**Bug Manager**

Sistema de Administración de Calidad de Proyectos de Software

Concepto de Operaciones

Rev 0.4

Fecha: 9 de marzo de 2011

Autor: MR, HG, EC

Distribución:

|  |  |  |  |
| --- | --- | --- | --- |
| **Autoridad** | **Nombre** | **Firma** | **Fecha** |
| Director de Tesis | Óscar Mondragón (OM) |  |  |
| Tesista | Marco Rangel (MR) |  |  |
| Tesista | Eduardo Campos (EC) |  |  |
| Tesista | Humberto García (HG) |  |  |

# Contenidos

|  |  |
| --- | --- |
|  | [Contenidos 2](#_Toc291696609)  [1. Control del Documento 4](#_Toc291696610)  [1.1. Propósito y alcance 4](#_Toc291696611)  [1.2. Audiencia 4](#_Toc291696612)  [1.3. Visión General del Documento 4](#_Toc291696613)  [1.4. Organización del Documento 4](#_Toc291696614)  [1.5. Definiciones 5](#_Toc291696615)  [1.6. Control de Versiones 5](#_Toc291696616)  [1.7. Visión General del Sistema 5](#_Toc291696617)  [2. Sistema Actual 6](#_Toc291696618)  [2.1. Antecedentes, alcance y objetivos del sistema actual 6](#_Toc291696619)  [2.2. Descripción del Sistema Actual 6](#_Toc291696620)  [3. Justificación y naturaleza del sistema propuesto 8](#_Toc291696621)  [3.1. Descripción de los cambios 8](#_Toc291696622)  [3.2. Prioridades respecto a las funcionalidades del sistema propuesto 9](#_Toc291696623)  [3.2.1. Funcionalidades Requeridas 9](#_Toc291696624)  [3.2.2. Funcionalidades Opcionales 28](#_Toc291696625)  [3.3. Supuestos y restricciones 28](#_Toc291696626)  [4. Sistema Propuesto 29](#_Toc291696627)  [4.1. Antecedentes, alcance y objetivos del sistema propuesto 29](#_Toc291696628)  [4.2. Descripción del sistema propuesto 29](#_Toc291696629)  [4.2.1. Módulos del sistema 30](#_Toc291696630)  [4.3. Modos de Operación para el Sistema Propuesto 30](#_Toc291696631)  [4.3.1. Uso individual 31](#_Toc291696632)  [4.3.2. Uso grupal 31](#_Toc291696633)  [4.4. Tipos de Usuario y Actividades 31](#_Toc291696634)  [4.4.1. Desarrollador 31](#_Toc291696635)  [4.4.2. Líder de Proyecto 33](#_Toc291696636)  [4.4.3. Gerente 33](#_Toc291696637)  [4.4.4. Administrador 34](#_Toc291696638)  [4.5. Ambiente para el sistema propuesto 34](#_Toc291696639)  [5. Escenarios de Operación 35](#_Toc291696640)  [5.1. Ingreso al sistema 35](#_Toc291696641)  [5.2. Funcionalidades Administrador 35](#_Toc291696642)  [5.3. Funcionalidades Gerente 36](#_Toc291696643)  [5.4. Funcionalidades Líder de Proyecto 37](#_Toc291696644)  [5.5. Funcionalidades Desarrollador 39](#_Toc291696645)  [6. Impacto 41](#_Toc291696646)  [7. Análisis del sistema propuesto 42](#_Toc291696647)  [7.1. Nuevas Capacidades 42](#_Toc291696648)  [7.2. Desventajas y limitaciones 42](#_Toc291696649) |

# Control del Documento

## Propósito y alcance

El presente documento refiere al concepto de operaciones para el sistema de administración de calidad de proyectos de software, cuyo nombre clave es “Bug Manager” (BM).

## Audiencia

Este documento está dirigido a:

* Clientes: Oscar Mondragón (Director de tesis).
* Equipo de desarrollo: Eduardo Campos, Humberto García y Marco Rangel.
* Usuarios del Bug Manager.

## Visión General del Documento

El propósito de este documento es describir las características y funciones del BM.

