la sección 6.3 se describe cómo se simuló la ejecución del proceso para proyectar el ROI de la propuesta en la fábrica
	Analizar Resultados	
	Desplegar el Proceso	Se describe en la próxima fila
Proceso Propuesto	Ejecutar el Proyecto	Las únicas modificaciones realizadas sobre los procesos de Validación y Verificación están asociadas a la clasificación de defectos según la taxonomía.
	Analizar Defectos	Vinculado al despliegue del proceso de análisis casual, las adaptaciones se describen en la sección 6.4.2
	Planificar	La selección y planificación del entrenamiento fue realizada por el departamento de capacitación de la

	Intervenciones	fábrica de software (sección 6.4.3)
	Ejecutar entrenamiento	El entrenamiento fue brindado por el departamento de capacitación de la fábrica de software, la sección 6.4.4 presenta las evaluaciones nivel 2 de los asistentes
	Evaluar el ROI	Se presenta en la sección 6.5
Análisis de Resultados	Comunicar resultados	Los resultados del cálculo del ROI de la propuesta se presentan en la sección 6.5.5.3. Además, para facilitar la comunicación de los resultados se construyó un modelo de decisiones de pensamiento sistémico (sección 6.6).

Tabla 12 Partes experimentadas de la propuesta

Como muestra la tabla, el objetivo al organizar la experimentación fue llevar la totalidad del proceso a la fábrica de software. Aceptando el riesgo de gestionar los procesos de cambio que un despliegue de procesos implica en una organización. En este contexto se respetó en todo momento la premisa de limitar el impacto de la implantación del proceso propuesto en las prácticas habituales de la organización. Las concesiones realizadas con la fábrica que se detallaron en la tabla, se realizaron teniendo en cuenta este objetivo. De la misma manera, los experimentos en los distintos laboratorios fueron diseñados para reforzar la experimentación de alguno de estos puntos.

6.1 Breve descripción de la fábrica de software

La organización donde se realizó la experimentación del proceso y el plan de despliegue es una fábrica de software ubicada en Montevideo, Uruguay. La organización tiene más de 10 años de vida, y la fábrica de software cuenta con 8 años de operación. Sus proyectos están organizados en cuatro líneas de producción:

- Desarrollo a Medida sobre Microsoft .NET, son proyectos de desarrollo basados exclusivamente en la plataforma Microsoft .NET
- Desarrollo de Portales, son proyectos de intranet y extranet basados en soluciones de gestión de contenidos, principalmente basadas en Microsoft Office Sharepoint Portal.
- Customizaciones de Productos Microsoft, son proyectos de desarrollo que se basan en la customización de productos Microsoft como Microsoft Biztalk Server, Microsoft Scorecard manager entre otros.
- Customizaciones de Productos propietarios. La organización ha desarrollado varios aceleradores (productos para agilizar el despliegue de soluciones comunes) para Bancos, para integración de sistemas. Esta línea de producción se encarga de la ejecución de estos proyectos.

En Abril de 2007, esta organización de software completó exitosamente un proceso de evaluación CMMI nivel de madurez 3. El alcance de la evaluación incluyó a las cuatro líneas de producción. De esta manera, a la luz de la limitación de la capacidad de los procesos, es esperable que la organización tenga en todos los procesos necesarios un nivel de capacidad 3.

La siguiente tabla presenta una breve discusión de cómo se posiciona esta organización frente a las limitaciones previstas en la sección 4.

Requerimiento	Detalle	Comentario
Capacidad de los procesos de la Organización	Existe un departamento responsable de planificar el entrenamiento	Si, el departamento de entrenamiento en su estructura actual está institucionalizado desde el 2004.
	Existe un sistema de métricas confiable que incluye el seguimiento de los defectos	Si, el sistema de métricas de la organización está organizado en función de la integración de los productos del paquete Microsoft Office. Los datos son recolectados en Microsoft Office Sharepoint Portal y son analizados en Bases de Datos Access o Microsoft SQL Server.
	La función de mejora de proceso está institucionalizada	Si, la oficina de procesos está formada desde el 2005 cuando se inició con la iniciativa de adopción del modelo CMMI.
	Se están realizando tareas de Verificación y Validación	Si, existe un equipo independiente de los proyectos que provee a los equipos de desarrollo de la fábrica servicios de testing funcional.
Ciclo de Vida iterativo de los proyectos		Si, el ciclo de vida estándar de los proyectos está basado en las recomendaciones del Microsoft Solutions Framework for CMMI[129].
Proceso Predecible		Para poder asegurar que un proceso sea predecible se necesita que la organización tenga un conocimiento cuantitativo de las capacidades de ese proceso (sección 4.2). Al momento de comenzar con la experimentación del proceso propuesto, la organización llevaba

		seis meses desde su evaluación nivel 3. Por lo que se consideró que los procesos estaban lo suficientemente maduros como para soportar las modificaciones que esta propuesta les imponía (ver sección 6.6).
--	--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

6.1.1 Algunas características distintivas de esta organización

Como toda organización, tiene ciertas características que influyeron el desarrollo de la experimentación. Los puntos aquí descritos serán referenciados durante el análisis de resultados de la experimentación para poner a los datos obtenidos en el contexto de la realidad de la organización.

- Rotación del personal entorno al 25%. La fábrica de software contaba en el 2007 con una rotación media histórica del 25% anual. Este es un número un poco superior a la media de la industria del software uruguayo. El movimiento de recursos entre empresas de software es normal en la industria de software uruguaya. En el año 2006, la Cámara Uruguaya de Tecnologías de la Información reportó que la industria del software era la industria de desempleo cero[160] en el Uruguay. Este distintivo se mantuvo hasta la desaceleración económica del 2009.
- El alcance y tamaño de un proyecto típico de la organización (en cualquiera de sus líneas de producción) es de 6-8 meses de duración, con un equipo integrado por un jefe de proyecto, un analista y entre 3 y 4 programadores.
- Historia de Capacitación. Esta organización comenzó brindando servicios corporativos de capacitación. El foco en capacitación se mantiene desde entonces. Una de las políticas de capacitación de la organización es que cada rol debe tener al menos un seminario de capacitación cada dos meses. El resultado de esta política es que los recursos de la organización están acostumbrados, y esperan, recibir seminarios.
- Sistema de Métricas. El sistema de métricas de la organización está basado en la recolección de datos a partir de la intranet corporativa (un despliegue de Microsoft Office Sharepoint Portal), que es integrado a través de otra aplicación Microsoft Office (Microsoft Infopath), con un servidor Microsoft SQL Server. Esta integración permite el fácil y rápido despliegue de formularios electrónicos para la carga de datos de los procesos. Los formularios Infopath son la base para la recolección de datos de defectos de testing, defectos de revisiones de pares, no conformidades de PPQA, eventos de capacitación, e incidentes de los proyectos.

Como todo elemento de software, estos formularios son mantenidos por la organización y el despliegue de nuevas versiones se realiza semanalmente cuando el uso del portal disminuye.

- Reducción en la plantilla y en los proyectos. Al final de año 2007, la organización pasó por un proceso de reestructuración interna que afectó al volumen de recursos y al tipo de proyectos. Esta modificación en la empresa ocurrió durante la experimentación en el momento de despliegue del proceso propuesto. Como resultado las métricas y resultados en esta sección van a reflejar esta reestructuración. En la sección 6.6 se presenta una discusión del impacto de este cambio en la organización sobre los resultados obtenidos.

6.2 Adaptaciones realizadas al proceso para el despliegue en la fábrica

El proceso de despliegue propuesto en la sección 5.3 tuvo que ser adaptado para atender a las necesidades y restricciones de la organización; parte de esta adaptación ya se hizo evidente en la Tabla 12.

Las adaptaciones realizadas al proceso propuesto fueron:

- **M1.** La organización clasificaba los defectos por prioridad, no contemplaba una clasificación por tipo de defectos. Por lo que se agregaron en el plan de implantación las actividades para conseguir esta capacidad de clasificación (secciones 6.2.1 y 6.2.2)
- **M2.** Los recursos de la fábrica debieron cambiar la forma en que registran defectos para acomodarse a los cambios del sistema de gestión de defectos (sección 6.1.1).
- **M3.** Se establecieron políticas y pautas para realizar el análisis causal de defectos en la organización (sección 6.4.2).
- **M4.** La organización modificó la priorización a la hora de seleccionar el contenido de las intervenciones (sección 6.4.3)
- **M5.** Los investigadores asumieron la tarea de realizar el cálculo del retorno de inversión (6.5).

La Tabla 13 presenta estas modificaciones en el contexto de los pasos del proceso propuesto.

Paso	Nombre	Modificaciones
0	Validar Criterios de Entrada	M1 Clasificar Defectos por Tipo.
1	Ejecutar el Proyecto	M2 Modificar registro de defectos.
2	Analizar Defectos	M3 Desplegar el proceso de Análisis Causal.
3	Planificar Intervenciones	M4. Adaptar criterios de selección de intervenciones propuestos.
4	Ejecutar el Entrenamiento	
5	Evaluar el Retorno de Inversión del Entrenamiento	M5 El cálculo del retorno de inversión fue realizado por los investigadores.
6	Comunicar Resultados	Sin modificaciones.
7	Evaluar continuidad del Proyecto	
8	Validar Criterios de Salida	

Tabla 13 Adaptaciones al proceso

El resto de este capítulo presenta y discute los pasos y resultados de cada una de estas etapas.

6.2.1 Diseño y validación de la taxonomía de Clasificación

Para poder clasificar defectos por tipo, se organizó en la fábrica de software un plan (Tabla 14) para diseñar una taxonomía de clasificación que respetara sus objetivos de negocio[119].

Actividades	Recursos	Producto de Trabajo
Desarrollar la clasificación		
Armaz grupos de trabajo		Grupo de trabajo (GT)
Especializar grupos de trabajo	GT	Sub equipos (TWG y TVT)
Estudio de Base de datos de defectos	TWG	Taxonomía en borrador
Crear Taxonomía	TWG	
Validar Taxonomía	TVT	Taxonomía validada
Criterios de Salida	Taxonomía validada	

Tabla 14 Plan de trabajo para el diseño de la taxonomía

En primer lugar la fábrica de software asignó los recursos que integrarían un grupo de trabajo responsable de armar la taxonomía. El grupo estuvo compuesto por analistas de software de la organización.

El grupo de trabajo fue dividido en 2, al primero se le encargó el desarrollo de la Taxonomía de clasificación. A este grupo se llamará Taxonomy Working Group - TWG. El segundo grupo estuvo encargado de validar los resultados del grupo de trabajo de la taxonomía (Taxonomy Validation Team - TVT).

A los miembros del TWG se les pidió que estudiaran defectos tomados de la base de datos del sistema de gestión de defectos de la organización. Con este objetivo se seleccionaron los defectos registrados para los proyectos del año 2007 de las diferentes líneas de producción de la fábrica de software.

Como resultado de este trabajo deberían producir una Taxonomía de clasificación, que contuviese criterios de clasificación para cada una de las categorías que esta incluyera. La taxonomía resultante del trabajo del grupo Taxonomy Working Group, es el resultado de dos sesiones de análisis de los defectos de la fábrica. Se desarrolló un esquema de clasificación que consiste en 16 categorías, con sus criterios de clasificación. La Tabla 15 muestra las categorías de clasificación que desarrollaron en la organización.

Para facilitar el análisis de datos de la investigación se agregó a la clasificación de la organización una equivalencia con la taxonomía de clasificación de ODC[125]. Esto se realizó con el objetivo de permitirnos concentrar en los tipos de defectos de *Educación o Transcripción*, y para proteger a la organización, ya que la información de defectos es información sensible.

Haber establecido un nivel de abstracción con respecto a la clasificación de la organización resultó útil ya que la taxonomía a la que llegaron en Diciembre de 2007 tuvo 2 revisiones durante el año 2008. Como se ve en la Tabla 15, la clasificación utilizada por la organización (columna "Categoría de Clasificación") no se parece a ninguna de las clasificaciones relevadas en la sección 2.3.3.1, sino que responde a las necesidades de información de la organización.

A continuación se les entregó la Taxonomía de clasificación al TVT. La tarea consistía en que debían asegurar que la Taxonomía desarrollada era completa. Es decir, que la totalidad de los defectos de la muestra pudieran clasificarse en alguna de las categorías. Durante este proceso las categorías y los criterios se fueron ajustando para que al segundo grupo le fuera posible completar su asignación.

El resultado final de este proceso fue la Taxonomía de clasificación que se puso en producción en la fábrica. El proceso de diseño de la taxonomía de clasificación terminó clasificando alrededor del 20% de los defectos registrados en el 2007. Al final de este proceso la fábrica y los investigadores contaban con una muestra de 319 defectos clasificados.

Categoría de Clasificación	Criterio de Clasificación	Equivalencia con Taxonomía ODC
Error de Integración del Producto	La implementación de la interfaz no sigue la especificación de la misma Cambios de la interfaz no han sido impactados en capas inferiores del producto Cambios en la Data o metadata no ha sido impactada en capas inferiores del producto	Comunicación
Error en uso o configuración de controles de interfaz de usuario	Los controles no responden (“Freeze”) Falta el paginado en las DataGrids Data Grids no tiene Headers	Educación
..

Tabla 15 Muestra de la Taxonomía de Clasificación de Defectos

6.2.2 Modificación del sistema de Gestión de defectos

El sistema de gestión de defectos, y todo el sistema de métricas de la organización está basado en la integración de Microsoft Office SharePoint Portal con Microsoft Office InfoPath y Microsoft Office Access. Estos tres componentes en su conjunto brindan flexibilidad y acceso a todos los miembros de la organización.

Cada equipo de proyecto tiene su sitio dentro del portal. Todos los sitios están estandarizados, y cada sitio tiene entonces una lista de SharePoint donde se registran los defectos. Para mejorar la usabilidad, se utilizan los formularios electrónicos de InfoPath como FrontEnd a esta lista de SharePoint. Al actualizar los requerimientos de los formularios, y por lo tanto de las listas, todos los sitios se actualizan por batch semanalmente.

Finalmente, la integración de Access con Sharepoint, permite que todas las listas de todos los sitios sean integrados en una única Base de Datos Access para el análisis a nivel organizacional realizado por el área de calidad y métricas (ver Figura 18).

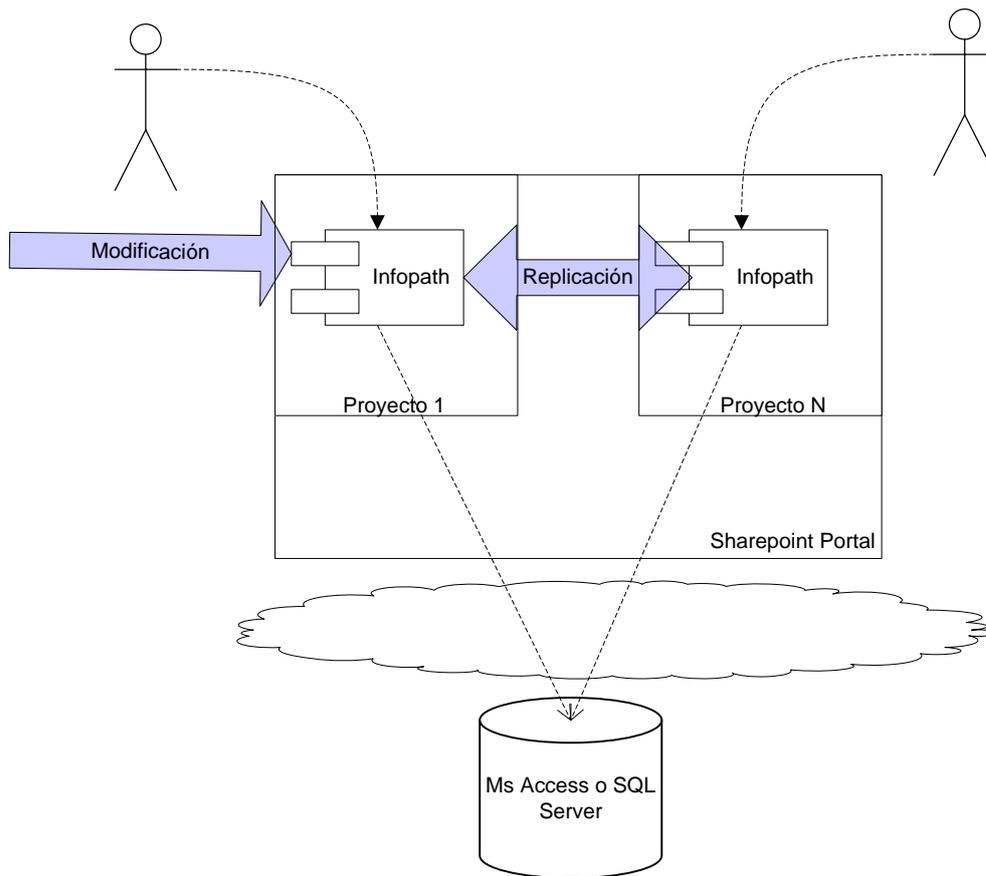


Figura 18 Arquitectura de componentes del sistema de gestión de defectos

La modificación para incluir la nueva taxonomía, supuso la modificación del formulario InfoPath y la consiguiente actualización en batch.

El esfuerzo para realizar estas modificaciones fue de 4 horas, y fue realizado por uno de los investigadores. De todas maneras el costo de la modificación se imputó a los costos en el cálculo del ROI (ver sección 6.5.5.1).

6.2.3 Estudio del despliegue de la clasificación

Como se describió en la sección 6.1, la fábrica de software había superado en el 2007 una evaluación CMMI nivel 3. En su implementación del modelo CMMI, la fábrica había asignado al departamento de entrenamiento la responsabilidad de brindar los servicios de capacitación para todos los proyectos de todas las líneas de producción.

