

Propuesta de Mejora del Proceso de Testing en Pequeñas Empresas

Francisco Oyarce Valderrama
Escuela de Ingeniería Informática
Pontificia Universidad Católica de Valparaíso
Valparaíso, Chile
oyarcef@gmail.com

Rodolfo Villarroel Acevedo
Escuela de Ingeniería Informática
Pontificia Universidad Católica de Valparaíso
Valparaíso, Chile
rodolfo.villarroel@ucv.cl

Resumen—Muchos modelos y estándares para la mejora de procesos de testing de software no están diseñados considerando la realidad de pequeñas organizaciones. La desventaja que tienen es que están dirigidos a grandes organizaciones, y adaptarlos a una pequeña empresa puede resultar complejo y excesivamente costoso para su realidad, por lo que se implementan de forma inadecuada o su implementación es descartada debido a las dificultades que trae. Por este motivo, se propone un modelo de mejora para el testing en pequeñas empresas, el cual ha sido desarrollado en base al estudio de los modelos de mejoramiento de testing más utilizados, analizando sus semejanzas y algunas experiencias. Dicha propuesta ha sido aplicada en una experiencia práctica, permitiendo reflejar bien la situación inicial y determinar el camino a seguir para alcanzar las mejoras que la organización deseaba.

Palabras Clave- testing, certificaciones, pequeña empresa, modelo de mejora.

I. INTRODUCCIÓN

El proceso de testing, permite verificar y determinar si los productos desarrollados cumplen con el fin para el que fueron creados. Sin embargo, además de determinar si el producto realiza las tareas para los cuales fue construido, el proceso de testing puede evaluar la calidad del producto y de los procesos involucrados en su desarrollo. Por lo que, someter al software a un proceso completo y adecuado de testing es de suma relevancia para poder contar con un producto de alto nivel.

Las organizaciones siguen la tendencia de certificar sus procesos, y en consecuencia la calidad de sus productos, a través de la adopción de estándares de distintos aspectos de calidad. En la industria del software se ve que existen una serie de certificaciones, normas y modelos (ej: ISO o CMMI) que permiten corroborar la calidad pretendida en los procesos de desarrollo. La gran desventaja, desde un cierto punto de vista, de estas certificaciones es que están dirigidas a grandes organizaciones y adaptarlas a una pequeña empresa puede resultar complejo y excesivamente costoso para su realidad [1].

Debido a esta última situación, es que resulta necesario generar un modelo que permita mejorar el proceso de testing en las pequeñas empresas, de tal manera que se aseguren los procedimientos de pruebas de sus productos, y así brindar

mayor calidad en sus proyectos. Para esto, se necesita analizar los modelos existentes y establecer sus semejanzas y discrepancias y ver como éstos pueden ser aplicados en el contexto de una organización pequeña. De esto nace una propuesta de modelo de mejoras para el testing, la cual junto a su modelo de diagnóstico, ha sido implementada en un caso práctico para obtener los resultados pertinentes de los alcances del modelo.

Este trabajo plantea la creación de un modelo para mejorar el testing en pequeñas empresas, para esto en la sección II se presenta una comparación entre algunos modelos de mejoras actualmente utilizados, para posteriormente, en la sección III, presentar el modelo de mejoras desarrollado en base al trabajo realizado. Se presenta también un caso práctico en la sección IV con la implementación del modelo, para finalizar con las conclusiones.

II. COMPARATIVA DE MODELOS EXISTENTES

Los modelos de testing considerados para la comparación son los más usados actualmente más algunas nuevas propuestas que están surgiendo, éstos son: TMM [2], TMMi [3], TIM [4], TestPAI [5], Tutelkán [6] [7], ISO 29119 [8] y TPI [9]. En la Tabla I se presenta un esquema comparativo entre ellos con el objetivo de poder obtener las ventajas que un modelo puede ofrecer sobre otro en distintos aspectos relacionados a la mejora y gestión del testing.

TMM entrega un marco de trabajo para guiar la mejora de prácticas basadas en información y relaciones con áreas claves de testing. El objetivo es identificar los obstáculos que la organización debe superar para realizar un mejor testing.

TMMi es un marco de trabajo dirigido a la mejora de procesos de testing y complementario a CMMI. Utiliza una escala con niveles de madurez para medir las mejoras conseguidas en la organización, regidas por sus metas y prácticas correspondientes.