Este documento provee un entendimiento en común entre los desarrolladores del software y los grupos de usuarios. Aquí también se expresan las necesidades del usuario, y las expectativas del equipo de desarrollo que proveerá soporte al sistema.

## Organización del Documento

|  |  |
| --- | --- |
| Sección 1 | Contiene información introductoria que incluye el propósito y los alcances del documento, la audiencia a la que está dirigido, definiciones de los términos utilizados en el documento, un control de versiones y una pequeña descripción de cada sección. |
| Sección 2 | Describe los objetivos, antecedentes y políticas del sistema actual. |
| Sección 3 | Describe las áreas de oportunidad del sistema actual y justifica la necesidad de realizar uno nuevo. |
| Sección 4 | Describe las características, metas, restricciones y políticas del BM. |
| Sección 5 | Describe los escenarios operacionales del BM. |
| Sección 6 | Describe el impacto organizacional y operacional del BM. |
| Sección 7 | Resume las ventajas, desventajas y limitaciones del BM. |

## Definiciones

|  |  |
| --- | --- |
| **Término** | **Significado** |
| BM | Bug Manager |

Tabla1: Definiciones

## Control de Versiones

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Versión** | **Fecha** | **Editor** | **Comentarios** | **Aprobación** |
| 0.1 | Marzo 09 | MR, HG | Versión inicial del ConOps. |  |
| 0.2 | Marzo 18 | EC, MR, HG | Se insertan los casos de uso. |  |
| 0.3 | Abril 5 | EC, MR, HG | Se hacen correcciones respecto a la revisión, se agrega la comparación con otros sistemas de software y se proponen métricas y reportes. |  |
| 0.4 | Abril 27 | EC, MR, HG | Se agregaron las estadísticas y reportes que generará el sistema. |  |

Tabla2: Control de Versiones

## Visión General del Sistema

El documento establece las metas, características, ambiente operacional y funcionalidad del BM. El BM será utilizado por cualquier empresa de desarrollo de software, que busque implementar técnicas de control de calidad en su proceso de desarrollo de software. Este producto va enfocado inicialmente a pequeñas y medianas empresas, principalmente mexicanas.

El BM permitirá que las empresas que lo utilicen puedan tener control sobre el avance y la calidad de sus proyectos por medio de lo siguiente:

* Guía en la elaboración del plan de calidad.
* Definición del ciclo de vida y actividades de desarrollo.
* Registro y seguimiento de actividades de aseguramiento de calidad.
* Registro y seguimiento de defectos.
* Generación de estadísticas personales, por proyecto, por equipo y por empresa.

# Sistema Actual

## Antecedentes, alcance y objetivos del sistema actual

Actualmente gran parte de las pequeñas y medianas empresas de desarrollo de software en México no suelen utilizar sistemas de administración de calidad en su proceso de desarrollo. El sistema BM pretende cubrir esta necesidad, mediante la instalación del producto y la adopción del mismo, así como del nuevo proceso implícito en la adopción del sistema.

En la actualidad, la mayoría de las empresas cuentan con sistemas de administración de proyectos o tareas donde se controla principalmente el avance del calendario y en algunos casos el esfuerzo real de las tareas. La mayoría de estos sistemas de administración carece del registro, administración y/o planeación de las actividades de calidad asociadas al desarrollo de software.

Algunas empresas cuentan con herramientas que les ayudan a dar seguimiento a los defectos encontrados. Estos sistemas se usan principalmente en las fases de pruebas o durante el mantenimiento de los sistemas. Los defectos no son analizados por las personas que los cometen y esto dificulta que los programadores de manera consciente eviten los errores que cometen de manera frecuente.

Algunas otras empresas que usan listas de chequeos para revisiones entre colegas, usan sistemas de seguimiento de corrección de defectos. No es frecuente que estás empresas realicen análisis de los errores cometidos, tampoco son asociados los errores a los programadores que los cometen.

Los sistemas arriba mencionados están enfocados al registro y seguimiento de defectos careciendo de un enfoque de prevención y análisis de datos.

## Descripción del Sistema Actual

Como se comentó en la sección pasada, el BM no sustituirá a ninguna aplicación o sistema en específico, siendo su principal objetivo el de optimizar y hacer más eficiente el proceso de desarrollo de software en pequeñas y medianas empresas, principalmente mexicanas, que deseen incorporar actividades de mejora continua en el ciclo de vida de desarrollo.