Para cumplir con los requerimientos de este proceso, el área de capacitación tendrá que ser capaz de analizar los defectos clasificados de todos los proyectos. Es necesario entonces que la organización cuente con *una cierta confianza de que los defectos son clasificados en forma consistente en todos los proyectos.*

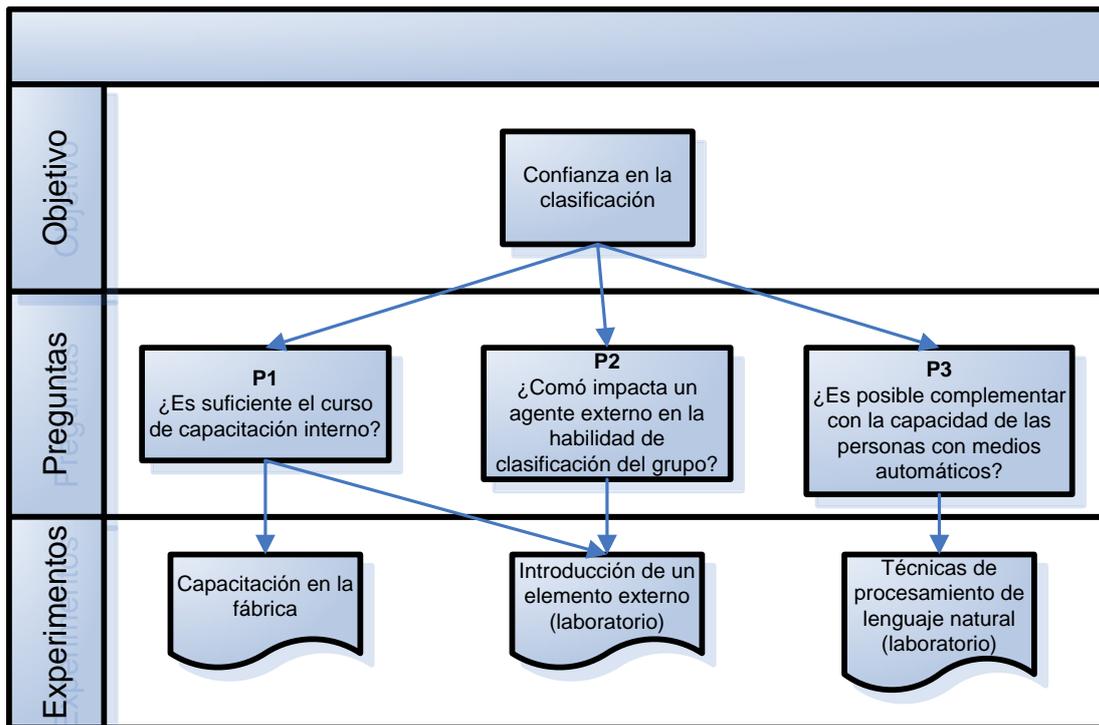


Figura 19 Preguntas de investigación para validar la capacidad de clasificación

La Figura 19 muestra la descomposición en objetivos de la estrategia de validación diseñada para investigar la confiabilidad de los recursos en la clasificación de defectos. Se diseñaron los siguientes experimentos:

- **P1:** En primer lugar se realizó una capacitación en la clasificación siguiendo los procesos de capacitación de la fábrica de software (sección 6.2.3.1). La evaluación consistió en entregar una muestra de defectos para clasificar, que los investigadores procesaron como se describe en la sección 6.2.3.2. El resultado de la evaluación se procesó en conjunto con el experimento asociado a **P2** en la sección 6.2.3.4.
- **P2:** En el laboratorio se repitió la evaluación con un sujeto que no tenía contacto con la fábrica de software (sección 6.2.3.3). El objetivo estuvo en determinar el impacto de su habilidad de clasificación al compararlo con un equipo entrenado en la Taxonomía (sección 6.2.3.4).
- **P3:** Otra experiencia consistió en determinar si es posible asistir a los clasificadores mediante algún método de procesamiento automático. Los resultados se describen en 6.2.3.5.

6.2.3.1 Descripción del curso de capacitación

Para desplegar la taxonomía se diseñó una capacitación sobre la taxonomía en el proyecto piloto³. La modalidad de la capacitación es de un curso de 4 horas más el tiempo de evaluación. Los cursos cortos de 4-6 horas preparados por los integrantes de la fábrica son representativos del esfuerzo estándar que esta organización invierte en iniciativas de entrenamiento. Este modelo de entrenamiento está basado en los resultados presentados por Motorola University [161]. La agenda del curso fue la siguiente:

- Presentación del objetivo de la introducción de la clasificación, donde se les dio a los participantes el contexto que motivaba la modificación a sus prácticas de trabajo. Les fue explicado el objetivo de la investigación y se les presentó en forma gráfica el proceso propuesto.
- Área de procesos de Análisis Causal; a grandes rasgos se les explicó el área del proceso de Análisis Causal y Resoluciones según lo entiende CMMI, previendo que la implantación del proceso iba a requerir un curso de seguimiento en el desarrollo del proceso de análisis causal. Como se adelantó en secciones anteriores y se verá en la sección 6.4.2, esto no fue finalmente realizado.
- Descripción de las 16 categorías. Para cada una de las 16 categorías se presentaron sus criterios de clasificación y se presentaron ejemplos de defectos ya clasificados, con el objetivo de que los asistentes entendieran y desarrollaran sus capacidades de clasificación.

6.2.3.2 Mecanismo de Evaluación de los asistentes al curso de capacitación

Hasta esta intervención de entrenamiento, el departamento de capacitación de la organización realizaba únicamente evaluaciones Nivel 1 y 2 del modelo de Kirkpatrick (ver sección 2.1.2) de sus cursos.

Como el objetivo del curso era comprobar la confiabilidad de la clasificación por parte de los distintos individuos de la organización, para esta investigación se realizó una evaluación distinta. A cada uno de los asistentes le fue entregado un conjunto de 31 defectos que deberían clasificar. A los resultados de las clasificaciones se les aplicaron dos funciones para el cálculo del *Acuerdo* de clasificadores (similar a los mecanismos utilizados por [162, 163]), llamadas Kappa.

La Kappa de Cohen[164] brinda una medida del grado de acuerdo que existe entre dos clasificadores. El objetivo de utilizar esta Kappa era para saber si existía diferencia en la habilidad de clasificar entre individuos entrenados e individuos no entrenados.

³ En el acuerdo llegado con la Software Factory en Diciembre de 2007, se incluyó una única línea de producto que en Enero-Febrero contaba con un único proyecto en Ejecución. Éste fue el proyecto para el que se armó la capacitación que se describe.

Durante el primer trimestre del 2008, la organización decidió extender el alcance del proceso de clasificación a sus otras dos líneas de producto. El experimento que se describe en esta sección abarca únicamente el grupo de entrenamiento del proyecto Piloto original.

La Kappa de Fleiss[165] brinda una medida del grado de acuerdo que existe ente varios clasificadores independientes. El objetivo de utilizar esta Kappa era para determinar si el grupo como tal mejoraba su habilidad de clasificar luego de ser entrenado.

La Tabla 16 presenta los rangos de interpretación de los valores de las Kappas. Esta interpretación fue propuesta por [166], y aunque el autor indica que el valor de los intervalos es subjetivo; se establece que el objetivo será obtener valores en el rango “Nivel de Acuerdo Satisfactorio”.

Valor de Kappa	Relevancia
< 0	No hay acuerdo
0,00 – 0,20	Poco nivel de acuerdo
0,21 – 0,40	Nivel de acuerdo bajo
0,41 – 0,60	Nivel de acuerdo Moderado
0,61 – 0,80	Nivel de acuerdo satisfactorio
0,81 – 1	Acuerdo casi completo

Tabla 16 Valores de Relevancia de las Kappas

Las siguientes dos secciones describen en detalle este experimento y sus resultados.

6.2.3.3 Descripción de los sujetos del experimento

Los asistentes a este primer evento de capacitación en la clasificación pertenecían todos al proyecto piloto en que se iba a desplegar el proceso. El equipo estaba integrado por un programador senior y dos programadores junior. Estos 3 individuos fueron invitados al curso. Además de estos 3 participantes se agregaron dos sujetos más para completar el experimento.

Se asignó a la capacitación a uno de los analistas de software que integraron el grupo de trabajo que diseñó la taxonomía (Taxonomy Working Group – ver sección 6.2.1). Este individuo será identificado con la letra **E** por considerarlo el *Experto* en la taxonomía de la organización. Se asumió que como él estuvo involucrado en el diseño de la taxonomía sus respuestas se deberían acercar a las respuestas correctas.

Además del Experto, se incluyó en el experimento a una figura que representara un elemento externo de la organización. El **outsider** es el un individuo que no tiene contacto con la organización. El Outsider, a quien se hará referencia con la letra **O**, viene a representar el índice de rotación de la fabrica (ver sección 6.1.1). Para la organización resultaba importante saber si su índice de rotación influiría en sus capacidades de clasificación (en [167] un estudio sobre el efecto de la capacitación en la rotación del personal, por lo que se podría inferir que este experimento será de interés también fuera de la organización).

El papel del outsider estuvo representado por uno de los Becarios del Laboratorio de Lenguajes y Sistemas de Información de la Universidad Politécnica de Madrid. A él se le brindó un resumen de la organización, similar al curso de inducción que esta fabrica realizaba en el 2007, y se le entregaron los mismos 31 defectos, y la taxonomía con sus categorías y criterios, la misma información que tendrían disponible en el sistema de gestión de defectos. No se le dio acceso a los materiales del curso de capacitación en la taxonomía descritos en la sección anterior.

Los resultados de esta evaluación se analizaron con los resultados de los otros sujetos. A los efectos de comunicar los resultados, al programador senior será identificado como **C1**, mientras que los dos programadores serán identificados como **C2** y **C3** respectivamente.

Como se mencionó en la sección 6.1.1, este conjunto de individuos puede tomarse como una representación de los integrantes típicos de un proyecto de desarrollo de la fábrica.

6.2.3.4 Resultados del experimento de clasificación

En el estudio del nivel de consistencia de la clasificación no es importante establecer la corrección de cada uno de los defectos clasificados, sino que el foco está en establecer el acuerdo para reafirmar la confianza en la clasificación del grupo. La Tabla 17 presenta los resultados de aparejar cada una de las respuestas de los sujetos de estudio.

Sujetos	Kappa de Cohen	Sujetos	Kappa de Cohen	Sujetos	Kappa de Cohen
E – C1	0,73	C1 – C2	0,62	C2 – C3	0,52
E – C2	0,55	C1 – C3	0,47	C2 – O	0,37
E – C3	0,52	C1 – O	0,36	C3 – O	0,30
E - O	0,34				

Tabla 17 Resultados de la Kappa de Cohen entre los sujetos

Al comparar estos resultados con la Tabla 16 se observa que sólo una pareja de individuos ha obtenido valores de relevancia dentro del rango esperado (E-C1). El otro dato interesante es que cualquier emparejamiento que incluya al outsider presenta resultados en el rango del 30%. Este resultado es consistente con los resultados de [166] que indicaba que el “azar” o “sentido común” debía llevar los resultados de acuerdo hasta cerca del 30%.

Por último, los emparejamientos que incluyen a los sujetos entrenados, superan el rango del Outsider. Esta mejora en performance está vinculada a los resultados del entrenamiento, pudiendo hasta llegar a afirmar que con un entrenamiento dirigido de 4 horas de esfuerzo se han mejorado los resultados del “sentido común” en un 10%. *Verificando entonces el impacto del entrenamiento en las capacidades de clasificación de los individuos.*

La Tabla 18 presenta los resultados de aplicar la Kappa de Fleiss a los “dos equipos”, uno que incluye al outsider, y el otro considerando únicamente los resultados de los individuos que pertenecían a la organización.

Sujetos	Kappa de Fleiss
E-C1-C2-C3	0,53
E-C1-C2-C3-O	0,44

Tabla 18 Resultados de aplicar la Kappa de Fleiss

Como se observa, existe una diferencia de alrededor de un 10% en el valor de la Kappa de Fleiss entre el equipo que integra al Outsider y el otro. Sin embargo, ambos valores caen dentro del mismo rango de acuerdo descrito en la Tabla 15.

Por lo que podemos concluir que *la inclusión de elementos no entrenados en la taxonomía de clasificación no afecta significativamente las capacidades de clasificación de los equipos*. O lo que es lo mismo, una organización puede confiar en las clasificaciones de un equipo entrenado en su taxonomía de clasificación y no es imperativo que los nuevos recursos sean entrenados en ella antes de que estos sean asignados a tareas productivas.

6.2.3.5 Evaluación de la capacidad de clasificación en las organizaciones

En la sección 6.2.2, se mencionó que con el objetivo de proteger la información de la organización se estableció una equivalencia entre la taxonomía de la organización y la taxonomía de ODC. Además, en el 2008 la organización realizó un mantenimiento de su taxonomía, reduciendo el número de categorías de 16 a 12.

Para esta investigación, el esfuerzo resultó en la actualización de la función de equivalencia para poder analizar los datos de los proyectos del 2008. Pero para la organización significó un costo mayor. Por un lado, está el esfuerzo invertido por parte de la organización en el desarrollo y mantenimiento de la clasificación de defectos, pero por otro lado, cada cambio en su taxonomía implica una pérdida de validez de su histórico de defectos.

Con esta iniciativa, en conjunto con el phd Vasile Rus de la Universidad de Memphis se comenzó a explorar alternativas para realizar la clasificación de defectos asistidos por las computadoras. El equipo de investigación de la Universidad de Memphis ya había trabajado en el problema de agrupar defectos sin la asistencia del humano [168, 169]. Y le fue propuesto aplicar sus resultados al problema de la confianza de los clasificadores.

Como se menciona en [168] la Universidad de Memphis contaba con un mecanismo para procesar reportes de defectos a partir del Procesamiento de Lenguaje Natural (NLP por sus siglas en inglés) y modelos de recuperación de la información (IR por sus siglas en inglés). Esta técnica se estaba utilizando para agrupar reportes de defectos tomados de la herramienta de gestión de defectos del navegador *Mozilla Firefox*. A este agrupamiento se le estaba denominando Cluster, donde la acción de realizar estos Clusters (Clustering) se

define como la clasificación sin supervisión humana de patrones en grupos (clusters) basado en una medida de similitud[170].

El objetivo de este trabajo consiste en suministrarle al computador una muestra clasificada de defectos para luego comparar los resultados de su clasificación con la clasificación del equipo de personas de la organización.

Los resultados de este experimento fueron presentados en [37]. Para realizar la clasificación se tomó como entrada para el mecanismo de la Universidad de Memphis las combinaciones de *Título* y *Descripción* de los reportes de defectos de la organización. El primer desafío consistió en adaptar la herramienta para que ésta procese lenguaje natural del idioma Castellano, y la exportación desde el sistema de gestión de defectos de la organización a un formato de comas para que estos puedan ser procesados por la herramienta.

En la Tabla 19, se presentan los resultados obtenidos; se utilizo una muestra de 236 defectos que fueron corridos contra dos algoritmos de clasificación, y validados por dos métodos de validación.

Campo de entrada	Algoritmo de Clasificación (Significance / Kappa Relevancia)		Método de Validación
	Naïve Bayes	Decision Trees	
Título	0.58/0.47	0.94/0.92	10x Fold
Descripción	0.48/0.31	0.94/0.92	
Título + Descripción	0.48/0.34	0.94/0.92	
Título	0.61/0.50	0.92/0.91	66-34% Split
Descripción	0.44/0.28	0.81/0.77	
Título + Descripción	0.44/0.25	0.81/0.80	

Tabla 19 Resultados Clasificación Automática

La tabla muestra como el método por Árboles de Decisión consigue valores de relevancia y de entendimiento de clasificadores que se encuentran en los primeros rangos de relevancia de la Tabla 16, mejorando la relevancia tanto del algoritmo de Bayes como el de los clasificadores humanos.

Los resultados de esta experiencia muestran que las organizaciones tienen alternativas para asegurar la confiabilidad de las clasificaciones de defectos en sus proyectos. Al comparar este resultado con los resultados del experimento de la sección anterior, se

puede afirmar que *la baja en relevancia introducida por el Outsider puede suplirse con creces por un mecanismo asistido por el computador.*

Este experimento abre nuevas líneas de investigación que serán discutidas en la sección 6.7.

6.3 Análisis de resultados del piloto

Previo al despliegue del proceso se negoció con la organización la identificación de un proyecto piloto para desplegar el proceso. Para conseguir esto se realizó una simulación del comportamiento del proceso para el cálculo del ROI en la organización.

1. Seleccionar Muestra de Defectos
2. Establecer equivalencia con ODC
3. Establecer la línea base de comportamiento
4. Estimar costos
5. Proyectar Beneficios
6. Calcular ROI

Como muestra representativa de la de defectos de la organización se utilizaron los 378 defectos clasificados. Los 319 clasificados durante el proceso de diseño de la taxonomía de la organización (sección 6.2.1) y otros 59 clasificados por la organización luego de cerrada la instancia de capacitación.

Al aplicar la equivalencia entre la taxonomía de la organización con la taxonomía de ODC se ve que para esta organización el **52%** (ver Figura 20) de los defectos estaban clasificados entre las categorías de *Educación* o *Transcripción*.

En el punto 5 de la sección Planteamiento, se estableció que el objetivo de la propuesta consiste en que los defectos cuyo origen son limitaciones en las habilidades o conocimientos de los recursos son más efectivamente atacados con iniciativas de capacitación que con iniciativas de mejora de procesos. La línea Base de defectos quedará para esta simulación determinada por:

- el volumen de defectos y la distribución de defectos de la organización.
- el costo medio de un defecto según el histórico de la fábrica (sección 4).