TIM es una guía para la mejora de procesos de pruebas que usa de base CMM y TMM. Este modelo está dirigido a cinco áreas clave: Organización, Planificación y Seguimiento, Casos de pruebas, Testware y Comentarios.

TestPAI se define como un área de proceso de pruebas integrado con CMMI. Se sitúa en el nivel 3, junto a los

procesos de Ingeniería. TestPAI define 5 metas específicas, cada una se logra a través de sus prácticas específicas.

TUTELKAN es un proyecto de mejora de procesos que incluye una plantilla llamada “Plan de Testing” que se plantea como una guía para el testing de cada proyecto.

ISO 29119 tiene como principal objetivo generar un estándar que reúna vocabulario, procesos, documentación y técnicas para el ciclo de vida del testing.

TPI fue desarrollado basándose en conocimiento práctico, ofreciendo un punto de vista de la madurez de los procesos de testing dentro de la organización. Es así, que este modelo ayuda a definir pasos para una mejora gradual y controlada de los procesos de testing.

TABLA I COMPARACIÓN DE MODELOS ASOCIADOS A TESTING

	TMM	TMMi	TIM	TPI	TestPAI	Tutelkán	ISO 29119
Pequeña Empresa					✓	✓	✓
Ambiente	✓	✓	✓			✓	
Planeación	✓	✓	✓	✓	✓	✓	✓
Información/Documentación	✓		✓	✓		✓	✓
Diseño de pruebas	✓	✓	✓		✓		
Testware	✓		✓	✓		✓	
Política Organizacional		✓	✓	✓	✓		✓
Control de testing		✓	✓		✓		
Ejecución de testing		✓			✓		✓
Organización de personal		✓	✓			✓	
Capacitación de personal		✓	✓	✓			
Integración con el ciclo de vida		✓	✓				✓
Testing no funcional		✓		✓		✓	✓
Métricas		✓	✓				
Prevención de defectos	✓		✓	✓			
Optimización continua	✓	✓	✓				
Reutilización de testing			✓	✓	✓	✓	
Gestión de riesgos			✓		✓		✓
Necesidades organizacionales	✓	✓	✓		✓		
Revisiones estáticas			✓	✓		✓	✓
Pruebas funcionales	✓	✓	✓	✓	✓	✓	✓
Adaptación al ciclo de vida				✓			✓
Comunicación	✓	✓	✓	✓			
Testing de bajo nivel	✓	✓	✓	✓			

III. MODELO PROPUESTO

El modelo de mejora propuesto se basa en siete secciones de aplicación: Política Organizacional, Testing de Bajo Nivel, Diseño, Integración al proyecto, Prevención, Testing no Funcional y Mejora Continua. Las secciones de aplicación se proponen en un cierto orden de aplicación, sin embargo éste no es estricto y puede variar dependiendo de las condiciones y necesidades de cada organización. Además, existe una relación bidireccional entre cada sección, pues en la medida que se trabajen nuevas secciones las anteriores pueden requerir modificaciones para alcanzar una coordinación a nivel total de la empresa. El trabajo de mejoras en las distintas secciones está basado en dos recursos fundamentales de toda organización: Organización

de Personal y Documentación. Estos recursos son elegidos tomando en cuenta que son los activos más accesibles en las organizaciones y permitirán una adecuada implementación y formalización de las actividades de mejoras de testing en una pequeña empresa. Además, se presenta el modelo de diagnóstico correspondiente que permite conocer la situación inicial del proceso de testing en una organización con el fin de determinar el trabajo necesario para implementar el modelo de mejoras.

En la Figura 1 se muestra el esquema que representa el modelo propuesto. Las secciones de mejora se relacionan de manera bidireccional, ya que los cambios en una sección pueden afectar a otra. Toda sección debe ser gestionada en base al uso eficiente de los dos recursos fundamentales.



Figura 1 Esquema de modelo propuesto

A. Recursos Fundamentales

1) *Personal*: Se entiende que el equipo humano son las personas encargadas de trabajar en las mejoras del testing. Es por esto, que una adecuada organización del personal y sus recursos permitirá que las mejoras se puedan implementar de forma correcta y que el camino a la mejora se complete y no quede detenido por falta de personal o por el desconocimiento de roles de los trabajadores en el mejoramiento. La gestión del personal es relevante, ya que es necesario lograr que el equipo se comprometa con el proceso de mejoramiento y que muchas de sus prácticas habituales sean corregidas para alcanzar el estado deseado.