Sin embargo, existen herramientas y programas de software que realizan tareas similares a las que realizará el sistema propuesto. Principalmente estos sistemas se dedican al registro y rastreo de defectos, así como al registro de las actividades realizadas dentro del ciclo de desarrollo, como una especie de bitácora. En general, consideramos que estos sistemas atacan una parte del problema y comúnmente carecen de la funcionalidad necesaria para agregar verdadero valor al proceso y método de desarrollo.

Hay más de 50 sistemas de registro y seguimiento de defectos, con diferentes tipos de licenciamiento y diferentes funcionalidades. A continuación presentamos una descripción de los sistemas que consideramos más significativos y similares en cuanto a funcionalidad respecto al sistema propuesto.

* *Bugzilla*: Es una herramienta de propósito general basada en tecnología web, pensada como un sistema de rastreo de defectos. Fue desarrollada para el proyecto Mozilla. Aunque su diseño soportaría ser una herramienta de administración de proyectos y actividades, el equipo de desarrollo ha decidido mantenerlo como un sistema de rastreo de defectos. Es prácticamente un sistema de tickets, en el que cualquiera puede levantar un nuevo ticket, que corresponde a un defecto, y este ticket se asigna a un desarrollador. Este sistema se utiliza tanto para proyectos propietarios como de open source.
* *PSP*: La herramienta de PSP permite el registro de las actividades realizadas para el desarrollo del producto, así como el registro de defectos (por tipo) inyectados y removidos en cada actividad. Además permite la estimación del proyecto de desarrollo. En base a estos registros, calcula una serie de métricas referentes a los defectos, al plan de estimación, al proceso y a la calidad. Todas estas métricas proporcionan información valiosa que posteriormente permite tomar decisiones para mejorar continuamente el proceso de desarrollo.
* *BugTracker .NET*: Esta herramienta cuenta con funcionalidades interesantes, como la de capturar una pantalla y registrarla como un defecto, poniendo las características del mismo como comentarios. También tiene la capacidad de registrar los correos electrónicos recibidos como defectos. Una parte importante es la facilidad para filtrar y ordenar la lista de defectos y sus columnas conforme a las necesidades del usuario. Además se pueden definir los estatus de defectos y el proceso a seguir para el seguimiento de los mismos.
* *BugNET*: Esta herramienta contiene un módulo para administración del proyecto, así como un módulo de administración de defectos. Dentro de las funcionalidades interesantes podemos mencionar el soporte para múltiples proyectos, el cálculo y reporte de métricas, la capacidad de crear un resumen con los defectos por proyecto, así como la creación de una bitácora con el histórico de cada defecto. Además de la posibilidad de agregar archivos a cada defecto, realizar comentarios por defecto, notificación de cambios en los estatus y la capacidad de configurar roles y permisos para cada usuario.
* *Bontq*: Es una de las herramientas más recientes para la administración de proyectos y seguimiento de defectos. Dentro de sus funcionalidades más destacables se encuentran la integración con Google Docs, la posibilidad de grabar videos (para mejor demostración de un defecto), capturar pantallas y agregar archivos a cada defecto. Además provee la capacidad de manejar los proyectos por compañías, es decir, el nivel más alto de administración es la empresa y no el proyecto, por lo que se puede contar con la administración de varias empresas con múltiples proyectos cada una.

También permite la posibilidad de registrar defectos, solicitudes de nuevas funcionalidades, avisos, actividades y casos de prueba. Además permite la configuración de la clasificación de defectos por su impacto, así como el envío de notificaciones vía correo electrónico.

Con esto nos podemos dar una clara idea del estado en el que se encuentran las herramientas para administración y seguimiento tanto de proyectos como defectos. Algunas herramientas cuentan con ciertas funcionalidades y otras se concentran en otro tipo de funcionalidades. Todo esto nos da un panorama de lo que ya está hecho y de cómo lo podemos combinar para lograr un sistema completo y eficiente.