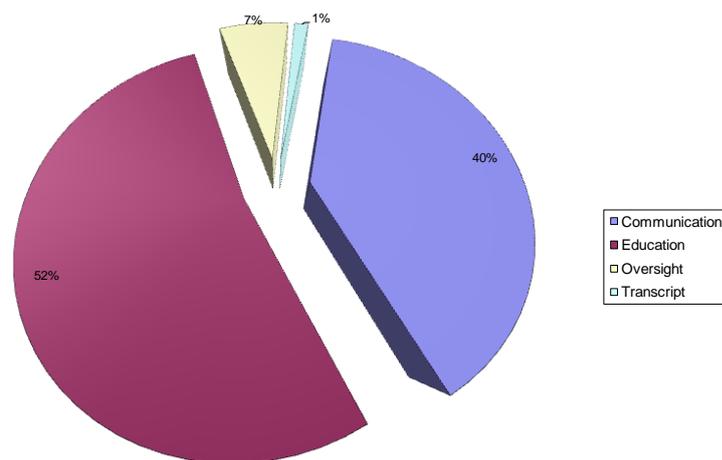


Figura 20 Distribución de defectos del 2007

Los costos asociados a la propuesta incluyen costos fijos y variables. Al momento de realizar este cálculo ya se habían ejecutado las tareas asociadas a los costos fijos asociados al despliegue del proceso. La Tabla 20 presenta los factores de costo tomados en cuenta para el cálculo del ROI.

Tipo de Costo	Descripción del Costo	Notas de Implementación en la Organización
Establecer el Proceso	Desarrollo y Despliegue de la Taxonomía (C_{TM})	Costo en valor hora normalizado de los recursos de la organización que trabajaron en las sesiones de diseño de la taxonomía.
	Modificar el Sistema de Gestión de Defectos (C_{Tool})	La tecnología que la organización utilizaba para el sistema de gestión de defectos permite una rápida actualización, implementación y puesta en producción desde el servidor a todos los sitios del proyecto. Esto se tradujo en un beneficio de reducción de costos que favorece los resultados de la propuesta en esta organización.
	Desarrollar la capacidad de Análisis Causal (C_{DCAR})	Costo del Esfuerzo en Valor Hora Normalizado
Ejecutar el Proceso	Esfuerzo en Reuniones de Análisis Causal (C_{CAM})	Costo en Valor Hora Normalizado del esfuerzo invertido en sesiones de análisis causal (estimado de la literatura).
	Preparación y ejecución de la intervención de entrenamiento (C_T)	A diferencia de la Tabla 10, donde esta categoría aparece dividida, en esta organización el área de entrenamiento no distingue preparación de ejecución. Todos los

costos (incluidos los de asistencia), son cargados bajo el mismo ítem en el sistema de seguimiento de la organización.

Tabla 20 Factores de Costo

$$(5) \quad \text{Costo} = \text{Costo}_{\text{Establecer Proceso}} + \text{Costo}_{\text{Ejecutar Proceso}}$$

Para probar la efectividad de la propuesta a la organización se realizó una proyección de los Beneficios Obtenidos. En la sección 2.1.3 se describe que la propuesta consistía en calcular los Beneficios en función de la Reducción de Defectos observada luego de implementada la modificación. El Cálculo de ROI lo realizaremos a partir de la fórmula clásica presentada en la ecuación (1).

Los Beneficios fueron estimados en función de la distribución de los defectos observada multiplicada por el costo de un defecto tomado de la base de datos de la organización. Para mantener la estimación de beneficios lo más objetiva posible, para estimar la reducción de defectos se utilizó la eficacia de los cursos histórica de entrenamiento de la organización. Este dato fue tomado de la base de datos de capacitación de la organización.

$$(6) \quad \text{Beneficios} = (\text{Defectos}_{\text{Esperados}} - \text{Defectos}_{\text{Observados}}) * \text{Costos}_{\text{Defecto}}$$

Por lo tanto, para el proyecto piloto se sustituye el factor entre paréntesis por la estimación de los defectos que esta propuesta puede reducir.

$$(7) \quad \text{Beneficios} = (\text{Defectos}_{\text{registrados}} * \text{Efectividad}_{\text{Entrenamiento}}) * \text{Costo}_{\text{Defecto}}$$

Factores	Valores (Horas Normalizadas Programador)
Costos	Establecer el Proceso = 30
	Ejecutar el Proceso = 231
Beneficios	Efectividad Capacitación = 77%
	Costo Medio Defectos = 4.63
	Defectos Muestra = 378
	Porcentaje de Defectos de Capacitación = 52%

Tabla 21 Factores de cálculo del ROI

Por lo tanto, sustituyendo en la ecuación (1) por los valores de la fábrica se obtiene un ROI de **173%** (Tabla 21).

$$ROI = \frac{(Beneficios - Costos)}{Costos} = \frac{(378 * 0.77 * 0.53 * 4.63) - (231 + 30)}{231 + 30} * 100 = 173\%$$

Con estos resultados se obtuvo la aprobación de la gerencia de la fábrica para continuar con el despliegue del proceso.

6.4 Despliegue del proceso propuesto

Esta sección describe la experiencia de desplegar el proceso propuesto durante el primer trimestre del año 2008.

6.4.1 Selección de los proyectos

Durante el primer trimestre del año 2008, la fábrica de software extendió el alcance del proceso a todos los proyectos de la fábrica.

Al proyecto seleccionado originalmente como piloto, se denominará Proyecto1⁴. Luego de un análisis primario de los datos por parte de los investigadores se decidió utilizar una muestra de tres proyectos. La decisión de descartar los otros proyectos se debió principalmente a problemas en la calidad de los datos reportados por los otros proyectos.

Como línea base de performance se utilizó una muestra de cuatro proyectos antiguos pertenecientes al año 2007.

La Tabla 22 muestra el perfil cuantitativo de cada uno de los proyectos involucrados.

⁴ El nombre de los proyectos fue reemplazado para proteger la identidad de los clientes de la organización.

Proyecto	Esfuerzo (horas totales normalizadas ⁵)	Esfuerzo en Testing (horas totales normalizadas)	Esfuerzo en Análisis Causal (horas totales normalizadas)	Defectos reportados	No conformidades
Proyecto1	5750	119	12	115	4
Proyecto2	922	40	5	53	33
Proyecto3	4491	87	11	54	15
2007_Proyecto1	4744	353	0	85	21
2007_Proyecto2	9136	296	0	125	22
2007_Proyecto3	21179	928	0	431	26
2007_Proyecto4	13983	722	0	138	9

Tabla 22 Resumen de Métricas de los proyectos

Como los datos tienen que ser observados en su contexto[145], también es importante destacar algunas de las características de cada uno de estos proyectos.

⁵ Para proteger la información sensible de la organización, decidimos medir el esfuerzo de los proyectos en “horas totales normalizadas”, lo que significa el equivalente de horas de programador si el costo de todos los roles del proyecto fuera de un programador.

Proyecto	Línea de Producto	Comentarios
Proyecto1	Desarrollo .NET	Proyecto típico de la fábrica con un analista, y tres desarrolladores. Se compraron componentes gráficos de terceros para cubrir una funcionalidad
Proyecto2	Customizacion Microsoft Office Sharepoint	No fue un proyecto típico de esta línea de desarrollo ya se tuvo que recurrir a una librería de componentes gráficos comprada a una tercera organización
Proyecto3	Desarrollo .NET	Proyecto típico de la línea de desarrollo
2007_Proyecto1	Customizacion Microsoft Office Sharepoint	Proyecto Típico de la línea de producción. En este proyecto se experimentaron con automatizaciones para acelerar el desarrollo del portal.
2007_Proyecto2	Desarrollo .NET	Proyecto típico de la línea de desarrollo
2007_Proyecto3	Customizacion Microsoft Office Sharepoint	Estimado como un proyecto típico, se identificaron problemas de integración entre algunas tecnologías que requirieron un sobreesfuerzo.
2007_Proyecto4	Customizaciones de Productos propietarios	Proyecto típico de la línea de desarrollo

Tabla 23 Resumen cualitativo de los proyectos

6.4.2 Despliegue del proceso de análisis causal

La fábrica de software definió las políticas para la realización de las sesiones de análisis causal. Y los jefes de proyecto incluyeron las tareas y la asignación de recursos en sus cronogramas.

La fábrica de software no preestableció ninguna de las herramientas relevadas (sección 2.3.3) como soporte al proceso de análisis causal. También se realizó una única sesión de entrenamiento con los recursos de los proyectos que se habían comprometido a seguir el proceso⁶.

En esa sesión se les presentó el objetivo del proceso y se les planteo una única pregunta: “Cuando se dispongan a realizar estas sesiones, traten de contestar. “¿Qué conocimiento o habilidad hubiesen necesitado para evitar cometer esos errores?”. Los miembros del los

⁶ Este entrenamiento incluyo al Proyecto1 y a dos proyectos más que descartamos de nuestra muestra de datos, por lo tanto, los integrantes de los proyectos que aquí aparecen como Proyecto2 y Proyecto3, no recibieron este entrenamiento.

equipos de desarrollo que implementaron la propuesta se auto-organizaron en sesiones de brainstorming.

Como resultado, los proyectos de la muestra:

- Registraron el esfuerzo invertido en análisis causal
- Los programadores ingresaron las propuestas en el sistema de gestión de defectos.
- En la fábrica de software se definió un proceso formal de análisis causal de defectos.

6.4.3 Adaptaciones al proceso de extracción de las iniciativas de entrenamiento

El proceso propuesto propone que las iniciativas de intervención del entrenamiento sean desarrolladas a partir de los defectos reportados y de las sugerencias de entrenamiento que surgen de las sesiones de análisis causal (ver Figura 21).

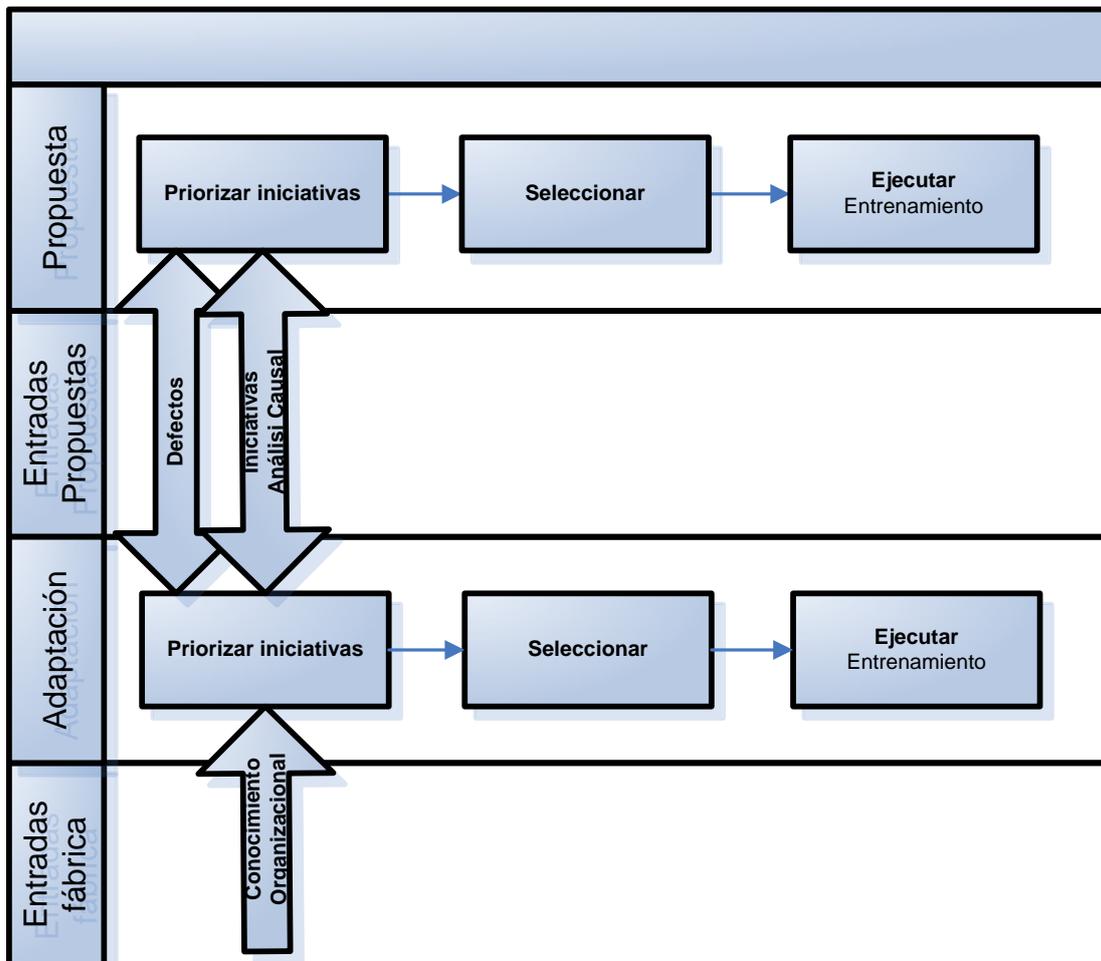


Figura 21 Proceso de extracción de las iniciativas de capacitación

Lo cierto es que la organización utilizó un tercer factor de ponderación que en la propuesta no está tenido en cuenta y es el conocimiento de la organización que los recursos del departamento de entrenamiento tienen. Esto significa que por más que exista una lista de los pedidos de entrenamiento aportados por los desarrolladores (Tabla 24), el departamento de entrenamiento hizo primar su conocimiento cualitativo de la organización por sobre los datos cuantitativos que tenía del sistema de gestión de defectos.

Sugerencias de Entrenamiento realizadas por los integrantes de la fabrica	Ratio
Herramientas para trabajo en equipo, portal de comunicación. Como incluir al cliente	0.81
Revisiones de Pares (de código y de requerimientos. Checklist, Algunas faltan, Como seguirlas?)	0.79
Frameworks de Desarrollo. Nuevos avances (Frameworks 3.5 WPF)	0.78
Validación y Verificación de Requerimientos, Como integrar al cliente	0.76
Herramientas de Gestión de cambios (VSTF, MOSS)	0.72
Desarrollo de casos de uso (a partir de requerimientos, características, cómo documentarlos)	0.72
Desarrollo de Casos de Prueba (Criterios de Bordas, Límites, Clases de equivalencia)	0.71
Confección de productos de trabajo de requerimientos (matriz, EsRe, Casos de Uso, Diagramas de flujo de negocio)	0.70
Criterios para obtener "buenos requerimientos"	0.67
Entrenamiento en ingeniería de software (ciclo de vida, administración de riesgos, etc)	0.66

Tabla 24 Top 10 sugerencias de entrenamiento

Por otro lado, el proceso propuesto propone una visión *proyecto a proyecto*. Sin embargo el departamento de entrenamiento tomó *una visión común de estado* de la fábrica y realizó el entrenamiento para todos los recursos de la misma. Esta decisión amortiza los costos de preparación del entrenamiento entre los 3 proyectos que integran la muestra.

6.4.4 Entrenamiento realizado y resultados obtenidos

Finalmente el objetivo del entrenamiento estuvo en familiarizar a los recursos con las herramientas de gestión de cambio y trazabilidad (relacionados con los puntos 4 5 y 6 de

la Tabla 24). El temario del entrenamiento según figuró en los registrós de capacitación de la organización fue el siguiente:

- Trazabilidad y Gestión de Requerimientos. Incluyendo una revisión del proceso en su estado a la fecha del entrenamiento y una capacitación en las herramientas (Team Foundation Source Control, Team System y Microsoft Office Sharepoint). La audiencia fue Analistas y Programadores.
- Gestión de Requerimientos: Incluyendo Ciclo de Vida, Relevamiento, y documentación de los requerimientos así como características. Cómo documentarlos y comunicarlos a los desarrolladores. La audiencia fue de Analistas y Programadores.
- Revisiones de Pares de Código. Pasos de una revisión de pares, objetivos y mejores prácticas.

El resultado de la evaluación de conocimientos (evaluación nivel 1) tuvo un promedio de aceptación del 54%, y de los trece asistentes siete superaron el mínimo de nota exigido por la organización (evaluación nivel 2).

6.5 Evaluación del retorno de inversión en los proyectos del año 2008

Esta sección describe las actividades realizadas para establecer el objetivo principal de este trabajo:

- **El proceso propuesto permite comunicar los resultados en términos de Retorno de Inversión teniendo en cuenta la variación natural de los procesos productivos**

Para determinarlo se aplicaron los pasos para calcular el retorno de inversión descritos en la sección 5.2.4.

1. Determinar la muestra histórica
2. Recolectar Datos
3. Evaluar disponibilidad de datos
4. Establecer nuevos límites de control
5. Calcular ROI
6. Evaluar los resultados

6.5.1 Determinar la muestra histórica

Este primer paso consiste en establecer las expectativas de defectos de los proyectos de la fábrica. La muestra de proyectos serán los cuatro proyectos del 2007 descritos en la sección 6.4.1.

6.5.2 Recolectar Datos

Atendiendo a las adaptaciones del proceso que realizó la fábrica (sección 6.4.3), se decidió realizar la evaluación a nivel de la fábrica y no a nivel de interacción por iteración como se planteó en la sección 5.1.

La implementación y ejecución del proceso propuesto dejó los datos presentados en la Tabla 25. Una vez recolectados estos datos, el siguiente paso consistió en establecer los límites de control para fijar la línea base de performance de los proyectos.

Proyecto	Número Total de Defectos	Defectos “Educación”	Porcentaje de Educación
Proyecto1_2007	86	47	0,540
Proyecto2_2007	125	71	0,568
Proyecto3_2007	431	191	0,443
Proyecto4_2007	174	107	0,615

Tabla 25 Datos de defectos de los proyectos de la línea base

Para establecer los límites de control se aplicó la técnica que se describe en [145].

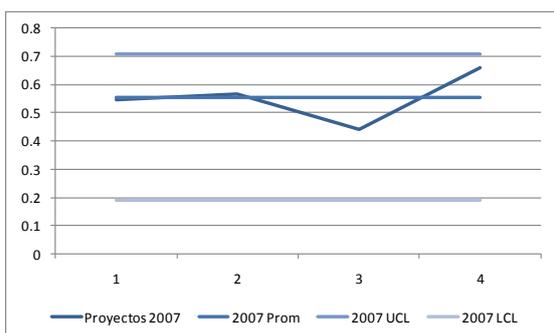


Figura 22 Límites de Control Defectos por Hora de Test Proyectos 2007

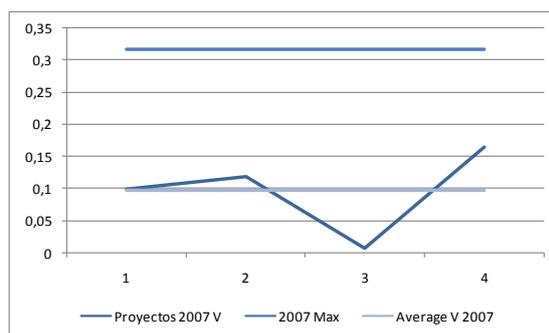


Figura 23 Máximo Rango de Variación. Defecto por hora de Test Proyectos 2007

6.5.3 Evaluar disponibilidad de datos

Como se mencionó en la sección 6.4.1, luego de evaluar los datos de los proyectos del 2008 de la fábrica se decidió utilizar tres proyectos como muestra de los proyectos que implementaron el proceso.