2) *Documentación*: Formalizando cada uno de los cambios y acciones realizadas, se puede mantener una comunicación y retroalimentación entre todas las partes involucradas, es por esto que en cada una de las etapas se debe implementar un adecuado método de documentación que permita cumplir con este objetivo. Una adecuada documentación permitirá que en la medida que nuevas personas se unan a la organización, éstas puedan comprender de forma correcta cómo se realizan las pruebas.

B. Secciones de Mejora

La elección de las secciones de mejora se realizó en base a adaptaciones de los distintos modelos descritos señalados en la sección II y experiencias como las de [1].

Se debe señalar que al ser un modelo de mejoras, se asume que ya existe un proceso de testing en la organización a ser aplicado y no se pretende comenzar el testing desde cero, es decir, en la medida que el modelo se aplique las viejas prácticas deberán convivir con las prácticas propuestas. Lo anterior, quiere decir que el estar enfocando los esfuerzos en una determinada área del testing no implica que el resto de los conceptos o áreas no se estén implementando de alguna manera.

Las secciones de mejora se presentan con una descripción de lo que representan en el testing y su eventual

relación con otras áreas, luego se analizan sus relaciones con las prácticas fundamentales, y finalmente se recomiendan algunas actividades para su gestión.

1) *Política Organizacional*: La política organizacional es la base del proceso de mejoramiento del testing, pues la política organizacional determina qué es lo que se quiere mejorar, cuáles son las metas y expectativas de la mejora, con qué recursos se cuenta para mejorar el testing, es decir, se define la política organizacional respecto a las mejoras. La política organizacional podrá variar en la medida de que se mejoren las demás áreas de mejoras y que las tecnologías varíen y permitan la inclusión de nuevas herramientas para los procesos de pruebas.

Relación con Recurso Personal: Dependiendo de la Política Organizacional adoptada se deberá organizar el personal disponible para la mejora del testing. Una propuesta inicial es que la política organizacional sea capaz de instaurar una nueva forma de ver las pruebas en el proceso de desarrollo. Por otro lado, se deben definir de manera clara los roles de los distintos actores en el ciclo de desarrollo y qué tareas le corresponde a cada uno en la mejora del testing.

Relación con Recurso Documentación: La política organizacional deberá definir de qué manera se documentan los hechos y trabajos que se realicen para las mejoras, de tal manera que la documentación permita la reutilización de casos de pruebas, por ejemplo, en futuros proyectos. Además, los errores cometidos en la mejora y ejecución de pruebas deben ser archivados y estar disponibles para evitar repetir esfuerzos ya realizados.

Actividades Recomendadas:

- Establecer políticas y objetivos del mejoramiento de testing
- Establecer política de planificación de pruebas
- Definición de estándares
- Política de organización de personal
- Política de organización de recursos

- Definición de evaluación de mejoras
- Control de política organizacional

2) *Testing de Bajo Nivel:* Una primera propuesta es mejorar el testing de bajo nivel, en específico el testing unitario. La idea es que en esta área se permita a los desarrolladores contar con un grado de libertad tal, que los mismos programadores puedan definir la forma en que realizan las pruebas unitarias a sus módulos desarrollados. La idea es que los programadores prueben debidamente sus trabajos antes de liberarlos y que sean sometidos a pruebas de integración y de sistema. Se debe permitir al desarrollador decidir qué herramientas implementar (dentro de los recursos disponibles), el tiempo dedicado y a la vez, en medida de los resultados obtenidos, poder ayudar en la mejora del testing de mayor nivel.

Relación con Recurso Personal: La política organizacional deberá definir con claridad al personal a cargo de esta sección de mejor, es decir, en la medida que evolucione esta sección se debe verificar la política establecida. Se debe contemplar los recursos disponibles para el personal en la mejora del testing unitario, como por ejemplo la compra de herramientas de software o capacitación pagada para los testers. Además, se puede establecer como estrategia de mejora la disposición de algún tiempo libre de trabajo de los testers para poder dedicarlo a la elaboración de ideas o de estudio sobre mejoras del testing y su trabajo.