# Justificación y naturaleza del sistema propuesto

El principal motivo para el desarrollo de este nuevo sistema se centra en la necesidad, cada vez mayor, de los adquisidores o compradores de productos de software, de contar con sistemas que realmente les representen una mejora significativa en sus procesos actuales; con sistemas de la más alta calidad. Esta necesidad se puede ver reflejada en la vida diaria empresarial, cuando observamos que las compañías requieren incrementar la calidad y decrementar el tiempo de servicio que proporcionan a sus clientes, en la necesidad de contar con la mayor y mejor información posible en el menor tiempo posible.

Todo esto lleva a que las empresas dedicadas al desarrollo de software estén envueltas en un proceso de mejora continua respecto a los productos que lanzan al mercado, es decir, productos que cada vez sean de mayor calidad y por tanto contengan menos defectos. La naturaleza del sistema propuesto (BM), consiste básicamente en provocar un cambio cultural evolutivo en la manera de desarrollar software:

* Promover una cultura de calidad personal en el programador contra una cultura de calidad asignada a grupos organizacionales ajenos al desarrollo (pruebas, adherencia a procesos).
* Inculcar una meta en el programador/grupo de desarrollo de cero defectos en pruebas de unidad contra número de componentes programados por hora.
* Promover la prevención de defectos en lugar de la búsqueda de defectos durante las pruebas.
* Enfocar el esfuerzo e técnicas de remoción de defectos al inicio del ciclo de vida en lugar de crecer los grupos dedicados a las pruebas al final del ciclo de vida.
* En resumen, promover un compromiso personal a la calidad del desarrollo de software y a las actividades asociadas para su mejora continua.

Es aquí donde la naturaleza del sistema propuesto se mezcla con la necesidad y justificación para el desarrollo del mismo.

## Descripción de los cambios

Con respecto a sistemas tradicionales de administración de proyectos o sistemas enfocados a administrar la resolución de defectos registrados, evidentemente el cambio más importante se dará en la manera diaria de trabajar de los usuarios del sistema, los programadores y los grupos de programadores.

Actualmente las herramientas de administración de desarrollo, cuentan con un módulo para el registro y seguimiento de errores, sin embargo, en este seguimiento no se hace una clasificación adecuada ni se generan métricas de valor real.

El BM pretende dar un valor agregado y generar métricas útiles en el proceso de remoción y administración de defectos con las siguientes características:

* Al ser una aplicación web se pretende que esta tenga una naturaleza multiusuario.
* La herramienta generará estadísticas por desarrollador, equipo de trabajo, conjunto de proyectos y por empresa, lo que le da un gran poder de análisis a los gerentes o personas encargadas el proceso y calidad en el desarrollo de software de las organizaciones.
* A través de definición y personalización de plantillas de listas de chequeo, se pretende dar al desarrollador la máxima flexibilidad para maximizar la efectividad de las técnicas de remoción.
* Aparte de registrar y dar seguimiento a los defectos, se hará una clasificación ortogonal de estos, destacando en qué fase fue inyectado el defecto, por qué miembro del equipo de desarrollo fue inyectado, en qué fase fue detectado, con qué técnica, en qué fase fue removido y cuál fue el esfuerzo de removerlo.

## Prioridades respecto a las funcionalidades del sistema propuesto

A continuación se presentan las Funcionalidades Requeridas, que son aquellas que tendrá el sistema en su primera versión, y las Funcionalidades Opcionales las cuales se determinará si se incluirán en la primera versión al hacer el diseño conceptual y el dimensionamiento del sistema.

### Funcionalidades Requeridas

Las funcionalidades indispensables incluyen:

Usuario administrador

* Alta, baja y cambio de usuarios. Los datos requeridos para la administración de usuarios son:
  + Nombre completo
  + Nombre de usuario
  + Contraseña
  + Tipo de usuario
  + Correo electrónico
* Alta, baja y cambio de proyectos. Los datos requeridos para la administración de proyectos son:
  + Nombre
  + Descripción
  + Fecha de inicio
  + Fecha estimada de fin
  + Fecha comprometida de fin.

Usuario Gerente

* Asignación de recursos humanos a los proyectos existentes. Se asignará a un proyecto en específico:
  + Líder de proyecto
  + Equipo de desarrollo
* Visualización de cualquier estadística y reporte que genere el sistema.