Los criterios para seleccionar a estos proyectos correspondieron a una evaluación de la calidad de datos. Se estableció el requisito de que los proyectos seleccionados deberían tener:

- Datos de esfuerzo en las sesiones de análisis causal registrado
- Informe del equipo de testing por cada iteración planificada
- Reportes y registros de las No Conformidades de PPQA.

Por lo tanto, atendiendo a lo descrito en la sección 5.2.4, se necesitan al menos tres puntos para proceder a calcular el retorno de inversión aplicando la adaptación a la fórmula del Retorno de Inversión propuesta en este trabajo.

6.5.4 Establecer nuevos límites de control

La Tabla 22 presenta los datos recolectados de los defectos de los proyectos del 2008.

Proyecto	Número Total de Defectos	Defectos “Educación”	Porcentaje de Educación
Proyecto1	115	48	0,42
Proyecto2	53	7	0,13
Proyecto3	54	23	0,43

Tabla 26 Datos de defectos de los proyectos del 2008

Para establecer el comportamiento de los proyectos del 2008, se aplicarán las mismas técnicas que con los proyectos de la línea base de performance. El resultado se presenta en las siguientes figuras.

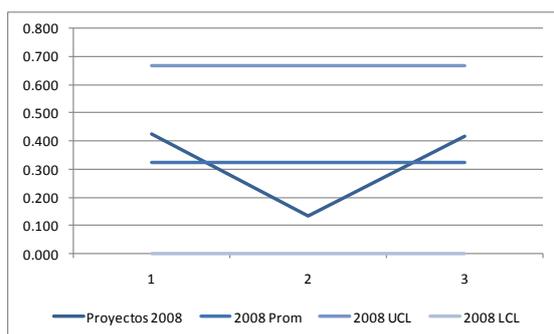


Figura 24 Límites de Control Defectos por Hora de Test proyectos 2008

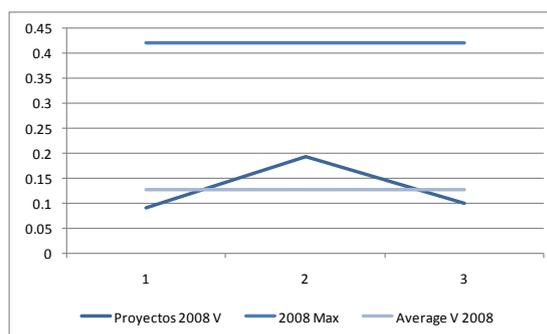


Figura 25 Máximo Rango de Variación. Defecto por hora de Test Proyectos 2008

6.5.5 Calcular ROI

El paso final consiste en aplicar la adaptación a la fórmula del ROI presentada en la sección 5.2.4.

- **Al mejor caso**, que resulta de comparar el límite de control inferior de los proyectos del 2007 (Figura 22), con el límite de control superior de los proyectos del 2008 (Figura 24).
- **Al peor caso**, que resulta de comparar el límite de control superior de los proyectos del 2007 (Figura 22), con el límite de control inferior de los proyectos del 2008 (Figura 24).
- **Y al observado**, que resulta de comparar los valores promedios de las muestras del 2007 y del 2008.

6.5.5.1 Cálculo de los Costos

Como se mencionó en la sección 5.2.4, los costos para la propuesta dependerán de la temporalidad asignada al cálculo de Retorno de inversión. Para esta evaluación se incluirán todos los costos, incluyendo:

- costos de ejecución del plan de implantación,
- costos de ejecución del proceso

Por lo tanto, para el cálculo final del ROI los factores de costo serán⁷:

- $Costo_{Taxonomia} = 28,8 \text{ Hnp}$
- $Costo_{AnalisisCausal} = 27,9 \text{ Hnp}$
- $Costo_{ModificacionHerramienta} = 4,8 \text{ Hnp}$
- $Costo_{Capacitacion} = 73,2 \text{ Hnp}$

Entonces, $Costo = (28,8 + 4,8 + 73,8 + 27,9) = 134,7 \text{ Hnp}$

6.5.5.2 Cálculo de Beneficios

Para calcular los beneficios se utilizarán los valores del observado y del Límite de control superior.

Los factores para el cálculo del ROI son:

- $Defectos_{Esperados}$; Tomados de los límites de control de la Figura 22, normalizado con el número de defectos observados en el 2008.
 - Numero de Defectos en el Peor Caso = 46 defectos.
 - Numero de Defectos en el Mejor Caso = 168 defectos.
- $Defectos_{Observados}$
 - Para el mejor caso equivale a **0**, que es el caso en que todos han sido eliminados, se asignó la cota en 1 ya que no tiene sentido valores negativos.
 - El real que es la cuenta de los defectos observados en la muestra de los proyectos de 2008 (78 defectos)
 - Para el peor caso, equivale al límite de control superior multiplicado por el número de defectos del 2008 (159 defectos).
- Costo Defecto. El costo de un defecto para la organización varió en los dos periodos estudiados:
 - Costo Medio Defecto 2007 = 4,63 Hnp
 - Costo Medio Defecto 2008 = 2,85 Hnp

Por lo tanto:

$$Beneficios_{MejorCaso} = (168 * 4,63) - (0 * 2,85) = 779 \text{ Hnp}$$

$$Beneficios_{CasoObservado} = (107 * 4,63) - (77 * 2,85) = 276 \text{ Hnp}$$

⁷ Todos los valores están en Horas normalizadas de programador (Hnp)

$$Beneficios_{PeorCaso} = (46 * 4,63) - (159 * 2,85) = -240Hnp$$

6.5.5.3 Evaluación del Retorno de inversión de la propuesta

El ROI observado para la propuesta fue el siguiente:

Se obtuvieron los siguientes Valores:

- Mejor Caso: $ROI_{MejorCaso} = \frac{(779 - 134,7)}{134,7} * 100 = 481\%$
- Caso Observado: $ROI_{CasoObservado} = \frac{(275 - 134,7)}{134,7} * 100 = 1,04\%$
- Peor Caso: $ROI_{PeorCaso} = \frac{(-241 - 134,7)}{134,7} * 100 = -280\%$

Todos los casos representan el retorno de inversión que se obtendría para el despliegue y el primer ciclo de proyectos (en este caso 6 meses).

No todos los casos representan un Retorno de Inversión mayor al 100%. Y en comparación con los resultados de la Proyección realizada en el año 2007 (173%, sección 6.3), el ROI en el mejor caso es inferior. En el Caso Observado, la propuesta presenta un ROI que es marginalmente superior al 100%.

Para identificar los factores que incidieron en este rango de variación se aplicaron técnicas de análisis causal. En la siguiente figura se presenta un diagrama de espina de pescado utilizado para entender las causas que influyeron en los resultados obtenidos:

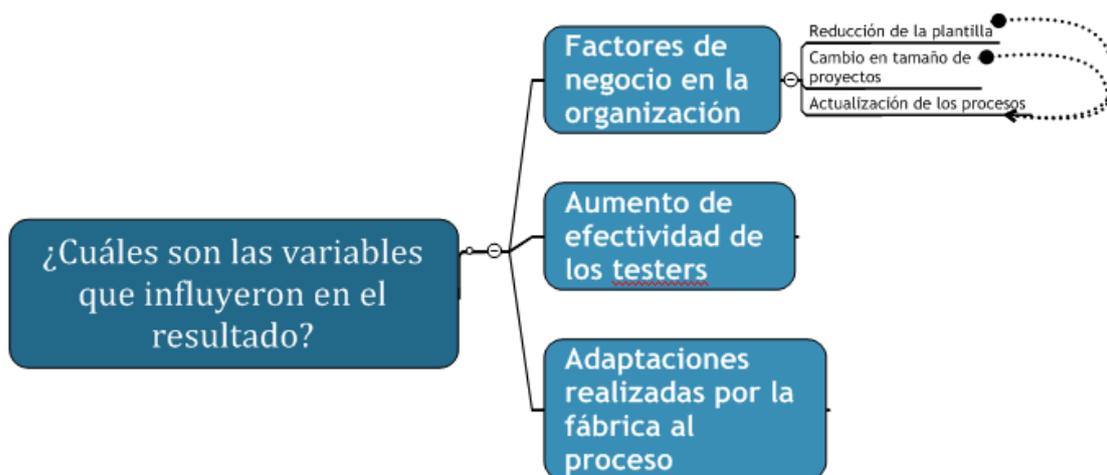


Figura 26 Variables que influyeron en los resultados

1. Factores de negocio de la organización

En un primer análisis, esta es la variable que incide con mayor peso en los resultados obtenidos.

- Reducción de la plantilla
La reducción de la plantilla es resultado de las modificaciones en la estructura de la fábrica. Los recursos que permanecieron en la organización comenzaron a adoptar nuevos roles para los que quizás necesitaran más experiencia o capacitación.
- Cambio en el tamaño de los proyectos
La reducción de tamaño de los proyectos en la fábrica impacta sobre la composición de los equipos. Favoreciendo los cambios de roles que trajo aparejada la reducción de la plantilla. No es posible determinar una relación causal entre ellas.
La reducción en tamaño disminuye también el número de defectos tanto en el código, como detectados por los testers. Esto influye a la baja en los beneficios.
- Actualización de los procesos
Los procesos deberían haberse modificado para adaptarse a la nueva realidad influenciada por los dos puntos anteriores. Sin embargo, no parece haber sido el caso, en el último trimestre de 2007 el equipo de mejora de proceso registra cerca de 100 solicitudes de mejora sin procesar. Para el primer semestre de 2008, el dueño de los procesos se pasó desde el grupo de mejora a toda la fábrica, con el objetivo de acelerar la adaptación de los mismos.

2. Aumento de la efectividad de los testers

El aumento de la efectividad de los testers, también impacta hacia la baja en nuestra propuesta, ya que reduce el Costo medio por defecto en el año 2008. Sin embargo, la diferencia no es significativa, y esta sección presentó los resultados del cálculo teniendo en cuenta esta variación.

3. Adaptaciones realizadas por la fábrica al proyecto

Las adaptaciones que la fábrica realizó al proyecto, tenderían a impactar hacia los resultados positivos del ROI, ya que al tomar los elementos comunes del despliegue y tratarlos al nivel de organización, los costos fijos se amortizan entre los distintos proyectos.

Por otro lado, no es posible saber cómo impactó la decisión de la organización de incluir un nuevo factor de ponderación a la hora de seleccionar el tema de la intervención del entrenamiento.

En conclusión, el mayor factor de resonancia en esta experiencia se debió a la variación en los factores de negocio. Como se vio en cada una de las gráficas presentadas (secciones 6.5.2 y 6.5.4), en cada uno de los indicadores utilizados, se observaron fluctuaciones en los rangos de variación de los datos. Esto se debió a que los procesos de la fábrica estaban preparados para otro tipo de proyecto, otras realidades de negocio, a los que la fábrica se enfrentó en el año 2008. Ese proceso de ajuste tendió a incrementar los defectos inyectados en función del volumen de esfuerzo producido.

El diagrama de espina de pescado y la explicación presentada en esta sección, son conclusiones cualitativas de los resultados obtenidos. En las siguientes secciones presentamos revisiones de algunos indicadores estudiados con el objetivo de confirmar cuantitativamente estas afirmaciones.

6.5.5.3.1 Distribución de defectos

El primer punto a revisar atañe a la diferencia en las distribuciones de defectos de la línea base y de los proyectos que aplicaron el proceso propuesto.

En la sección 6.3 se argumentó que esta fábrica de software podía obtener resultados del orden del 173% del retorno de inversión, debido a que distribución de defectos favorecía la reducción de defectos por métodos de entrenamiento.

Al analizar la distribución de defectos del 2008 el objetivo era descartar que la propuesta hubiese incrementado los defectos totales inyectados por la fábrica.

La tabla y la gráfica siguiente muestran la evolución de la distribución de los defectos en los dos años observados.

Categoría	Defectos 2007	año	Porcentaje Defectos 2007	año	Defectos 2008	año	Porcentaje Defectos 2008	año
Communication		148		40%		83		35%
Oversight		25		7%		68		29%
Education		196		52%		83		35%
Transcript		5		1%		4		2%

Tabla 27 Distribución de los defectos

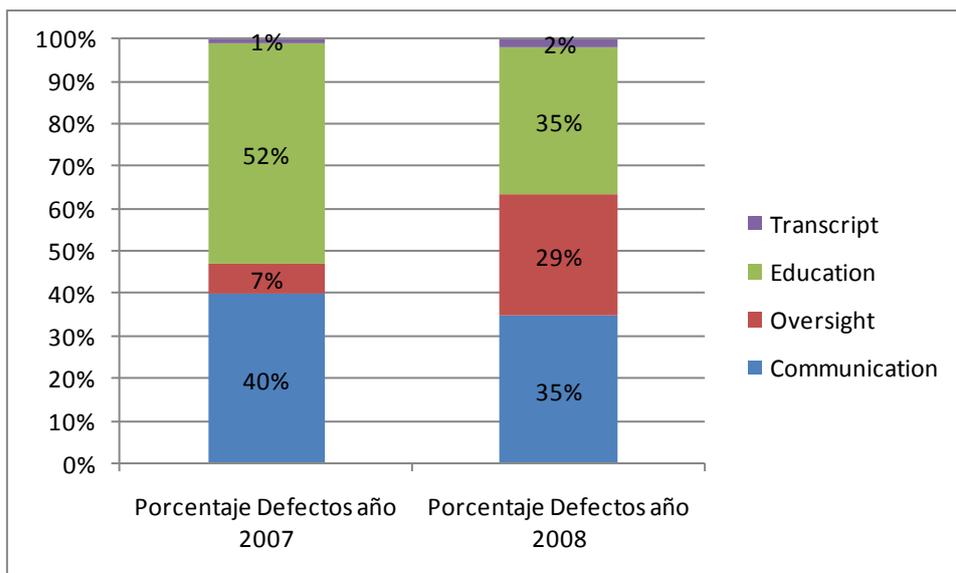


Figura 27 Distribución de defectos por año

En estas gráficas se puede observar cómo la distribución de los tipos de defectos se ha modificado. Los defectos de *Educación* han disminuido de 52% a 35%. El objetivo del proceso propuesto estaba en reducir los defectos de educación, sin embargo para poder afirmarlo es necesario explicar el aumento de los defectos de Oversight y Comunicación.

Por un lado se observa que el porcentaje relativo de Comunicación se ha mantenido (de 40% a 35%), pero los que han aumentado son los defectos de Oversight. Si se interpreta a los defectos de *Oversight* y *Comunicación* como fallas en el proceso, es posible establecer dos escenarios:

1. El proceso está correcto (alineado con las necesidades de la fábrica). Entonces los proyectos de la fábrica no están siguiendo los procesos de la fábrica y por lo tanto es esperable que los proyectos cometan más defectos de Comunicación.
2. El proceso no está correcto (no está alineado con las necesidades de la fábrica), por lo que el aumento en los defectos de Oversight puede estar evidenciando que el proceso estándar no es el adecuado para el contexto de los proyectos. Y esto sería un indicativo de que el entorno de la fábrica se vio modificado en forma más rápida de lo que sus procesos pudieron adaptarse.

Para ambos casos, para completar el razonamiento se estudiarán las métricas de adherencia al proceso de la fábrica aportadas por el equipo de PPQA.

6.5.5.3.2 Efecto de las no conformidades de proceso sobre la distribución de defectos

Para estudiar la evolución del comportamiento de la fábrica en función de la adherencia al proceso, se tratará al “seguimiento de los procesos” como un proceso que puede ser controlado estadísticamente según los criterios de [145].

Por lo que siguiendo los mismos pasos de:

1. Establecer la línea base de performance
Nuevamente la línea base de performance serán los mismos proyectos de Proyectos 2007.
2. Ejecutar proyectos y recolectar datos
Los datos a utilizar representan las no conformidades de los proyectos de la fábrica (Tabla 28).

Proyectos	No Conformidades Producto	No Conformidades Proyecto
Proyecto1_2007	5	16
Proyecto2_2007	11	11
Proyecto3_2007	7	16
Proyecto4_2007	15	9
Proyecto1	4	1
Proyecto2	16	17
Proyecto3	Este punto se establecerá como el primero en temporalidad para minimizar la variación que él produce	

Tabla 28 No conformidades de los proyectos

- La organización discrimina las No Conformidades de los proyectos según la categoría de área de proceso al que esté asociada:
- No Conformidad de Producto: Para aquellos procesos de la fábrica que están asociados a la categoría de Ingeniería de CMMI
 - No Conformidad de Proyecto: Para aquellos procesos de la fábrica que están asociados a la categoría de Gestión de Proyecto de CMMI
3. Establecer nuevas líneas de control
Con los mismos tres proyectos del 2007. Las figuras siguientes muestran en la misma gráfica la variación en los límites de control.