Relación con Recurso Documentación: La documentación es vital para la mantención de las mejoras que se logren con la gestión de mejoras de las pruebas unitarias. Las estrategias decididas por los trabajadores, la utilización de software y otras herramientas deben ser reportadas e idealmente poder ser explicadas a otros miembros del equipo con el fin de generar una retroalimentación y avanzar juntos en la mejora del testing. Una correcta aplicación de la documentación puede permitir con el tiempo la reutilización de material y una mejor adaptación al nuevo personal que pueda integrarse a la organización.

Actividades Recomendadas:

- Capacitaciones individuales
- Capacitaciones organizacionales
- Evaluaciones de pares
- Herramientas de apoyo
- Documentación de testing
- Planeación de testing

3) *Diseño:* Las pruebas de mayor nivel (de integración, de sistema y de aceptación) requieren de mayor preparación antes de ser ejecutadas, ya que deben representar las necesidades y requerimientos del usuario para poder dar de alta al sistema que se está desarrollando. Los requisitos del usuario tienen un rol fundamental en el diseño de las pruebas de alto nivel, ya que éstas definen si el sistema que se está desarrollando cumple con las necesidades y expectativas del cliente. Debido a esto, el diseño deberá

estar relacionado con el levantamiento de requisitos que los analistas realicen al comienzo del proyecto.

Relación con Recurso Personal: Para el personal a cargo del diseño de pruebas es ideal que sea con dedicación exclusiva a las tareas de testing. Lo recomendado es que en la medida que las mejoras se vayan implementando en la organización, aumentar el personal en ésta área o invertir en mejorar la preparación de éste.

Relación con Recurso Documentación: Todo diseño bien documentado mejora la explicación de éste, es más si los diseños están bien documentado podrán ser reutilizados en trabajos futuros. La documentación debe incluir casos de pruebas, especificaciones técnicas sobre su implementación y otros aspectos.

Actividades Recomendadas:

- Diseño de casos de pruebas
- Documentación pruebas-requerimientos
- Especificación de procedimientos de ejecución
- Control de diseño

4) *Integración al Proyecto:* Hasta las áreas anteriores se ha subentendido al testing como una etapa que se hace presente en el ciclo de desarrollo a partir de la implementación. Se entenderá, entonces, por integración al concepto de hacer del testing una etapa transversal del ciclo de desarrollo, es decir, el testing debe ser considerado desde la toma de requerimientos hasta la ejecución de las pruebas y la eventual aprobación por parte del cliente. La integración trae como consecuencia una integración entre distintos niveles del ciclo de desarrollo, y que permitirá poder mejorar la comunicación a nivel general de la organización.

Relación con Recurso Personal: Se necesita gran comunicación entre los distintos niveles del proyecto y serán necesarias reuniones de coordinación para proyectos específicos y de preparación para futuros clientes. No se requiere, en principio, el aumento de recurso humano, sino más bien potenciar sus habilidades blandas de manera de lograr acuerdos en grupos de trabajo que normalmente no estén muy integrados.

Relación con Recurso Documentación: La documentación será necesaria para formalizar los acuerdos que puedan llegar a tomarse entre los distintos grupos de trabajo, lo que se podrá ver reflejado, además, en cambios en la Política Organizacional. A partir de la mejora de esta área la documentación de los proyectos estará orientada más hacia el desarrollo en base a la aceptación de los requerimientos de los clientes y cómo se medirán éstos. Por lo tanto, un documento oficial de métricas se hace necesario.

Actividades Recomendadas:

- Reuniones de organización entre distintos niveles
- Revisiones avanzadas de pares
- Diseño de pruebas basado en requerimientos
- Documento de métricas

5) *Prevención*: Por prevención se entenderá al enfoque que hará del testing un proceso, que además de detectar fallas en los sistemas que se están desarrollando, permita prevenir los defectos desde la concepción misma de los proyectos. Por lo tanto, resulta evidente la cercanía que tiene esta área con la el Diseño e Integración. La prevención requiere de una nueva forma de comprender el testing y la utilidad que éste presta al desarrollo. El gran peligro, es creer que una adecuada prevención implica la inexistencia de errores, pues como bien se sabe todo producto software tiene errores en su implementación. Es más, en la medida que la prevención se vaya adaptando al proceso de desarrollo, se hará necesario que el diseño se dedique a idear casos de pruebas más específicos que busquen poner a prueba de un modo más agresivo al sistema.