Usuario Líder de proyecto

* Creación del plan de calidad. Consiste en la definición de actividades de calidad, técnicas de remoción y la definición de objetivos para llevar a cabo el producto de software con la calidad requerida.
* Creación del ciclo de vida de desarrollo para un proyecto en específico. La creación del ciclo de vida consiste en la definición de las fases de desarrollo.
* Alta, baja y cambio de actividades de desarrollo para el proyecto en específico.
* Alta, baja y cambio de actividades de calidad para el proyecto en específico.
* Asignación de actividades de desarrollo y calidad al equipo de desarrollo.
* Alta, baja y cambio de plantillas para cada actividad de aseguramiento de la calidad, por ejemplo listas de chequeo personal y entre colegas por tipo de producto de trabajo y fase. Estas plantillas estarán disponibles para uso de cualquier miembro del equipo de desarrollo.
* Visualización de estadísticas y reportes por desarrollador, por proyecto, por proyecto y desarrollador, por proyectos en los que ha participado.

Usuario Desarrollador

* Alta, baja y cambio de actividades de desarrollo para el proyecto en específico.
* Actualización de actividades de desarrollo. Consiste en el cambio de estatus de inicio de la actividad, actualización del esfuerzo requerido, así como el cambio de estatus de fin de la actividad.
* Actualización de actividades de calidad. Consiste en el cambio de estatus de inicio de la actividad, actualización del esfuerzo requerido, registro y seguimiento de defectos, así como el cambio de estatus de fin de la actividad.
* Registro y seguimiento de defectos. Consiste en el registro de los parámetros iniciales del defecto, en la corrección del mismo y en el registro de sus parámetros finales.
* Alta, baja y cambio de plantillas para cada actividad personal de aseguramiento de calidad (listas de chequeo de revisiones personales). Estas plantillas estarán disponibles únicamente para el desarrollador que las creó.
* Visualización de estadísticas y reportes sobre un proyecto en específico y un desarrollador en específico, así como la visualización de reportes sobre un desarrollador en específico a lo largo de los proyectos en los que ha participado.

\* La funcionalidad dentro del sistema se hereda para los diferentes tipos de usuario. El gerente puede realizar las funciones del líder de proyecto y el líder de proyecto puede realizar las funciones del desarrollador.

Reportes y estadísticas

* Generación de estadísticas, métricas y reportes a medida que los desarrolladores y líderes de proyecto registran el esfuerzo realizado (sobre actividades de desarrollo y calidad), registran y corrigen los defectos encontrados.
  + Estadísticas y reportes generales

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **REQUERIDA/DESEABLE** | **METRICAS** | **FORMULAS** | **ALCANCE** |
| Tiempo de fase | Requerida | - Tiempo fase  - Tiempo proyecto | % = Tiempo fase/Tiempo proyecto | Por proyecto |
| **Bosquejo**    En la gráfica se muestra el porcentaje de tiempo invertido en cada fase del proyecto. La suma de todas las fases debe ser 100. | | | | |
|  | **REQUERIDA/DESEABLE** | **METRICAS** | **FORMULAS** | **ALCANCE** |
| Productividad por fase | Requerida | * Total esfuerzo fase * Tamaño del product | Productividad = Tamaño del producto / Total esfuerzo fase | Por proyecto |
| **Bosquejo**    La gráfica representa el número de productos que se realizan por hora. Para la etapa de diseño puede ser documentos, para la de pruebas casos de prueba y para la de codificación líneas de código. | | | | |
|  | **REQUERIDA/DESEABLE** | **METRICAS** | **FORMULAS** | **ALCANCE** |
| Yield de fase | Requerida | * Total defectos * Total defectos encontrados fase | Yield = 100\* Total defectos encontrados fase / Total defectos | Por proyecto |
| **Bosquejo**    La gráfica muestra el porcentaje de defectos encontrados en cada fase del proyecto respecto al total de defectos encontrados en el proyecto. | | | | |