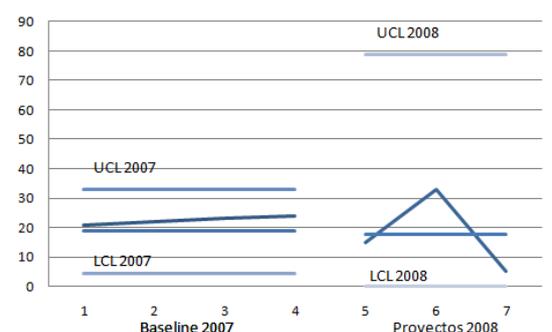


Figura 28 Evolución del Seguimiento de procesos

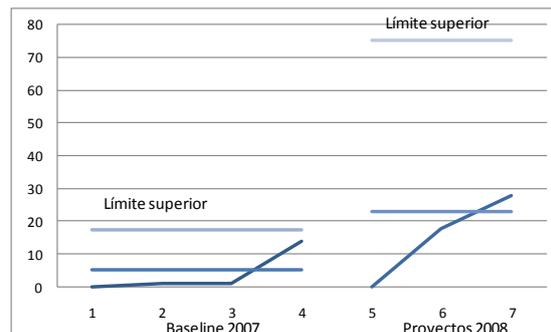


Figura 29 Gráfico de Control de la variación del seguimiento de procesos

La Figura 28 presenta el gráfico de control (sección 2.3.4) del proceso “No Conformidades de PPQA”, en ella se han resaltado los límites de control para los dos grupos de proyectos (2007 y 2008).

4. Evaluar.

Al estudiar estos gráficos de control se observa que existe una leve disminución de las no conformidades promedio (del 2007 al 2008 el promedio de no conformidades por proyecto cae de 19 a 18).

Sin embargo, al igual que con los análisis anteriores, el rango de variación es más grande en el año 2008 (de 0 a 79) que en el 2007 (de 4,6 a 33).

Esta observación confirma al escenario 2 del apartado anterior, y es reflejo cuantitativo del conocimiento del contexto de la organización (ver sección 6.1.1). Se puede afirmar que:

- Las métricas obtenidas de No Conformidades de PPQA pueden ser utilizadas como indicador de la estabilidad de los procesos.
- Las presiones del entorno de negocio de la organización tuvieron un mayor efecto en la introducción de defectos que la fuerza de nuestra propuesta realizaba en la reducción de defectos.

En la sección 6.6 se presenta una explicación de estos eventos aplicando modelos de sistemas. El objetivo de esta tarea es contrastar la afirmación anterior.

6.6 Evaluación del impacto de la volatilidad de los procesos en la propuesta

En esta sección se presenta un modelo de sistémico[148] que fue construido para explicar el comportamiento observado en la fábrica.

Los modelos de pensamiento sistémico se utilizan para facilitar el razonamiento en sistemas complejos. El objetivo de estas simulaciones es entender el comportamiento del sistema cuando se lo somete a distintos estímulos. Por lo tanto, en este contexto, no es tan

importante conseguir reproducir exactamente los datos observados, sino conseguir un modelo cuyo comportamiento se ciña al comportamiento relevado.

Los resultados de esta sección fueron presentados en [40], donde se realizó un modelo de simulación utilizando la información disponible de los proyectos de la organización de software.

6.6.1 Construcción del modelo de pensamiento sistémico

En la sección 2.3.5.3, se presentó el proceso recomendado por los autores del software de pensamiento sistémico utilizado. En la Tabla 29 se resumen los pasos llevados a cabo para construir este modelo.

Definir el sistema a Modelar	
<i>Un modelo que represente el comportamiento de los defectos según se observó en la fábrica de software.</i>	
Desarrollar la hipótesis <i>a. El entrenamiento de los desarrolladores influye sobre los defectos inyectados en el código</i> <i>b. La estabilidad del proceso de desarrollo tiene más injerencia en los defectos finales que la capacitación de los desarrolladores</i>	Identificar la audiencia Objetivo <i>Docentes, Investigadores, y profesionales con formación en software, y conocimientos de CMMI</i>
Probar la Hipótesis <i>Al final de esta tabla en esta sección</i>	Definir los objetivos de aprendizaje <i>¿Cómo incide la estabilidad de los procesos en la inyección de defectos?</i>
Extraer Conclusiones Validar Conclusiones <i>En la sección 6.6.2</i>	Definir la estrategia de aprendizaje Implementar la estrategia de aprendizaje <i>En la sección 6.6.2</i>

Tabla 29 Aplicación del proceso para construir modelos de sistemas

La Figura 30 presenta el modelo obtenido. Este modelo representa el comportamiento de los proyectos de la fábrica en función de los defectos reportados, y estima los defectos remanentes en el código a partir de la eficiencia de los testers reportada por la organización. Existen dos reservorios centrales, “DefectsInCode” (hacia el centro y arriba) y “DefectsReported” (hacia el centro y abajo).

El modelo describe cómo al codificar (flujo “coding”) se inyectan defectos en el stock “DefectsInCode” y estos sólo se reducen cuando se retrabaja sobre el código que tiene estos defectos (flujo “reworking”). Sin embargo, no se puede trabajar en un defecto si éste no ha sido identificado y reportado por los testers (flujo “testing”). Los defectos reportados pasan a un estado “completado” (representado por el flujo de salida “Marking Done”), pero este vaciado del Stock “DefectsReported” es también dependiente del ritmo en que los desarrolladores retrabajan (conector del flujo “reworking” al flujo “MarkingDone”).

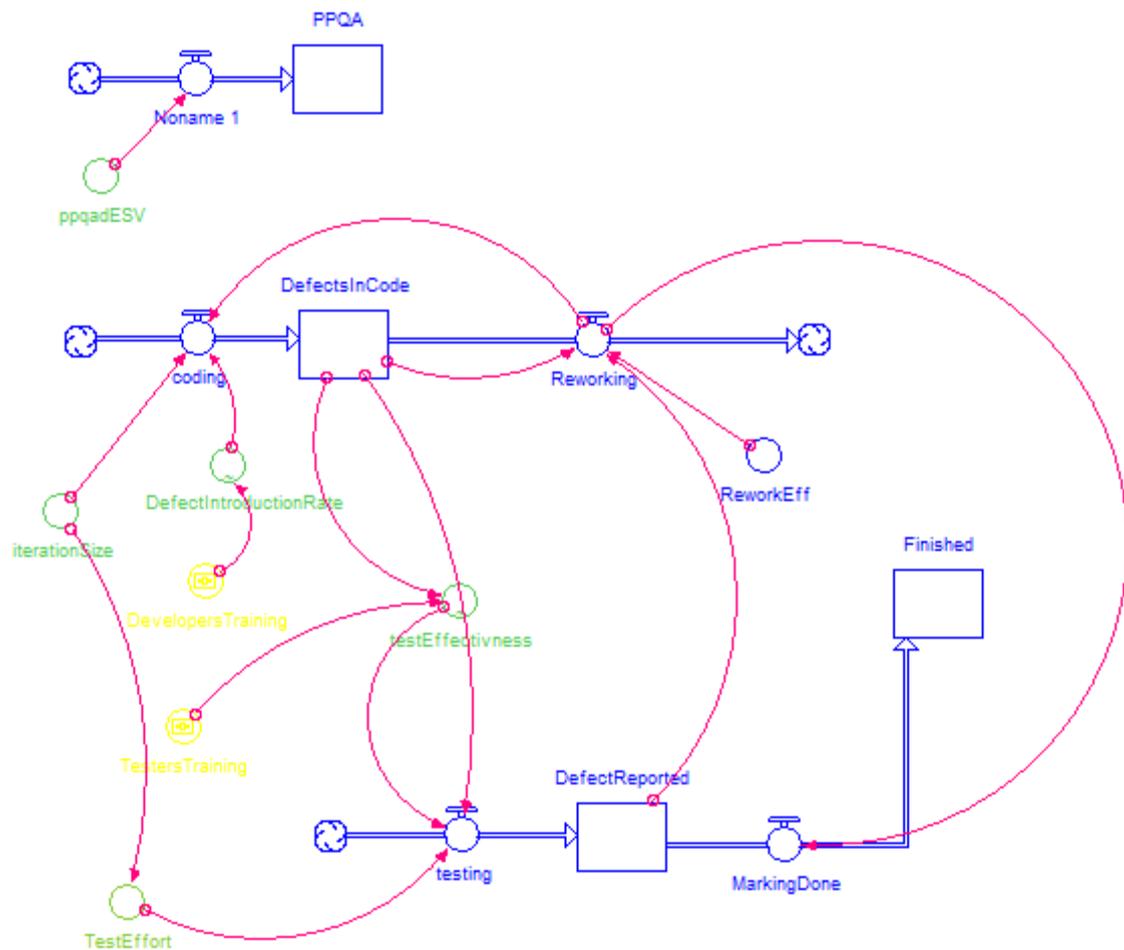


Figura 30 Modelo de sistemas de un proyecto de la organización

Las otras entidades importantes para destacar son:

- los convertidores:
 - En Verde: Representan las variables del modelo que pudieron ser extraídas desde los datos históricos de la organización.
 - iterationSize: Medido en horas, representa el tamaño de las iteraciones. Está preestablecido con el tamaño de las iteraciones observadas en los proyectos. Cada intervalo de tiempo representa el esfuerzo en tiempo de desarrollo tomado de los datos de los proyectos de 2007. El valor de cada iteración surge de las medias del esfuerzo de las iteraciones de los proyectos..
 - DefectIntroductionRate: Medido en Defectos/Hora. Representa el ritmo con el que los desarrolladores inyectan defectos. Al incluir en el modelo la ponderación por la efectividad del entrenamiento se modificó la constante por una función continua

cuyos valores estuvieran entre el promedio y una desviación estándar de los valores históricos.

Representa una función F: F(Entrenamiento → Defectos/Hora) tal que:

$$0,012 < F(x) < 0,030$$

- testEffectiveness: Medido en Defecto/Hora. Esta preestablecido con la efectividad histórica de los testers por hora de desarrollo.

Inicializado según la media histórica de la Organización: 0.60 Defectos/Hora.

De la misma manera que en la función anterior, al incluir la ponderación de la efectividad del entrenamiento. El convertidor se representa como una función F: F(Entrenamiento → Defectos/Hora).

Tal que:

$$0,29 < F(x) < 0,93$$

- testEffort: Medido en horas, esta preestablecido al 10% de *iterationSize*, respetando las estimaciones de la fábrica.
- En Amarillo: Representa las variables del modelo que serán modificadas para probar las hipótesis.
 - DevelopersTraining: Es un porcentaje que representa el entrenamiento de los programadores.
El rango en que será variado irá de 0,5 a 1, donde 1 representa desarrolladores altamente entrenados.
 - TestersTraining: Es un porcentaje que representa el entrenamiento de los programadores.
El rango en que será variado irá de 0,5 a 1, donde 1 representa testers altamente entrenados.
- En Azul: Representa un valor calculado que es utilizado para asegurar la estabilidad del modelo.
 - reworkEffectiveness: Representa la efectividad de corrección de defectos.
Fue calculado con un valor de 0.3 Defectos / Hora.

La Tabla 30 resume los datos utilizados para inicializar al modelo.

Proyecto	Número de Iteraciones en el ciclo de vida	Esfuerzo Total en horas normalizadas	Esfuerzo en Testing (horas totales normalizadas)	Defectos reportados	Defectos/Hora Test	Defectos/Hora Desa
Proyecto1	3	5750	119	115	0.97	0.02
Proyecto2	2	922	40	53	1.33	0.06
Proyecto3	3	4491	87	54	0.62	0.01
2007_Proyecto1	3	4744	353	85	0.24	0.02
2007_Proyecto2	4	9136	296	125	0.42	0.01
2007_Proyecto3	5	21179	928	431	0.46	0.02
2007_Proyecto4	5	13983	722	138	0.19	0.01

Tabla 30 Datos utilizados para ajustar al modelo

La ejecución del modelo se presenta en la Figura 31.

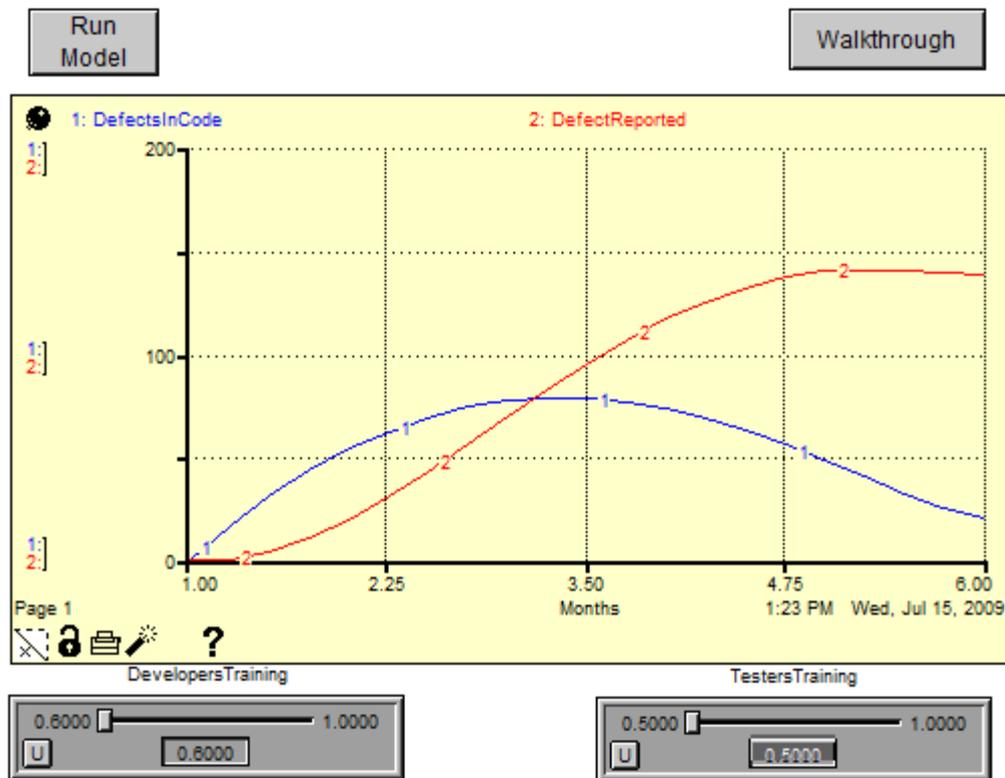


Figura 31 Comportamiento del Modelo de Simulación en su "estado estable"

6.6.2 Ejecución del modelo y aprendizaje realizado

Una vez que se dispone del modelo estable y consistente con las observaciones, el siguiente paso consiste en modificar sus variables para observar cómo se comporta el sistema ante diferentes estímulos:

H1 Si se aplica entrenamiento a los testers, su efectividad mejora y se observarán más defectos reportados.

H2 Si se aplica entrenamiento a los programadores, su efectividad mejora, se observarán menos defectos reportados

H3 La conformancia a los procesos tiene más impacto sobre la variación de defectos que el entrenamiento a los recursos. Como se ve en el modelo (Figura 30) este escenario no va a ser posible investigarlo con el modelo. En él se observa que el stock PPQA, no está conectado con ninguna otra parte del modelo. No se consiguió una relación que respetara la integridad de unidades en el modelo para vincular las No Conformidades de PPQA con los resultados de la ejecución del proceso (ver sección 6.6.3).

La Figura 32, muestra el comportamiento del sistema para el escenario **H1**. Este escenario representa un entorno donde los Testers están altamente entrenados (la barra de TestersTraining está al máximo). La variación en la gráfica es casi imperceptible, pero al

observar los resultados en la Tabla 31 se evidencia que los testers son capaces de reportar más defectos que si no estuviesen entrenados.

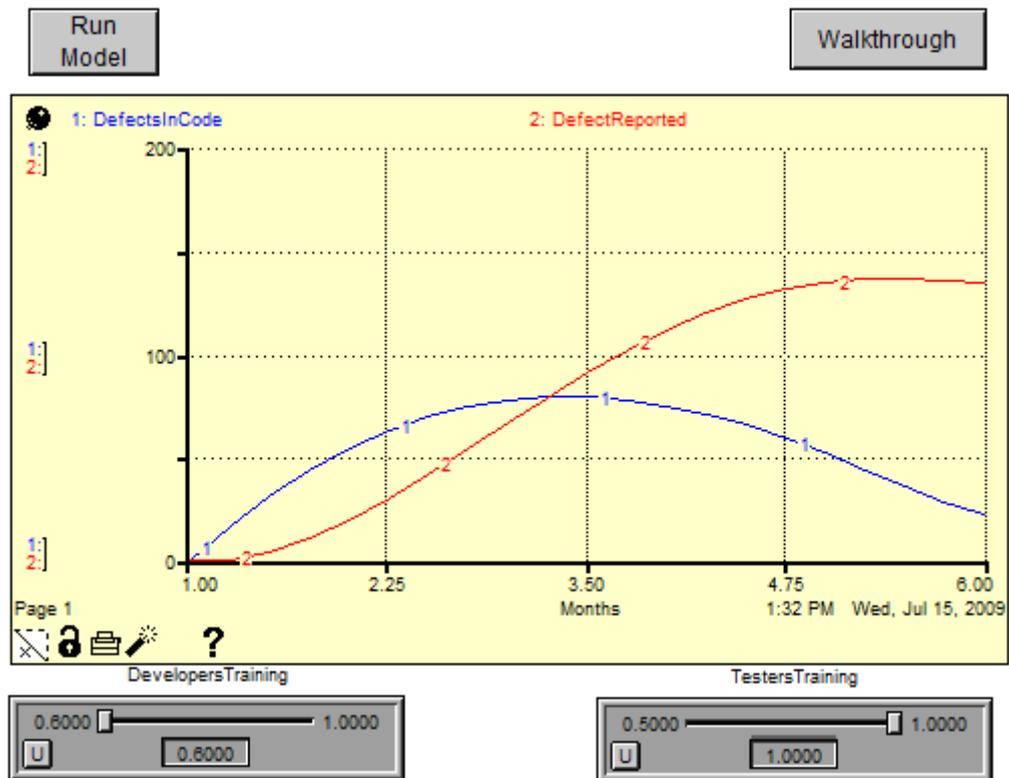


Figura 32 Comportamiento del Modelo de simulación con Testers Altamente Efectivos

Para **H2**, representa el escenario donde los desarrolladores están altamente entrenados y tienen foco en la calidad del producto. Al observar tanto la Figura 33 como la Tabla 31 se evidencia que se redujo el número de defectos en el código, y también por lo tanto los defectos reportados.