Relación con Recurso Personal: No se requieren grandes variaciones en la organización del personal de la organización, sin embargo se debe considerar la posibilidad de separar al equipo actual de diseño de testing, en un grupo encargado de la prevención de fallas y otro para la mejora de pruebas en busca de errores durante las pruebas en sí.

Relación con Recurso Documentación: Está muy relacionado con los cambios que se definan en la política organizacional sobre cómo almacenar los métodos que permitirán adelantarse a la fallas que los sistemas puedan presentar. Resulta muy importante almacenar en forma histórica errores que se detecten en módulos que por estrategia de testing se puedan considerar libres de fallas, pero que al ser testeados en pruebas de sistema se descubran problemas no contemplados.

Actividades Recomendadas:

- Rotación de equipos (prevención-detección)
- Capacitación en prevención
- Revisión de pares
- Creación de casos de pruebas “agresivos”

6) *Testing no funcional*: El testing no funcional, además de evaluar el cumplimiento de requerimientos no funcionales, permite darle mayor valor al sistema desarrollado, pues las pruebas no funcionales pueden demostrar fortalezas del sistema en cuanto a aspectos como la robustez, confiabilidad, usabilidad y otros aspectos relacionados a la calidad. Dependiendo de los recursos disponibles para la realización de estas tareas, debe definirse hasta qué punto se quiere testear el sistema por sobre los aspectos no funcionales explícitamente considerados en el levantamiento de requisitos.

Relación con Recurso Personal: El personal debe ser dotado de las herramientas que permitan realizar en forma correcta el testing no funcional, los recursos destinados a estas pruebas dependerán del alcance que tengan. Eventualmente, si la organización no cuenta con personal experto en el tema, debe considerar la posibilidad de recibir apoyo de alguien especializado.

Relación con Recurso Documentación: Se deberán formalizar las pruebas que se realicen y sus resultados, así como establecer métricas para evaluar aspectos no funcionales que deben estar claramente definidas a nivel organizacional. Es fundamental para el equipo de esta área contar con los documentos formales de requerimientos del proyecto de manera de conocer en forma clara los aspectos a evaluar para cumplir con las solicitudes del cliente.

Actividades Recomendadas:

- Métricas de pruebas no funcionales
- Diseño de testing no funcional
- Estrategia organizacional de testing no funcional
- Implementación de herramientas
- Reuniones de colaboración interdisciplinaria

7) *Mejora Continua*: Las mejoras alcanzadas en el testing deben controlarse periódicamente y los esfuerzos deben apuntar a mantener los cambios y buscar nuevas alternativas para obtener nuevas mejoras. La organización no debe conformarse con los logros alcanzados, ni debe dar por superada sus problemáticas de testing. El trabajo de mantenimiento debiese apuntar a la prevención y no en corregir fallos descubiertos por el uso del sistema en su ambiente de explotación.

Relación con Recurso Personal: La organización debe fomentar la comunicación entre todo el equipo involucrado en la mejora de testing y en caso de que exista desvinculación laboral de personal se deberá buscar un reemplazo que esté dispuesto a hacerse parte de los procesos de mejoras establecidos. En la medida que este modelo esté plenamente implementado puede evaluarse la posibilidad de realizar rotación entre los distintos roles, siempre que las habilidades técnicas del personal lo permita, ya que permitirá al personal tener una visión más amplia de las mejoras.

Relación con Recurso Documentación: La formalización de todas las mejoras, causas y efectos de problemas estarán documentados. Esto reflejará la condición de tomar las decisiones basándose en hechos propuestos para esta área. Las métricas establecidas para la evaluación de los distintos criterios de calidad deben quedar disponibles en un repositorio para todo el personal y el desarrollo de los proyectos es documentado en forma progresiva

Actividades Recomendadas:

- Rotación de roles
- Reuniones de evaluación
- Mantenimiento preventivo
- Capacitación constante
- Encuestas de satisfacción
- Determinación de valor agregado

C. Modelo de Diagnóstico

Para la correcta aplicación de las mejoras, el modelo de diagnóstico propuesto consta de tres fases: Evaluación de la organización, propuesta de mejoras y el acuerdo de mejoras.