* + Estadísticas y reportes sobre costo de la calidad

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **REQUERIDA/DESEABLE** | **METRICAS** | **FORMULAS** | **ALCANCE** |
| Productividad y productividad compuesta | Requerida | - Total LOC  - Tiempo desarrollo  - Tiempo corrección | P = Total LOC/Tiempo desarrollo  PC = Total LOC/(Tiempo desarrollo + Tiempo corrección) | Por persona, por equipo, por empresa, por proyecto. |
| **Bosquejo**    Este reporte nos ayudará a comparar la productividad de un programador contra su productividad compuesta. La productividad compuesta está ponderada con el tiempo que se tomó en corregir los defectos que inyectó al momento de programar. | | | | |
|  | **REQUERIDA/DESEABLE** | **METRICAS** | **FORMULAS** | **ALCANCE** |
| ROI de proyecto con y sin actividades de calidad | Requerida | - Esfuerzo en técnicas de calidad  - Inversión en herramientas para calidad  - Bugs, fase de inyección y fase de remoción.  - Ponderación de esfuerzo y costo de los errores. | ROI = (Total CoQ Ahorrado – Inversión en calidad) / Inversión en Calidad | Por proyecto, por empresa. |
| **Bosquejo**  Este reporte es un análisis del retorno de inversión de las actividades de calidad del proyecto. Te ayuda a saber que tantos recursos se ahorraron gracias a la implementación de estas técnicas de calidad. Finalmente te ayuda a hacer una comparativa contra el caso en que no se hubieran utilizado las actividades de calidad. | | | | |
|  | **REQUERIDA/DESEABLE** | **METRICAS** | **FORMULAS** | **ALCANCE** |
| ROI de las estrategias de calidad | Requerida | - Esfuerzo en técnicas de calidad  - Inversión en herramientas para calidad  - Bugs, fase de inyección y fase de remoción.  - Ponderación de esfuerzo y costo de los errores. | ROI = (Total CoQ Ahorrado – Inversión en calidad) / Inversión en Calidad | Por persona, por equipo, por empresa, por proyecto. |
| **Bosquejo**    Este reporte nos ayuda a comparar el esfuerzo que tomó cierta actividad de calidad, contra el tiempo que ahorró en caso de que los errores encontrados con la respectiva técnica hubieran pasado a producción. | | | | |
|  | **REQUERIDA/DESEABLE** | **METRICAS** | **FORMULAS** | **ALCANCE** |
| COQ vs CNQ | Requerida | - Esfuerzo en técnicas de calidad  - Inversión en herramientas para calidad  - Bugs, fase de inyección y fase de remoción.  - Ponderación de esfuerzo y costo de los errores. | - Ponderar esfuerzo en técnicas de calidad.  - Ponderar el costo de los defectos. | Por persona, por equipo, por empresa, por proyecto. |
| **Bosquejo**    Esta gráfica nos ayuda a comparar los costos de la conformidad y no conformidad. | | | | |