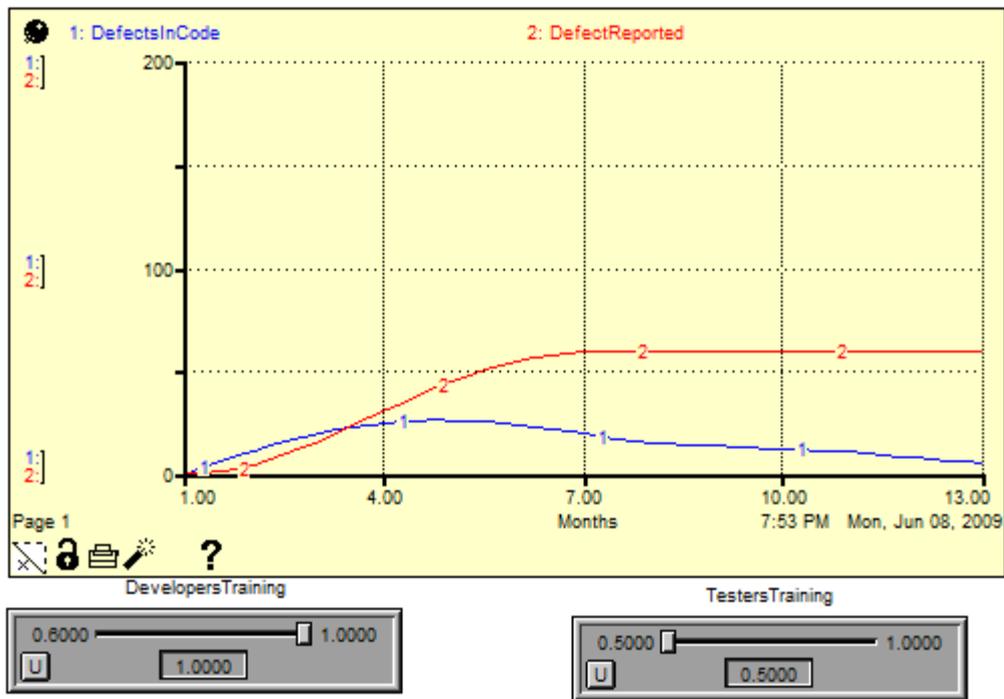


Figura 33 Comportamiento Modelo con desarrolladores Altamente Efectivos

Iteración	Modelo Estable		Modelo Con Testers Entrenados (H1)		Modelo Con Programadores Entrenados (H2)	
	DefectosEnCodigo	DefectosReportados	DefectosEnCodigo	DefectosReportados	DefectosEnCodigo	DefectosReportados
1	0	0	0	0	0	0
2	55	0	55	0	33	0
3	88	19	89	0	53	4
4	99	65	110	19	63	29
5	87	114	112	74	58	62
6	55	145	89	130	40	86

Tabla 31 Resultado de la ejecución de los escenarios

Por lo tanto, de estudiar los escenarios **H1** y **H2** se puede establecer que:

- el entrenamiento de los desarrolladores tiene impacto en los defectos incurridos en los proyectos.
- Entrenar a testers, no disminuye los defectos remanentes en el código fuente, sino que aumenta la velocidad con que éstos son detectados.

Ambas observaciones son consistentes con los resultados de la experimentación en la organización.

6.6.3 No Conformidades como predictor de defectos del proyecto

Finalmente, para el escenario **H3**, se intentó establecer una correlación entre los defectos de PPQA y la información de defectos.

La hipótesis de trabajo que surge del escenario H3 sería:

- Las No Conformidades de PPQA pueden utilizarse como predictores de los defectos asociados al proceso (los de Comunicación y Oversight).

Para evaluar esta afirmación se realizaron regresiones lineales entre los datos de los proyectos. Se analizó la correlación entre los datos de NoConformidades y los datos de defectos:

- No Conformidades de Producto
- No Conformidades de Proyecto
- Total de NoConformidades
- Total de Defectos
- Defectos por Hora Normalizada de Programador
- Porcentaje Defectos de Comunicación y Oversight

Las siguientes tablas presentan los resultados obtenidos:

	No Conformidades Producto	No Conformidades Proyecto	Todas las No Conformidades
Correlación	0.050	0.029	0.046
Correlación 2007	0.024	0.133	0.224
Correlación 2008	1.000	0.121	0.976

Tabla 32 Correlación No Conformidades y total de defectos

	No Conformidades Producto	No Conformidades Proyecto	Todas las No Conformidades
Correlación	0.21	0.00	0.05
Correlación 2007	0.02	0.13	0.22
Correlación 2008	1.00	1.00	0.12

Tabla 33 No Conformidades y Defectos por Hora Normalizada de Programador

	No Conformidades Producto	No Conformidades Proyecto	Todas las No Conformidades
Correlación (todos)	0.08	0.04	0.01
Correlación 2007	0.88	0.93	0.32
Correlación 2008	1.00	1.00	1.00

Tabla 34 No Conformidades y Porcentaje Defectos de Comunicación y Oversight

En las tres tablas se observa cómo casi todas las correlaciones que involucran sólo a los puntos de los proyectos del 2008 tienen correlaciones de valor 1 o cercanos. Esto resulta de la falta de datos de la muestra, por lo que no se podrán extender las conclusiones a la muestra de proyectos del 2008.

Sin embargo, es interesante rescatar el resultado de la Tabla 34; en esta tabla en la segunda fila se ve una correlación significativa entre las No Conformidades de Proyecto y el porcentaje de defectos de Comunicación y Oversight.

En la sección anterior se mencionó que las No Conformidades de Proyecto representan para la fábrica faltas a los procesos asociados a las áreas de proyecto de la categoría de Gestión de Proyecto de CMMI. Por otro lado, los defectos de Comunicación y Oversight se pueden asociar a las faltas en los procesos.

Por lo tanto, en el entorno estable del año 2007, es relevante que la correlación haya dado significativa. Implica que cuanto más faltas al proceso realizan los proyectos, más defectos tiene el proyecto.

El siguiente paso consiste en graficar los puntos para estudiar la recta a la que pueden aproximarse mediante regresión lineal.

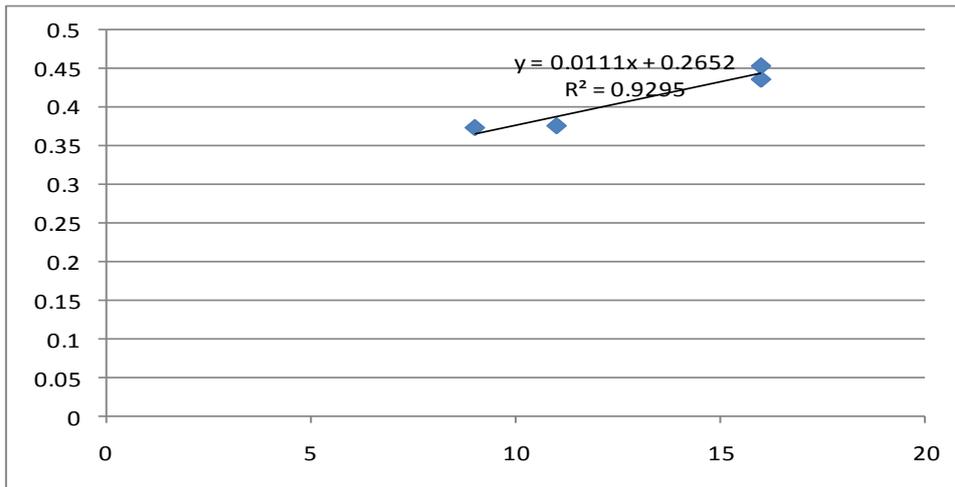


Figura 34 Gráfico de la correlación entre No Conformidades y Porcentaje de Defectos de proceso

La figura anterior muestra que la recta de regresión no pasa por el centro del eje, esto es indicativo de que los proyectos de la fábrica incurren en un 26% de defectos, sólo por ejecutar los procesos.

Por lo que se puede concluir que en un entorno estable, las no conformidades asociadas a las áreas de procesos de Gestión de Proyecto, pueden utilizarse como predictores de los defectos de los proyectos. No se puede realizar una afirmación cuando un proceso no es estable (año 2008).

A pesar de la falta de datos para poder extender las conclusiones, este resultado es interesante por que confirma, por lo menos levemente, que existe una relación entre la adherencia a los procesos y los defectos incurridos. Dejando planteadas preguntas como si es posible interpretar la recta de regresión como medio para medir la alineación de procesos al negocio.

6.7 Resumen de las experimentaciones

En esta sección resumimos los resultados obtenidos de las experimentaciones realizadas en la propuesta. A modo ilustrativo, presentamos este resumen sobre una modificación de la Tabla 12 Partes experimentadas de la propuesta.

Propuesta	Etapas de la Propuesta	Sección	Resumen de la Experimentación
Plan de Despliegue	Limitaciones y Pre-requisitos	6.1 y 6.2	Presentamos un estudio de cómo la fábrica objetivo cumple con las limitaciones de esta propuesta. La debilidad de la fábrica en uno de los puntos (Estabilidad de los procesos sección 4.2), queda de manifiesto en los resultados finales del cálculo del ROI (sección 6.5.5.3)
	Establecer la capacidad de la organización	6.2	La sección 6.2.1 y 6.2.2 describe el despliegue de la taxonomía en la fábrica de software. En el experimento de clasificación realizado con sujetos de la fábrica y voluntarios externos a ella se concluye que: <ul style="list-style-type: none"> • el entrenamiento tiene influencia en la capacidad de clasificación de los grupos y de los individuos • el impacto de un individuo no entrenado no es significativo en la capacidad de clasificación de un grupo de 4 personas. <p>La sección 6.2.3.5 presenta la aplicación de técnicas de procesamiento de lenguaje natural automático al mismo problema.</p> <ul style="list-style-type: none"> • Un mecanismo de software que consigue capacidades de clasificación superiores a la del humano experto.
	Desplegar Proceso de Análisis causal	6.4.2	
	Seleccionar e Implementar el proceso completo en	6.3	El análisis de resultados primeros en el año 2007 proyecta un Retorno de Inversión en la propuesta del 173%

	un piloto		
	Analizar Resultados		
Proceso Propuesto	Ejecutar actividades de Verificación y Validación		No se realizaron modificaciones sobre los procesos de Verificación y Validación.
	Realizar sesiones de análisis causal	6.4.2	Los resultados de estas sesiones en la fábrica se presentaron en la Tabla 24
	Planificar el entrenamiento o en función de los resultados	6.4.3 6.4.4	Se presentaron los resultados de las evaluaciones Nivel 1 y Nivel 2 de la intervención de entrenamiento
	Realizar el entrenamiento		
	Evaluar el Retorno de Inversión	6.5 y 6.6	<ul style="list-style-type: none"> Se obtuvo un rango de retorno de Inversión de ente -280% y 481% para un periodo de 6 meses (sección 6.5) Se estableció mediante simulación el impacto del entrenamiento en la inserción de defectos en el código (sección 6.6)
	Comunicar los resultados a la alta gerencia		

Tabla 35 Resumen de las experimentaciones

A la luz de los resultados presentados en este capítulo y resumidos en la Tabla 35, podemos realizar las siguientes afirmaciones con respecto a los elementos de la propuesta:

Con respecto al **plan de implantación**, las etapas previstas fueron seguidas por la organización. Dejando aparte las etapas en las que por motivos estratégicos, principalmente para facilitar la adopción del proceso en la fábrica, no intervinimos; las adaptaciones realizadas al proceso por la fábrica no afectaron el espíritu del plan de

despliegue propuesto. Aunque no se mencionó en ningún momento en este capítulo, el mejor indicador del éxito de este plan de despliegue es que el proceso propuesto se llegó a ejecutar, y se está ejecutando todavía, en la fábrica de software. Durante las secciones dedicadas a la experimentación del proceso propuesto acotamos las conclusiones al alcance individual del paso, pero en el conjunto, el éxito de la sección 6.2 permite experimentar el proceso y el cálculo del ROI de las intervenciones de entrenamiento.

Por otro lado, la fábrica adaptó el proceso al proyecto para amortizar los costos de entrenamiento. En lugar de planificar y ejecutar intervenciones de entrenamiento para cada proyecto, se ejecutaron para todos los proyectos que ejecutaban en el mismo momento. Aunque esto favorece al cálculo en término de amortización de costos (una sola planificación de entrenamiento), los beneficios también se ajustan a la baja.

De todas formas, con respecto al proceso es justo afirmar que *fue ejecutado correctamente. Las modificaciones fueron necesarias para adaptarlo, y el impacto no compromete la esencia del proceso.*

Asociado al proceso, el procedimiento para el cálculo del ROI pudo ser probado, creando así un mecanismo reproducible para asignar resultados a las inversiones de entrenamiento (punto 1 del Planteamiento).

En este caso, la experimentación también confirmó la importancia de tener procesos estables para poder introducir cambios y calcular beneficios de estos cambios (limitación que habíamos impuesto en la sección 4.2).

Se construyó un modelo de sistemas que permitió:

- Establecer el impacto del entrenamiento en los defectos de los proyectos.
- Mostrar que la inversión en entrenamiento de los tester no presenta el mismo tipo de impacto que la inversión en entrenamiento de los desarrolladores.
- Presentar una estimación de los defectos remanentes en los proyectos.
- Establecer la existencia de una fuerza desestabilizadora del contexto de la fábrica. Este resultado nos ha abierto una nueva línea de investigación que estudiará la aplicabilidad de las No Conformidades como proxy de los resultados de defectos de los proyectos de desarrollo.

En términos generales, todas las experiencias realizadas en los laboratorios se utilizaron para fortalecer la experiencia en el entorno de la fábrica.

La evaluación de los clasificadores (sección 6.2.3.4) y su extensión mediante la aplicación de métodos de procesamiento de lenguaje natural (sección 6.2.3.5) confirman que:

- Para las organizaciones, *el acuerdo del grupo, es más importante que la habilidad de clasificación de un sujeto en especial.*
- El *mecanismo de clasificación por ordenador, una vez desplegado, podrá permitir a organizaciones conseguir clasificaciones de defectos más confiables a través de sus distintos proyectos, recuperar el histórico de defectos luego de cambios a las Taxonomías.*

7 Conclusiones y futuras líneas de investigación

El software esta cada vez más presente en la vida diaria. Mirando hacia el futuro próximo, la producción de software será una tarea realizada principalmente por humanos, recursos que mas allá de estar altamente capacitados necesitarán una actualización constante a lo largo de sus carreras profesionales. Es por esto que la capacitación recibida por medio de las instituciones formales de educación no es suficiente para cumplir con las necesidades de las organizaciones productoras de software.

Las organizaciones de software llevan ya años invirtiendo en programas de mejora de proceso, inclusive cuando la mayoría de estos programas no consiguen obtener beneficios tangibles en términos de retorno de inversión. La inversión en entrenamiento, tanto sea por requerimiento de estos programas, por convencimiento propio de la organización o por necesidad de mantener a su fuerza de trabajo actualizada, las organizaciones de software mantendrán su inversión en programas de capacitación internos.

Durante el estudio del Estado del Arte en el capítulo 2 se presentaron los trabajos de cada uno de los pilares que soportan esta tesis; en el área de entrenamiento se vio cómo las organizaciones en general tienen dificultades para evaluar cuánto contribuye el entrenamiento a sus resultados. Se vio también cómo los modelos de mejora de proceso han ganado confianza y están cada vez más presentes en las organizaciones productoras de software, algunos de estos modelos resaltan la importancia del entrenamiento organizacional como una disciplina clave para mantener la competitividad, retener el talento y favorecer la adopción de los propios modelos. De esta forma, el entrenamiento organizacional, y los problemas que otras industrias tienen con él se mantienen vigentes a la hora de aplicar las conclusiones en los métodos en las organizaciones dedicadas a la producción de software.

En este trabajo de tesis se presentó un proceso que permite a organizaciones productoras de software obtener un valor tangible de su inversión en entrenamiento. Este proceso vincula al entrenamiento organizacional con el análisis causal de defectos para proveer a las organizaciones productoras de software de un mecanismo para calcular el Retorno de Inversión en Intervenciones de Entrenamiento. Este proceso contempla la variación natural presente en los entornos de producción de software, por lo que no está sujeto a los problemas de estimación de promedios de otros mecanismos de retorno de inversión.

Como parte de nuestra propuesta quisimos asegurar que otros practicantes pudieran reproducir nuestros resultados, por lo que la misma incluye un plan de despliegue para que una organización de software que haya seguido un camino de mejora de procesos utilizando el modelo CMMI, pueda desplegar el proceso propuesto.

La experimentación de nuestra propuesta de tesis estuvo dirigida a validar el proceso de cálculo del ROI. Realizándose un caso de estudio en una organización productora de

software. Este caso de estudio fue complementado con experiencias de laboratorio para asegurar el cubrimiento de todas las propuestas asociadas a este trabajo de tesis.