1) *Evaluación de la Organización:* Esta fase es la encargada de obtener los datos necesarios sobre el estado de la organización. Esto, se hace evaluando por separado el estado organizacional de cada una de las secciones de aplicación, además de un estudio global del entorno organizacional. Se requiere de un evaluador único que dirija el proceso de evaluación y recopile la información pertinente del caso. El proceso se basa principalmente en la resolución de un breve cuestionario que permite obtener una respuesta certera (sí o no) para luego extraer datos adicionales. Estos cuestionarios buscan establecer la proporción de actividades recomendadas por cada sección de mejora que se están actualmente aplicando en la organización, y así obtener una idea de cómo se encuentra aplicada la sección en general

2) *Propuesta de Mejora:* Una vez terminada la etapa de evaluación, los datos deben ser trabajados y comparados con el modelo de mejoras. Se determina cuáles aspectos del modelo están siendo trabajados, en qué grado, y cuáles no. Lo anterior, se traduce en un informe de propuestas destinado a la organización, el que contiene aspectos a mejorar, determinando plazos de trabajo, recursos y otros aspectos relevantes. Los contenidos mínimos de este informe deben ser:

- Recursos estimados para la ejecución del plan de mejoras
- Fortalezas y debilidades por cada sección de aplicación
- Fortalezas y debilidades a nivel organizacional
- Plan de aplicación de mejoras por sección (pueden trabajarse simultáneamente más de una sección)

3) *Acuerdo de Mejoras:* Una vez que el informe con la "Propuesta de Mejora" es revisado por la organización, se debe decidir en qué aspectos del testing se está dispuesto a invertir los recursos. Al tratarse de un modelo desarrollado para pequeñas empresas, la flexibilidad de éste es fundamental para su implementación en forma correcta. Por lo tanto, en gran parte depende del criterio de los encargados organizacionales.

IV. UN CASO PRÁCTICO

La organización en que se aplicó el modelo es una empresa formada hace cinco años originada para entregar servicios a empresas en las áreas de redes informáticas, diseño y tecnologías de la información y en el último tiempo en el desarrollo de software a medida. El área de desarrollo está constituida por dos analistas y diseñadores, y por otra parte, se cuenta con un equipo en promedio de tres programadores que son quienes ejecutan el testing. Entre sus principales clientes se encuentra el rubro hotelero, firmas de abogados y el centro de formación técnica de una universidad regional.

Con el fin de determinar el alcance del modelo posterior a su implementación se presentan los resultados obtenidos. Estos resultados se basan en el impacto generado en la

organización y su personal, resultados de productos e impacto en clientes (usuarios).

A. Diagnóstico

1) *Modelo de Evaluación:* Para la evaluación se aplicó el modelo de diagnóstico (cuestionarios) a cuatro personas del equipo de trabajo. En resumen, el proceso es que los programadores se encargan de probar individualmente sus módulos, para luego hacer en conjunto a los analistas-diseñadores pruebas de sistema (sin planificación y sin registros relevantes de sus resultados) para pasar a un período de marcha blanca con el cliente que por lo general termina siendo la entrega de un prototipo, pues en este período se detectan cambios o problemas de requerimientos y el sistema debe ser modificado y nuevamente probado en marcha blanca por el usuario antes de realizar la entrega final. A nivel general, queda claro que el proceso de testing es pobre con respecto al modelo de mejoras establecido y es una actividad no valorada.

2) *Propuesta de Mejora:* La Tabla II pretende entregar en forma resumida el grado de aplicación de cada una de las secciones de aplicación, considerando qué actividades recomendadas se realizan.

TABLA II APLICACIÓN DE SECCIONES DE MEJORA

SECCIÓN DE APLICACIÓN	ACTIVIDADES REALIZADAS	ACTIVIDADES NO REALIZADAS
Política Organizacional	8,33%	91,67%
Testing de Bajo Nivel	45%	55%
Diseño	6,25%	93,75%
Integración al Proyecto	20%	80%
Prevención	5%	95%
Testing No Funcional	20,83%	79,17%

Luego de conversar con los dueños de la empresa para establecer el plan de trabajo se ha llegado al siguiente acuerdo de trabajo:

- No se destinarán recursos económicos para la adquisición de herramientas ni de personal
- El testing de bajo nivel será inmediatamente mejorado
- El testing no se incluirá formalmente al proceso entero de desarrollo, sin embargo existe el

compromiso de que el testing esté relacionado plenamente a los requerimientos del usuario

- El testing no funcional se formalizará. Dentro de un proceso de mantenimiento preventivo se incluirá la revisión de estos aspectos por parte de los usuarios finales

B. Trabajo Práctico

Se enumeran las actividades realizadas por cada sección de mejora del modelo.