* + Estadísticas y reportes sobre técnicas de remoción

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **REQUERIDA/DESEABLE** | **METRICAS** | **FORMULAS** | **ALCANCE** |
| Yield de técnica | Requerida | * Total defectos removidos por cada técnica * Total defectos * % defectos removidos por cada técnica * Total esfuerzo por cada técnica * Total esfuerzo * % esfuerzo por cada técnica * Eficiencia de remoción * Razón de revisión | * % defectos = Total defectos técnica / Total defectos * % esfuerzo = Total esfuerzo técnica/ Total esfuerzo * Eficiencia de remoción = Total defectos técnica / Total esfuerzo técnica * Razòn de revisión = LOC / Total esfuerzo técnica | Por persona, por equipo, por proyecto y por empresa. |
| **Bosquejo**   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **PROYECTO 1** | | | | | | | |  | Total defectos removidos | % defectos removidos | Total esfuerzo | % esfuerzo | Eficiencia de remoción | Razón de revisión | | Revisión Personal | 50 | 50% | 100 | 40% | .5 | 200 | | Caminata | 25 | 25% | 80 | 32% | .3125 | 150 | | Inspección | 25 | 25% | 70 | 28% | .35 | 150 |   Matriz que contiene el yield de cada técnica de remoción para un proyecto en específico o de la empresa en general. Se compara el porcentaje de defectos que se encuentran en cada técnica contra el total de esfuerzo invertido, además de mostrar la eficiencia de la técnica, así como la velocidad de remoción de cada una. | | | | |
|  | **REQUERIDA/DESEABLE** | **METRICAS** | **FORMULAS** | **ALCANCE** |
| Eficiencia relativa | Requerida | * Eficiencia de remoción de cada técnica. | * Eficiencia relativa = Eficiencia técnica 1 / Eficiencia técnica 2 | Por persona, por equipo, por proyecto y por empresa. |
| **Bosquejo**   |  |  | | --- | --- | | **PROYECTO 1** | | | Revisión Personal vs Caminata | 1.6 | | Revisión Personal vs Inspección | 1.42 | | Caminata vs Inspección | .90 |   La tabla muestra la eficiencia relativa de cada técnica respecto a todas las demás. Es decir, se muestra la razón de la eficiencia de dos técnicas entre sí y se muestran todas las combinaciones posibles entre todas las técnicas. | | | | |
|  | **REQUERIDA/DESEABLE** | **METRICAS** | **FORMULAS** |  |
| Pareto por tipo de defecto. | Requerida | * Total de defectos por técnica y tipo. | Sumatoria de los defectos removidos de cada técnica por tipo de defecto. | Por persona, por proyecto, por equipo y por empresa. |
| **Bosquejo**  En cada gráfica se muestra la cantidad de defectos removidos por cada técnica (de mayor a menor) para un tipo de defecto en específico. En el ejemplo se muestra gráficas para defectos de sintaxis, de interfaz y de función. | | | | |
|  | **REQUERIDA/DESEABLE** | **METRICAS** | **FORMULAS** | **ALCANCE** |
| Distribución normal de la efectividad de cada técnica de remoción | Requerida | * Efectividad de cada técnica * Tamaño del producto. | Efectividad normalizada = 1000 \* Efectividad técnica / Tamaño del producto. | Por proyecto. |
| **Bosquejo**  La gráfica consiste en mostrar la distribución normal de la efectividad de una técnica de remoción en específico. La efectividad estará normalizada al tamaño del producto de software, para que se pueda realizar una comparación consistente entre diferentes proyectos. | | | | |

* + Estadísticas y reportes sobre caracterización de defectos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **REQUERIDA/DESEABLE** | **METRICAS** | **FORMULAS** | **ALCANCE** |
| Densidad de defectos por usuario | Requerida | - Total LOC  - Total de defectos  - Medida estándar (1000 líneas de código, 40 puntos de función, etc.) | D = (Total de defectos / Total LOC) \* Medida estándar | Por persona. |
| **Bosquejo** | | | | |
|  | **REQUERIDA/DESEABLE** | **METRICAS** | **FORMULAS** |  |
| Defectos por tipo | Requerida | - Total de defectos | DT = Total de defectos organizados por tipo | Por usuario, por proyecto, por empresa. |
| **Bosquejo** | | | | |
|  | **REQUERIDA/DESEABLE** | **METRICAS** | **FORMULAS** | **ALCANCE** |
| Fase de inyección y remoción | Requerida | * Total de defectos | DF = Total de defectos organizados por fase de inyección/remoción. | Por persona, por equipo, por empresa, por proyecto. |
| **Bosquejo** | | | | |

### Funcionalidades Opcionales

Las funcionalidades consideradas como opcionales incluyen:

* Generación de modelos de predicción de defectos sobre los desarrollos actuales, basado en datos históricos y en la opinión de los integrantes del proyecto.
* Generación de modelos de predicción sobre el desempeño de los desarrolladores en base a información histórica de su desempeño en proyectos anteriores.
* Generación de modelos de predicción sobre el tiempo y esfuerzo requerido para la elaboración del sistema.
* Recomendación en la conformación del equipo de desarrollo sobre un proyecto en específico en base a las habilidades de cada posible miembro.
* Introducción de la información generada sobre calidad en la etapa de pruebas para actualización y comparación del proceso de calidad previo a la fase de pruebas.