Los resultados y aportaciones de este trabajo de tesis se resumen a continuación:

- **Un proceso para la evaluación de intervenciones de entrenamiento que contempla la variación natural de los procesos productivos.** El proceso propuesto en la sección 5.1, es la contribución que viene a resolver el problema planteado en esta tesis. El proceso comunica una serie de pasos que deben ser ejecutados para conseguir una evaluación de las intervenciones de entrenamiento en términos de Retorno de Inversión.
 - **Un mecanismo para el Cálculo de Retorno de Inversión en intervenciones de entrenamiento en proyectos de producción de software.** Como parte del proceso de evaluación se propone un mecanismo que permite calcular el retorno de inversión de las intervenciones de entrenamiento. **Este mecanismo tiene en cuenta la variación natural** de los procesos de software y la utiliza como entrada para dar una respuesta que en lugar de presentarse como un único valor de ROI, es presentada como un rango esperado. Entendemos que este mecanismo es más aplicable a la industria de software que la fórmula clásica del ROI, y que además se presta a menos interrogantes ya que lo importante en estos cálculos es demostrar contribución “más allá de una duda razonable”[28] con los objetivos de la organización.
 - **Vinculación de los defectos con las necesidades de entrenamiento.** El proceso propuesto utiliza técnicas de análisis causal de defectos para seleccionar intervenciones de entrenamiento orientadas a reducir defectos. El proceso vincula así a las necesidades de entrenamiento, con los defectos de producción con el análisis causal.
 - **Impacto del entrenamiento en la habilidad de clasificación de los recursos de una fábrica.** Las organizaciones de software que basen sus procesos de análisis causal en sus defectos clasificados, deberán asegurarse que sus diferentes proyectos clasifican sus defectos de manera consistente. En los resultados presentados en 6.2.3.4 vimos como el entrenamiento tiene impacto en la habilidad de clasificación de los individuos. En el caso concreto de esta fábrica de software, el entrenamiento de 4 horas mejoró en un 10% el acuerdo de clasificación entre los individuos entrenados. Se podría argumentar que en este caso, mejorar el entrenamiento mejoraría los índices de acuerdo de los individuos.
Además, se vio que si se introduce un individuo no entrenado en un equipo de 4 personas entrenadas, la habilidad de clasificación del grupo no sufre significativamente. Este resultado es importante para una industria como la de producción de software en Uruguay, donde las organizaciones deben coexistir con un entorno que facilita el movimiento de recursos entre distintas compañías.

- **Es posible clasificar defectos basándose en estudio del lenguaje natural y esta clasificación es tan precisa como la clasificación de recursos expertos.** Como mecanismo de validación adicional, realizamos un experimento en conjunto con la Universidad de Memphis para aprovechar los mecanismos de análisis semántico de lenguaje natural que ellos habían desarrollado. Aplicando sus resultados al problema de clasificación de los individuos en una fábrica de software vimos como es posible llegar a acuerdos que igualan los de un individuo con experiencia y en la organización. Este resultado puede implicar alternativas para el desarrollo de clasificaciones de defectos y su uso en fábricas de software (como veremos en la sección 7.1)
- **Un proceso de Despliegue para que organizaciones de software en las condiciones de la sección 4 puedan desplegar y aplicar el proceso propuesto.** Uno de nuestros objetivos con el proceso propuesto fue de proveer un mecanismo que sea reproducible para distintas empresas en distintos entornos. De esta manera era necesario proveer un proceso de despliegue que llevara a las organizaciones a estar en condiciones de aprovechar los beneficios del proceso propuesto.
Este plan de despliegue presentado en la sección 4.3 fue experimentado en la fábrica de software y sus pasos llevaron al despliegue del proceso con los resultados ya mencionados
 - Es posible utilizar técnicas de análisis causal en entornos de baja madurez. El punto clave para el éxito de este proceso es que los desarrolladores en entornos de baja madurez sean capaces de realizar sesiones de análisis causal efectivas. Este punto fue incluido dentro del proceso de despliegue, y el éxito de su despliegue confirma los resultados de otros investigadores que ya habían intentado desplegar procesos de análisis causal y resoluciones en organizaciones que han conseguido nivel de madurez 3 en el modelo CMMI. Como veremos en la sección siguiente, éste es uno de los puntos en que seguiremos trabajando en el futuro.

7.1 Futuras líneas de Investigación

Durante el proceso que nos llevó a elaborar este trabajo de tesis, se descartaron distintos caminos y opciones de investigación que se encuentran en la periferia del alcance de esta tesis. Los tres ítems de esta sección representan algunos de estos puntos en los que ya hemos comenzado a trabajar.

- **Análisis Automático de defectos mediante aplicación de técnicas de lenguaje natural.** En la sección 6.2.3.5 mostramos cómo es posible aplicar técnicas de procesamiento de lenguaje natural para realizar clasificaciones de defectos. Sin embargo, todavía quedan bastantes preguntas por relevar en este contexto.
 - ¿Pueden conducirse sesiones de análisis causal en base a defectos clasificados por un ordenador? Estas sesiones, ¿serán más o menos efectivas que las basadas en defectos clasificados por humanos?

- Puede el algoritmo de clasificación desarrollado aplicarse a todo el histórico de defectos de una organización de software? ¿Qué beneficios puede esperar esta organización de tener su histórico clasificado?
- Se puede aplicar el concepto de Clustering de defectos al problema de clasificación? ¿Podrán las organizaciones realizar análisis causal de Grupos de Defectos (clusters) que variarían en cada ejecución del mecanismo de clasificación por el ordenador?
- **Despliegue de Análisis causal en entornos de baja madurez (sin gestión cuantitativa).** Ya se mencionó en la sección 2.2.3 que la próxima versión de CMMI incluirá algunas prácticas de análisis causal a niveles de madurez 3. Sin lugar a duda esto va a llevar a las organizaciones que ya tienen ese nivel a encontrar maneras eficientes de implementar este nuevo requerimiento. En este contexto entendemos que es importante poder brindarles las siguientes respuestas a las organizaciones.
 - ¿Qué herramientas y métodos existen para realizar análisis causal en entornos donde los procesos no son necesariamente estables (o que no están cuantitativamente gestionados)? ¿Puede medirse la eficacia del análisis causal en este contexto?
 - ¿Cuáles son las condiciones mínimas que un proceso debe cumplir para ser causalmente analizado?
- **Estudio de la viabilidad de utilizar No Conformidades de PPQA como predictor de Costo, Calidad, y Cronograma. (estudio de correlación con defectos de software).** Los resultados de nuestra experimentación mostraron como la predictibilidad de los procesos impacta en los resultados esperados del mismo. Sin embargo no pudimos llegar a contestar las siguientes preguntas:
 - Confirmar que las No Conformidades de PPQA pueden usarse como predictores de los defectos de los proyectos. Puede aplicarse esta conclusión para justificar la inversión en mejora de procesos por parte de las organizaciones? O, ¿Puede establecerse a partir de esta predicción la adecuación de los procesos definidos a las necesidades de procesos reales de la organización?
 - Las No conformidades podrían utilizarse como predictores de:
 - Cronograma, en el contexto de que los proyectos eviten mecanismos de control de los procesos para acelerar los tiempos de desarrollo (al tiempo que aumentan el riesgo sobre la calidad del producto)
 - Costo, tanto teniendo en cuenta el costo de retrabajo por los defectos ocasionados por las fallas a los proyectos como por el esfuerzo en mantener la conformancia.
 - Calidad, los defectos relacionados con las fallas de los procesos. ¿Podrán reducirse mediante iniciativas de mejora de procesos? ¿Se podrá aplicar el mecanismo de ROI propuesto en esta tesis para verificarlo?
- **Utilización de modelados y simulaciones como herramientas para realizar experimentos en Ingeniería de Software.** Para entender los resultados de la experimentación se utilizaron herramientas de modelado y simulación de procesos

de software. En la sección 2.3.5.1, se mostró cómo este es un camino incipiente en la investigación en ingeniería de software. Entendemos que puede ser un primer nivel de validación de nuevas propuesta antes de llevarlas a estudios más costosos como casos de estudio o experimentos controlados.

8 Bibliografía

- 1 Gray, E.M., and Smith, W.L. 'On the limitations of software process assessment and the recognition of a required re-orientation for global process improvement', *Software Quality Journal*, 7, (1), (1998). pp. 21-34.
- 2 Charette, R.N. *Software engineering risk analysis and management*. New York: Intertext Publications : McGraw-Hill Book Co.: 1989.
- 3 Banred: Servicio de Pagos por internet. Available at <https://www.banred.com.uy/institucional/data/servicios2.php>, accessed 08/07/2009 2009
- 4 Johnson, J. 'The Chaos Report', Standish Group. West Yarmouth, MA, (2000). pp.
- 5 The Curious Case of the CHAOS Report 2009. Available at <http://www.projectsmart.co.uk/the-curious-case-of-the-chaos-report-2009.html>, accessed 8/07/2009 2009
- 6 Brooks, F. 'No silver bullet: essence and accidents of software engineering. ', *IEEE Computer*, (1987). pp. 4.
- 7 Deming, W.E. *Out of the Crisis*. (First). Cambridge, Massachusetts: MIT Press: 2000.
- 8 Juran, J.M., and Godfrey, A.B. *Juran's Quality Handbook*. 2000.
- 9 Chrissis, M.B., Konrad, M., and Shrum, S. *CMMI : Guidelines for Process Integration and Product Improvement*. (Second). Upper Saddle River, NJ: Addison-Wesley: 2007.
- 10 Paulk, M.C., Curtis, B., Chrissis, M.B., and Weber, C.V. 'Capability maturity model, version 1.1', *Software*, IEEE, 10, (4), (1993). pp. 18-27.
- 11 International Standards Organization. 'ISO 9001:2008(E). Quality Management Systems - Requirements'. 2008.
- 12 International Standards Organization. 'ISO/IEC 15504-1:2004 Information technology Process assessment Part 1: Concepts and vocabulary'. 2004.
- 13 Herbsleb, J., Carleton, A., Rozum, J., Siegel, J., and Zubrow, D. *Benefits of CMM-Based Software Process Improvement: Initial Results*. Inst Carnegie-Mellon Univ Pittsburgh Pa Software Engineering. Carnegie Mellon University, Software Engineering Institute. 1994.
- 14 Herbsleb, J., Zubrow, D., Goldenson, D., Hayes, W., and Paulk, M. 'Software Quality and the Capability Maturity Model', *Communications of the ACM*, 40, (6), (1997). pp. 30-40.
- 15 Curtis, B., Hefley, W.E., and Miller, S.A. *People Capability Maturity Model (P-CMM) Version 2.0*. 2001.
- 16 Humphrey, W.S. *PSP(sm): A Self-Improvement Process for Software Engineers*. Addison-Wesley Professional: 2005.
- 17 Humphrey, W.S. *A Discipline for Software Engineering*. Addison-Wesley Professional 1995.
- 18 Humphrey, W.S. *Introduction to the Team Software Process*. Addison-Wesley Professional: 1999.
- 19 Tajima, D., and Matsubara, T. 'Computer Software Industry in Japan', *Computer*, 14, (5), (1981). pp. 89-96.
- 20 Clements, P., and Northrop, L. *Software product lines*. Addison-Wesley Boston: 2002.
- 21 Greenfield, J., Short, K., Cook, S., and Kent, S. *Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools*. (1st edition). Wiley: 2004.

- 22 Shifting economies--from craft production to flexible systems and software factories. Available at <http://dspace.mit.edu/handle/1721.1/2367?show=full>
- 23 Microsoft Products Lifecycle Information. Available at <http://support.microsoft.com/gp/lifeselect>, accessed November 19 2010
- 24 El Emam, K., and Garro, I. 'Estimating the extent of standards use: the case of ISO/IEC 15504', *The Journal of Systems & Software*, 53, (2), (2000). pp. 137-143.
- 25 Humphrey, W.S. *Winning with Software: An Executive Strategy*. Addison-Wesley Professional: 2001.
- 26 Humphrey, W.S. Why don't they Practice what we preach. SEI.
- 27 Humphrey, W.S. 'Introducing the personal software process', *Annals of Software Engineering*, 1, (1), (1995). pp. 311-325.
- 28 Kirkpatrick, D.L., and Kirkpatrick, J.D. *Evaluating Training Programs: The Four Levels*. (3rd). Berrett-Koehler Publishers: 2006.
- 29 Kirkpatrick, D. 'Four-level training evaluation model', *US Training and Development Journal*, (1959). pp.
- 30 Alliger, G.M., Tannenbaum, S.I., Bennett, W., Traver, H., and Shotland, A. 'A Meta-Analysis of the Relations among training criteria', *Personnel Psychology*, 50, (2), (1997). pp. 341-358
- 31 Rubin, H.A. 'Software process maturity: measuring its impact on productivity and quality'. *Proceedings of Proceedings of the 15th international conference on Software Engineering 1993* 468=476.
- 32 Gopal, A., Krishnan, M.S., Mukhopadhyay, T., and Goldenson, D.R. 'Measurement programs in software development: determinants of success', *IEEE Transactions on Software Engineering*, (2002). pp. 863-875.
- 33 Salas, E., and Cannon-Bowers, J.A. 'THE SCIENCE OF TRAINING: A Decade of Progress', *Annual Review of Psychology*, 52, (2001). pp. 471-499. doi:10.1146/annurev.psych.52.1.471
- 34 Alliger, G.M., and Janak, E.A. 'Kirkpatrick's Levels of Training: Thirty years later', *Personnel Psychology*, 42, (2), (1989). pp. 331-342. doi:10.1111/j.1744-6570.1989.tb00661.x
- 35 Matalonga, S., and San Feliu, T. 'Calculating Return on Investment of Training using process variation', *IET Software*, Accepted for publication., (2011). pp.
- 36 Matalonga, S., and San Feliu, T. 'Using Process and Product Quality Assurance Audits to Measure Process Business Alignment'. *Proceedings of International Conference of Software and Systems Engineering and their Applications*, 2010.
- 37 Matalonga, S., San Feliu, T., and Rus, V.: 'Automatic Defect Classification: An Experience Applying Natural Language Processing'. *Proc. CLEI 2009, Pelotas, Brazil 2009* pp. Pages
- 38 Matalonga, S., and San Feliu, T. 'Using defect data to drive Organizational Training efforts'. *Proceedings of International Conference on Software Engineering Theory and Practice*, 2008 61-68.
- 39 Matalonga, S., and SanFeliu, T. 'Linking Return on Training Investment with Defects Causal Analysis'. *Proceedings of 20th Conference of Software Engineering Theory and Practice*, 2008 42-47.
- 40 Matalonga, S., and San Feliu, T. 'Using defects to account for the investment in support process areas'. *Proceedings of Europe SEPG 09*, 2009.
- 41 Matalonga, S., and San Feliu, T. 'Defect Driven Organizational Training'. *Proceedings of Europe SEPG 2008*, 2008.
- 42 Buren, M.E.V. *2001 State of the Industry Report*. ASTD. 2001.
- 43 Thompson, C., Koon, E., Woodwell, W.H., and Beauvais, J. *TRAINING FOR THE NEXT ECONOMY: An ASTD State of the Industry Report on Trends in Employer-Provided Training in the United States*. ASTD. 2002.
- 44 Sugrue, B., and Rivera, R.J. *State of the Industry Report*. ASTD. 2005.

- 45 Rivera, R.J., and Paradise, A. State of the Industry Report. ASTD. 2006.
- 46 Paradise, A. State of the Industry Report. ASTD. 2007.
- 47 Paradise, A. State of the Industry Report: Investment in Learning Remains Strong. ASTD. 2008.
- 48 Paradise, A., and Patel, L. The 2009 ASTD State of the Industry report. ASTD. 2009.
- 49 Patel, L. ASTD State of the Industry Report. ASTD Annual State of the Industry Report. Association for Training and Development. 2010.
- 50 Shelton, S., and Alliger, G. 'Who's Afraid of Level 4 Evaluation? A Practical Approach', *Training and Development*, 47, (6), (1993). pp. 43-46.
- 51 Russ-Eft, D., and Preskill, H. 'In search of the Holy Grail: return on investment evaluation in human resource development', *Advances in Developing Human Resources*, 7, (1), (2005). pp. 71.
- 52 Wang, G.G., and Wilcox, D. 'Training Evaluation: Knowing More Than Is Practiced', *Advances in Developing Human Resources*, 8, (4), (2006). pp. 528-539.
- 53 Wang, G.G., and Spitzer, D.R. 'Human Resource Development Measurement and Evaluation: Looking Back and Moving Forward', *Advances in Developing Human Resources* 7, (5-15), (2005). pp.
- 54 Phillips, J.J. Return on Investment in Training and Performance Improvement Programs, Second Edition (Improving Human Performance). Butterworth-Heinemann: 2003.
- 55 Holton, E.F., III. 'The flawed 4-level evaluation model', *Human Resource Development Quarterly*, 7 (1), (1996). pp. 5—2.
- 56 Holton, E.F., III. 'Holton's Evaluation Model: New Evidence and Construct Elaborations', *Advances in Developing Human Resources* 2005, (7), (2005). pp. 37-54.
- 57 Holton, E.F., III, and Kirkpatrick, D.L. 'The Flawed Four-Level Evaluation Model [and] Invited Reaction: Reaction to Holton Article [and] Final Word: Response to Reaction to Holton Article', *Human Resource Development Quarterly*, 7, (1), (1996). pp. 5-29.
- 58 Kirkpatrick, D.L. 'How Effective Are Your Supervisory/Management Training Programs?', in Elaine Beich and John E. Jones (Ed.): 'HRD Handbook' (1996)
- 59 Faerman, S.R., and Ban, C. 'Trainee Satisfaction and Training Impact: Issues in Training Evaluation', *PUBLIC PRODUCTIVITY AND MANAGEMENT REVIEW*, 16, (1993). pp. 299-299.
- 60 Ruona, E.A., Leimbach, M., Holton Iii, F., and Bates, R. 'The relationship between learner utility reactions and predicted learning transfer among trainees', *International Journal of Training and Development*, 6, (4), (2002). pp. 218-228.
- 61 Andrews, T.L., and Crewe, B.D. 'Examining training reactions, lessons learned, current practices and results', Rogelberg, S. *Informed decisions: Research-based practice notes*. TIP *The organizational-industrial psychologist*, 36, (4), (1999). pp.
- 62 Wang, G.G., Dou, Z., and Li, N. 'A systems approach to measuring return on investment for HRD interventions', *Human Resource Development Quarterly*, 13, (2), (2002). pp.
- 63 Ilian, H. 'Levels of Levels: Making Kirkpatrick Fit the Facts and the Facts Fit Kirkpatrick', *Proceedings of the 6 th Annual Human Services Training*, (2004). pp.
- 64 Stolovitch, H.D., and Maurice, J.G. 'Calculating the return on investment in training: A critical analysis and a case study', *Performance Improvement*, 37, (8), (1998). pp.
- 65 Laiken, M.E. 'Models of Organizational Learning: Paradoxes and Best Practices in the Post Industrial Workplace', *ORGANIZATION DEVELOPMENT JOURNAL*, 21, (1), (2003). pp. 8-19.
- 66 Moy, J. Showing that enterprise training pays: lessons from the literature. National Centre for Vocational Education Research. 2001.
- 67 Swanson, R.A., and Holton, E.F. *Foundations of Human Resource Development*. (2nd Edition). Berrett-Koehler Publishers: 2009.