1) Política Organizacional

- Definir objetivos para el proceso de testing
- Definir política de planificación de pruebas
- Definir estándares de testing
- Establecer recursos de tiempo para las mejoras
- Establecer personal a participar de las mejoras

2) Testing de Bajo Nivel

- Adquisición de herramientas para automatización de pruebas
- Registrar y documentar pruebas
- Elaboración de tipo de documentación
- Establecer calendarios de pruebas

3) Diseño

- Establecer un esquema para diseñar pruebas en base a requerimientos

4) Integración al Proyecto

- Establecer un documento de métricas para la aprobación de los requerimientos

5) Prevención

- Generar planificación de pruebas en fases de diseño
- Desarrollar modelos de pruebas que permitan encontrar errores críticos del sistema

6) Testing No Funcional

- Establecer políticas sobre atributos no funcionales a entregar a clientes
- Especificaciones de entrega
- Criterios de usabilidad
- Criterios de eficiencia
- Criterios de fiabilidad
- Criterios de portabilidad
- Establecer planes de mantenimiento preventivo

7) Mejora Continua

- Establecer un proceso periódico de evaluación para el proceso de testing

C. Resultados

Antes de analizar los resultados obtenidos con la implementación del modelo de mejoras se debe hacer un resumen cronológico sobre cómo se llevó a cabo el trabajo. Lo primero, como resulta evidente fue el área de Política Organizacional, lo cual no llevó más allá de una semana en donde se tomaron las decisiones necesarias, posteriormente se mejoró el testing de bajo nivel aplicando las actividades planteadas en los proyectos que ya se encontraban en

desarrollo, y posteriormente sumando nuevos. De forma simultánea se gestionaron dos áreas del modelo de mejoras: Testing No Funcional y Diseño, esto porque en cuanto a características no funcionales se podía avanzar sin la necesidad de estar en un proyecto en particular. Por otro lado, las normas de diseño tampoco requerían enfrentar un proyecto en particular por lo que se pudo terminar sin grandes inconvenientes estas dos áreas. Finalmente, y en el siguiente orden, se gestionaron las áreas restantes: Integración al Proyecto, Prevención y Mejora Continua.

1) *Ciclo Definido de Testing*: Si se compara la situación inicial del proceso de testing existente con la situación lograda tras la aplicación del modelo de mejoras, se encontrarán diferencias determinantes en cuanto al nivel de procesos de desarrollos de los productos. La organización no poseía un proceso de testing definido dentro de su ciclo de desarrollo, ya que solo se dedicaban tiempos a pruebas, pero éstos no debían cumplir ningún estándar, políticas o expectativas planteadas. Tras la aplicación del modelo de mejoras, se ha logrado consolidar un proceso de testing definido en tiempo, alcance y otros aspectos relevantes. La organización ya cuenta con un proceso formal y los productos desarrollados son sometidos a éste.

2) *Disminución de Repetición de Trabajo*: Al documentar las pruebas, tanto en su planeación, diseño y ejecución, se cuenta con información valiosa para futuros proyectos. Este aspecto se pudo comprobar inmediatamente en los proyectos que fueron parte de la implementación de las mejoras, ya que permitió la reutilización de casos de prueba en piezas de software vinculadas; y la omisión de pruebas en piezas idénticas de código reutilizado.

3) *Disminución de costo en detección de errores*: Debido a la aplicación de pruebas más rigurosas, mejor planeadas y diseñadas, se han podido descubrir errores de implementación con mayor frecuencia, si bien esto implica un costo de re-implementación, es un costo muy inferior al tener que lidiar con un cliente insatisfecho o que descubre fallas una vez que le es entregado el producto.

4) *Baja en insatisfacción de clientes*: Ligado al punto anterior la baja en la insatisfacción de clientes no solo se ha visto reflejada por el no descubrimientos de fallas por parte de ellos. Si el sistema cumple con sus requerimientos los usuarios estarán relativamente satisfechos, pero si además se hacen los esfuerzos por verificar que se cumplan los mínimos aspectos no funcionales, los usuarios tienen una actitud aún más positiva con el nuevo sistema.

5) *Mejora parcial en la valoración del testing*: El convencimiento por parte de la organización, principalmente por los programadores, aún no es total. Esto se debe principalmente a la burocratización que implica la adopción o mejora de un proceso de testing. En principio se tuvo que elaborar todos los formatos de documentación, aprender a seguir las reglas, documentar todo, rehacer trabajo por no

cumplir los acuerdos y otros aspectos que implican desmotivación por parte del personal.

6) *Retraso en desarrollo de productos*: Este aspecto se aborda de manera independiente al punto anterior, pues indiscutiblemente los proyectos (sobre todo los primeros) se vieron retrasados por la implementación de una etapa formal de testing en el ciclo de desarrollo. Sin embargo, este aumento en el ciclo de producción solo es notorio en las primeras aplicaciones de testing, pues en la medida que existan más antecedentes previos a las pruebas, los tiempos se van recuperando, como ya ha sucedido con los proyectos web en donde se dejan de cometer errores históricos.

V. CONCLUSIONES

Con esta investigación se ha visualizado que los modelos, en general, tienen componentes y estructuras similares. Es común la utilización de escalas para calificar el estado de las organizaciones y sus mejoras. Además, existen relaciones explícitas entre modelos, por ejemplo cuando un modelo se elaboró basándose en otro. Por otro lado, existen modelos que en una primera impresión pueden resultar complejos debido a la gran cantidad de aspectos y tareas que implican, sin embargo las propuestas que entregan son sumamente interesantes y merecen ser analizadas para idear una propuesta específica.

El modelo de diagnóstico resultó práctico para esta implementación, debido a que pudo reflejar bien la situación inicial y determinar el camino a seguir para alcanzar las mejoras que la organización deseaba. Sin embargo, se plantea la necesidad de conjugar de mejor manera la propuesta de mejoras que el encargado hace con las mejoras que la organización finalmente acepte.

Los desafíos más importantes se pueden dar en el ámbito de lograr que el modelo de mejoras pueda ser aplicado más ágilmente, con el fin de no causar la impresión de que se están burocratizando los procesos, aunque esto depende del proceso de testing a mejorar. En este caso, el proceso de

testing mejorado era bastante básico y poco formal, por lo que el modelo debió aplicarse en una organización poco estructurada en este sentido.

Otro desafío importante para seguir comprobando la eficacia del modelo sería poder aplicarlo en otras organizaciones de características similares en cuanto a personal y recursos, pero con otros estilos de trabajo. Así, se podría ver la capacidad real de la adaptación a distintos entornos del modelo. Es por esto, que el modelo debería seguir mejorando constantemente en función de las distintas realidades en la que sea aplicado.

REFERENCIAS

- [1] Tanja Vos, Jorge Sánchez, Maximiliano Mannise, “*Mejorando el testeado en las PYME. ¿Cómo empezar?*”, Software Quality Usability and Certification, Proceedings JTS 2008, páginas 71-80.
- [2] David Gelperin y Alden Hayashi, “*How to support better software testing*”, Application Development Trends, Mayo 1996, páginas 42-48.
- [3] *Test Maturity Model Integration (TMMi) Version 2.0*, TMMi Foundation.
- [4] Thomas Ericson, Anders Subotic y Stig Ursing, *TIM – A Test Improvement Model*, 1997.
- [5] “TestPAI: Un Área de Proceso de Ingeniería de Nivel de Madurez 3”.
- [6] Gonzalo Valdés, Hernán Astudillo, Marcello Visconti, Claudia López, “The Tutelkan SPI Framework for Small Settings: A Methodology Transfer Vehicle. 17th EuroSPI Conference, Grenoble Institute of Technology, France, septiembre de 2010, páginas 142-152.
- [7] Gonzalo Valdés, Marcello Visconti, Hernán Astudillo, “The Tutelkan Reference Process: A Reusable Process Model for Enabling SPI in Small Settings. 18th EuroSPI Conference, Roskilde University, Denmark, junio de 2011, páginas 179-190.
- [8] ISO/IEC 29119 Software Testing, *Four Parts Standard*.
- [9] Jari Andersin, TPI- a model for test Process Improvement, Seminario en Modelos de Calidad para Ingeniería de Software, Departamento de Ciencias de Computación, Universidad de Helsinki, 5 de octubre de 2004.