- 68 Boehm, B. 'Value-based software engineering: reinventing', SIGSOFT Softw. Eng. Notes, 28, (2), (2003). pp. 3. 10.1145/638750.638775
- 69 Card, D. 'Building a Business Case for Process Improvement'. Proceedings of Europe SEPG, 2008.
- 70 Holton, E.F., Bates, R.A., and Ruona, W.E.A. 'Development of a generalized learning transfer system inventory', Human Resource Development Quarterly, 11, (4), (2000). pp. 333-360.
- 71 Shewhart, W.A. Economic control of quality of manufactured product. Milwaukee, Wis.: American Society for Quality Control: 1980.
- 72 Humphrey, W.S. Introduction to Software Process Improvement. Software Engineering Institute. 1992. CMU/SEI-92-TR-007 ESC-TR-92-007.
- 73 'ISO 9000:2008'. 2008.
- 74 Hoyle, D. ISO 9000 Quality Systems Handbook. (Fifth). Butterworth-Heinemann: 2005.
- 75 Tick It Guide <http://www.tickit.org/guide5-5.htm>. Available at, accessed 2 de Julio de 2011
- 76 International Standards Organization. 'ISO/IEC 15504-2:2004 Information technology Process assessment Part 2: A model for Process Management'. 2004.
- 77 Stelzer, D., Mellis, W., and Herzwurm, G. 'A critical look at ISO 9000 for software quality management', Software Quality Journal, 6, (2), (1997). pp. 65-79.
- 78 Paulk, M.C. 'Analyzing the Conceptual Relationship Between ISO/IEC 15504 (Software Process Assessment) and the Capability Maturity Model for Software'. Proceedings of, 1999.
- 79 Pyzdek, T. The Six Sigma Handbook: The Complete Guide for Greenbelts, Blackbelts, and Managers at All Levels. (Second Edition). McGraw-Hill: 2003.
- 80 Sivi, J.M.: 'Notes for the CMMI and SixSigma Masters Course', in Editor: 'Book Notes for the CMMI and SixSigma Masters Course' (2007, edn.), pp.
- 81 Poppendieck, M., and Poppendieck, T. Lean Software Development: An Agile Toolkit. Addison-Wesley Professional 2003.
- 82 Liker, J. The Toyota Way: 14 management principles. McGraw-Hill: 2003.
- 83 Sivi, J.M., and Forrester, E.C. 'Using Six Sigma to Accelerate the Adoption of CMMI for Optimal Results'. Proceedings of, 2004.
- 84 Sivi, J., Penn, M.L., and Harper, E. 'Relationships Between CMMI and Six Sigma', Software Engineering Institute, CMU/SEI-2005-TN-005, (2005). pp.
- 85 Antony, J., and Fergusson, C. 'Six Sigma in the software industry: results from a pilot study', Managerial Auditing Journal, 19, (8), (2004). pp. 1025-1032.
- 86 Pan, Z., Park, H., Baik, J., and Choi, H. 'A Six Sigma Framework for Software Process Improvements and its Implementation'. Proceedings of, 2007 446-453.
- 87 CMMI Product Team. CMMI for Development, Version 1.3. Improving processes for developing better products and services. Software Engineering Institute. 2010. CMU/SEI-2010-TR-033.
- 88 Leeson, P.: 'The Auditor as an Obstacle to Quality'. Proc. SEPG Europe, Munich, 10-13 June 2008 pp. Pages
- 89 Keefer, G., and Authority, A. 'The CMMI Considered Harmful For Quality Improvement And Supplier Selection', AVOCA GmbH, (2008). pp.
- 90 Glazer, H., Dalton, J., Anderson, D., Konrad, M., and Shrum, S. 'CMMI or Agile: Why Not Embrace Both?', Software Engineering Institute/Carnegie Mellon, November, (2008). pp.
- 91 'Standard CMMI® Appraisal Method for Process Improvement (SCAMPI) A, Version 1.2: Method Definition Document'. 2006.

- 92 Turner, R., and Jain, A. 'Agile meets CMMI: Culture clash or common cause?', LECTURE NOTES IN COMPUTER SCIENCE, (2002). pp. 153-165.
- 93 Sutherland, J., Jakobsen, C.R., and Johnson, K. 'Scrum and CMMI Level 5: The Magic Potion for Code Warriors', AGILE 2007, (2007). pp. 272-278.
- 94 Glazer, H. 'Love and Marriage: CMMI and Agile need each other', Crosstalk, Jan/Feb 2010, (2010). pp.
- 95 Staples, M., Niazi, M., Jeffery, R., Abrahams, A., Byatt, P., and Murphy, R. 'An exploratory study of why organizations do not adopt CMMI', The Journal of Systems & Software, 80, (6), (2007). pp. 883-895.
- 96 Nasir, M., Ahmad, R., and Hassan, N.H. 'An Empirical Study of Barriers in the Implementation of Software Process Improvement Project in Malaysia', Journal of Applied Sciences, 8, (23), (2008). pp. 4362-4368.
- 97 de Oliveira, S.B., Valle, R., and Mahler, C.F. 'A comparative analysis of CMMI software project management by Brazilian, Indian and Chinese companies', Software Quality Journal, 18, (2), (2009). pp. 177-194.
- 98 Santos, G., Montoni, M., Vasconcellos, J., Figueiredo, S., Cabral, R., Cerdeiral, C., Katsurayama, A.E., Lupo, P., Zanetti, D., and Rocha, A.R. 'Implementing Software Process Improvement Initiatives in Small and Medium-Size Enterprises in Brazil'. Proceedings of, 2007 187-198.
- 99 Pyster, A. 'What Beyond CMMI Is Needed to Help Assure Program and Project Success?', LECTURE NOTES IN COMPUTER SCIENCE, 3840, (2005). pp. 75.
- 100 Team, S.U. Standard CMMI® Appraisal Method for Process Improvement (SCAMPI) A, Version 1.3: Method Definition

Document. Carnegie Mellon University. 2011. CMU/SEI-2011-HB-001.

- 101 Vu, J.D. 'Process Improvement in retrospective (Lessons Learned from Software Projects)'. Proceedings of SEPG 2005, 2005.
- 102 Curtis, B., Kellner, M.I., and Over, J. 'Process modeling', (1992). pp.
- 103 CMMI Version 1.3 Product Suite. Available at <http://www.sei.cmu.edu/collaborating/spins/021009webinar.html>, accessed 3/03/2008
- 104 Kellner, M.I. 'Representation formalisms for software process modelling'. Proceedings of, 1988 93-96.
- 105 El Emam, K., and Madhavji, N.H. 'The Reliability of Measuring Organizational Maturity', 1, (1995). pp. 3-26.
- 106 Kirshnan, M.S., and Kellner, M.I. 'Measuring Process Consistency: Implications for Reducing Software Defects', IEEE Transactions on Software Engineering, 25, (6), (1999). pp. 800-815.
- 107 Kasunic, M. The State of Measurement Practice: Results of 2006 Survey. Carnegie Mellon University. 2006. CMU/SEI-2006-TR-009.
- 108 Harter, D.E., Krishnan, M.S., and Slaughter, S.A. 'The life cycle effects of software process improvement: a longitudinal analysis'. Proceedings of International Conference on Information systems, 1998 346-351.
- 109 Clark, B. 'Effects of Process Maturity on Development Effort', unpublished paper available at <http://sunset.usc.edu/bkclark/Research>, (1999). pp.
- 110 Boehm, B.W. Software cost estimation with Cocomo II. Upper Saddle River, NJ: Prentice Hall: 2000.
- 111 King, J., and Diaz, M. 'How CMM Impacts Quality, Productivity, Rework, and the Bottom Line', Crosstalk, (2002). pp.
- 112 Spiewak, R., and McRitchie, K. 'Using Software Quality Methods to Reduce Cost and Prevent Defects', Crosstalk, (December), (2008). pp.
- 113 Rico, D.F.: 'Software Process Improvement (SPI): Modeling Return on Investment (ROI)', in Editor: 'Book Software Process Improvement (SPI): Modeling Return on Investment (ROI)' (2002, edn.), pp.

- 114 Diaz, M., and Sligo, J. 'How Software Process Improvement Helped Motorola', IEEE Software, (1997). pp.
- 115 van Solingen, R. 'Measuring the ROI of software process improvement', 21, (3), (2004). pp. 32-38.
- 116 El Emam, K. The ROI from software quality. Boca Raton, Fla.: Auerbach Publications: 2005.
- 117 Wagner, S. 'Towards Software Quality Economics for Defect-Detection Techniques'. Proceedings of, 2005.
- 118 Rust, R.T., Zahorik, A.J., and Keiningham, T.L. 'Return on quality (ROQ): making service quality financially accountable', The Journal of Marketing, (1995). pp. 58-70.
- 119 Slaughter, S.A., Harter, D.E., and Krishnan, M.S. 'Evaluating the cost of software quality', 41, (8), (1998). pp. 67-73.
- 120 Buglione, L. 'Strengthening CMMI Maturity Levels with a Quantitative Approach to Root-Cause Analysis'. Proceedings of R5th Software Measurement European forum, 2008.
- 121 Buglione, L., and Abran, A. 'Introducing Root-Cause Analysis and Orthogonal Defect Classification at Lower CMMI Maturity Levels'. Proceedings of MENSURA, 2006 29.
- 122 Kalinowski, M., Mendes, E., Card, D., and Travassos, G. 'Applying DPPI: A Defect Causal Analysis Approach Using Bayesian Networks', Product-Focused Software Process Improvement, (2010). pp. 92-106.
- 123 Card, D.N.: 'Understanding Causal Systems', in Editor: 'Book Understanding Causal Systems' (2004, edn.), pp.
- 124 Endres, A. 'An analysis of errors and their causes in system programs', ACM SIGPLAN Notices, 10, (6), (1975). pp. 327-336.
- 125 Chillarege, R., Bhandari, I., Chaar, J., Halliday, M., Moebus, D., and Ray, B. 'Orthogonal Defect Classification-A Concept for In-Process Measurements', IEEE Computer, 18, (11), (1992). pp.
- 126 Card, D.N. 'Defect-causal analysis drives down error rates', IEEE Software, 10, (4), (1993). pp. 98-99.
- 127 Card, D.N. 'Learning from Our Mistakes with Defect Causal Analysis', IEEE Software, 15, (1), (1998). pp. 7.
- 128 Diane, K., and Terry, S.: 'A case study in the use of defect classification in inspections'. Proc. Proceedings of the 2001 conference of the Centre for Advanced Studies on Collaborative research, Toronto, Ontario, Canada 2001 pp. Pages
- 129 Microsoft Solution Framework for CMMI Process Guidance Available at <http://msdn2.microsoft.com/en-us/teamsystem/aa718802.aspx>
- 130 Beizer, B. Software Testing Techniques. (2a). New York: Van Nostrand Reinhold: 1990.
- 131 Grady, R.B. 'Software Failure Analysis for High-Return Process Improvement Decisions', 47, (1996). pp. 15-24.
- 132 Fredericks, M., Basili, V., Business, G., and Park, T. Using Defect Tracking and Analysis to Improve Software Quality. 1998.
- 133 Ishikawa, K. Guide to Quality Control. Tokyo, Japan: Asian Productivity Organization: 1986.
- 134 Bhandari, I., Halliday, M.J., Chaar, J., Chillarege, R., Jones, K., Atkinson, J.S., Lepori-Costello, C., Jasper, P.Y., Tarver, E.D., Lewis, C.C., and Yonezawa, M. 'In-process improvement through defect data interpretation', IBM Syst. J, 33, (1), (1994). pp. 182-214.
- 135 Peled, A., Salzman, L., Danon, A., and Rogoway, P. 'Defect Prevention Techniques for High Quality and Reduced Cycle Time'. Proceedings of EuroSPI1998, 1998.
- 136 Fredericks, M., and Basili, V. 'Using Defect Tracking and Analysis to Improve Software Quality', Crosstalk, (1999). pp.
- 137 Norman, E.F. 'A Critique of Software Defect Prediction Models', IEEE Transactions on Software Engineering, (1999). pp.

- 138 Leszak, M., Perry, D.E., and Stoll, D. 'A case study in root cause defect analysis', Proceedings of the 22nd international conference on Software engineering, (2000). pp. 428-437.
- 139 Jacobs, J.C., van Moll, J.H., Krause, P.J., Kusters, R.J., Trienekens, J.J.M., and Semiconductors, P. 'Effects of Virtual Development on Product Quality: Exploring Defect Causes', Software Technology and Engineering Practice, 2003. Eleventh Annual International Workshop on, (2003). pp. 6-15.
- 140 Bibi, S., Tsoumakas, G., Stamelos, I., and Vlahvas, I. 'Software Defect Prediction Using Regression via Classification', Computer Systems and Applications, 2006. IEEE International Conference on., (2006). pp. 330-336.
- 141 Carleton, A., and Nichols, W. 'Getting high performance out of high maturity'. Proceedings of SEPG Europe, 2008.
- 142 Card, D., N., Domzalski, K., and Davies, G. 'Making Statistics Part of Decision Making in an Engineering Organization', IEEE Software, 25, (3), (2008). pp. 37-47. <http://dx.doi.org/10.1109/MS.2008.66>
- 143 Goldenson, D.: 'The State of Software Measurement Practice'. Proc. Europe SEPG '08, Munich, Germany, 12-14, June 2008 pp. Pages
- 144 Radice, R. 'Survey Results of Baselines and Models Used by Level 4 and 5 Organizations'. Proceedings of SEPG 2010, 2010.
- 145 Wheeler, D.J. Understanding Variation: The Key to Managing Chaos. (Second Revised Edition). Knoxville, Tennessee: Spc Press Inc.: 1993.
- 146 Florac, W.A., Park, R.E., and Carleton, A.D. Practical Software Measurement: Measuring for Process Management and Improvement. Software Engineering Institute. 1997. CMU/SEI-97-HB-003.
- 147 Weller, E., Card, D., N., Curtis, B., and Raczynski, B. 'Point/Counterpoint: Applying SPC to Software development: Where and Why', IEEE Software, 25, (3), (2008). pp. 48-51. 10.1109/MS.2008.68
- 148 Senge, P. The Fifth Discipline: The Art & Practice of The Learning Organization. (1st). Broadway Books: 2006.
- 149 Madachy, R.J. Software Process Dynamics. Wiley-IEEE Press: 2008.
- 150 Anderson, D. Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results. Prentice Hall PTR: 2003.
- 151 Pfahl, D., and Lebsanft, K. 'Integration of system dynamics modelling with descriptive process modelling and goal-oriented measurement', Journal of Systems and Software, 46, (2-3), (1999). pp. 135-150.
- 152 Abdel-Hamid, T., and Madnick, S. Software Project Dynamics: An Integrated Approach. Prentice Hall PTR: 1991.
- 153 Pfahl, D., Klemm, M., and Ruhe, G. 'A CBT module with integrated simulation component for software project management education and training', Journal of Systems and Software, 59, (3), (2001). pp. 283-298.
- 154 Barros, M.O., Werner, C.M.L., and Travassos, G.H. 'Using process modeling and dynamic simulation to support software process quality management'. Proceedings of XIV Simpósio Brasileiro de Engenharia de Software, Workshop de Qualidade de Softwarw, 2000.
- 155 Müller, M., and Pfahl, D. 'Simulation Methods', in Forrest Shull, Singer, Janice, and Sjoberg, Dag I.K. (Eds.): 'Guide to Advances Empirical Software Engineering' (Springer, 2008)
- 156 High Performance Systems. An Introduction to Systems Thinking. High Performance Systems: 2008.
- 157 Forrester, J.W. Urban dynamics. MIT Press: 1969.
- 158 Ambler, S.W. The elements of UML 2.0 style. Cambridge [U.K.] ; New York: Cambridge University Press: 2005.

- 159 Schwaber, K. Agile project management with Scrum. Redmond, Wash.: Microsoft Press: 2004.
- 160 Informatica: Industria de desempleo cero. Available at <http://www.cuti.org.uy/Portals/0/PRESENTACION%20CONFERENCIA%20DE%20PRENSA%2015%2006.ppt>, accessed 19/05/2009
- 161 Smith, C. 'Achieving organizational training objectives with short course development', 13th Conference on Software Engineering Education & Training, 2000. Proceedings. , (2000). pp. 32-38.
- 162 Freimut, B., Denger, C., and Ketterer, M. 'An industrial case study of implementing and validating defect classification for process improvement and quality management', Software Metrics, 2005. 11th IEEE International Symposium, (2005). pp. 10.
- 163 Henningsson, K., and Wohlin, C. 'Assuring fault classification agreement-an empirical evaluation', Empirical Software Engineering, 2004. ISESE'04. Proceedings. 2004 International Symposium on, (2004). pp. 95-104.
- 164 Cohen, J. 'A Coefficient of Agreement for Nominal Scales', Educational and Psychological Measurement, 20, (1), (1960). pp. 37.
- 165 Fleiss, J.L. 'Measuring nominal scale agreement among many raters', Psychological Bulletin, 76, (5), (1971). pp. 378-382.
- 166 Landis, J.R., and Koch, G.G. 'The Measurement of Observer Agreement for Categorical Data', 33, (1), (1977). pp. 159-174.
- 167 Devins, D., Johnson, S., and Sutherland, J. 'Different skills and their different effects on personal development', Journal of European Industrial Training, 28, (1), (2004). pp.
- 168 Rus, V., Sajjan, S.G., and Mohammed, S. 'Automatic Clustering of Defect Reports'. Proceedings of International Conference on Software Engineering and Knowledge Engineering. , 2008 291-296.
- 169 Rus, V., and Shiva, S.G. 'A general framework for quantitative software testing'. Proceedings of First International Workshop on advances and Innovations in software Testing, 2007.
- 170 Jain, A.K., Murty, M.N., and Flynn, P.J. 'Data clustering: a review', ACM computing surveys, 31, (3), (1999). pp.