## Supuestos y restricciones

Supuestos:

* Los usuarios del sistema cuentan o contarán con la habilidad de uso básico de Internet y computadoras necesarias para utilizarlo.
* El cliente cuenta con red local e Internet necesario para el manejo e instalación del sistema.
* El cliente cuenta con un servidor o un servicio de hosting en Internet para el alojamiento de la aplicación.
* Dicho servidor o servicio de hosting funcionará bajo el Sistema Operativo Windows o Linux.
* La empresa de desarrollo cuenta con el rol de gerentes de proyectos.
* La empresa de desarrollo cuenta con el rol de líder de proyecto.
* La empresa de desarrollo cuenta con el personal adecuado para realizar el rol de administrador del sistema.
* La empresa de desarrollo cuenta con el rol de desarrollador.

Restricciones:

* El sistema trabaja sobre un ambiente Web, por lo que es indispensable que el servidor y el usuario que acceda a la aplicación cuenten con Internet o en su defecto acceso a la red local en caso de ser utilizado dentro de la red de la empresa.
* La visualización del sistema desde un navegador para teléfonos inteligentes o aparatos móviles de comunicación similares no será la misma que aquella de un navegador convencional, por lo que afectará la manera de utilizar el sistema.

# Sistema Propuesto

## Antecedentes, alcance y objetivos del sistema propuesto

El sistema propuesto no cuenta con antecedentes de desarrollo o versiones anteriores del mismo, por lo tanto será la primera versión que se liberará al mercado. Los antecedentes se refieren más bien a la evolución histórica de las técnicas y actividades de aseguramiento de calidad creadas e implementadas para el desarrollo de software. El sistema pretende implementar y adaptar actividades tanto de índole personal como grupal para incrementar al máximo posible la calidad del producto final.

Se puede definir como alcance en cuanto a funcionalidad lo siguiente:

* Registrar y dar seguimiento a las actividades de desarrollo y calidad establecidas para el ciclo de vida de desarrollo.
* Hacer un seguimiento puntual a la inyección, remoción y corrección de defectos a lo largo de las diferentes etapas del ciclo de vida.
* Generar estadísticas y métricas de valor para la empresa y el personal en base a la información proporcionada por los usuarios del sistema.
* Dar una guía en los procedimientos principales de aseguramiento de la calidad.

Los objetivos principales del sistema son:

* Lograr una mejora continua en el proceso de desarrollo de software, así como una mayor calidad en el producto final.
* Reducir el costo de implementar actividades de calidad dentro de la empresa.
* Proporcionar información valiosa a la empresa para la toma de decisiones respecto a cambios en sus procesos de desarrollo de software.
* Facilitar la evolución y adaptación de las diferentes actividades de aseguramiento de la calidad.
* Promover una cultura de calidad personal enfocada en la prevención de defectos.
* Proveer datos sobre el esfuerzo (costo) de las actividades de remoción de defectos.
* Facilitar un cambio cultural respecto a la forma de trabajar de pequeñas y medianas empresas.

## Descripción del sistema propuesto

El sistema BM se puede definir como un sistema de administración, seguimiento y de mejora continua de la calidad personal y de grupo en el desarrollo de sistemas de software; todo esto mediante el seguimiento de proyectos, actividades de desarrollo y de calidad a lo largo del ciclo de vida, de defectos inyectados, removidos y corregidos, y de la creación de estadísticas y métricas verdaderamente útiles a la empresa.

El sistema BM contará con la flexibilidad necesaria para que la persona responsable dentro de la empresa defina el ciclo de vida y las actividades a realizar. Así mismo podrá definir la taxonomía a utilizar respecto a los diferentes defectos dentro del sistema, así como las plantillas a utilizar para las actividades de calidad. Esta flexibilidad permitirá que la aplicación pueda ser adoptada por un gran número de empresas con diferentes métodos y ciclos de vida de desarrollo. De esta manera se logrará una mayor penetración en el mercado.

El sistema estará basado en la tecnología web, por lo que no será necesaria la instalación física de la aplicación en los clientes que pretendan acceder a ella. Sólo será necesario el uso del navegador para acceder a este sistema.

### Módulos del sistema

* Módulo de Administración (Altas, Bajas y Cambios de proyectos, actividades y usuarios)
* Módulo de Actualización de Estatus (Actualización de estatus de actividades)
* Módulo de Calidad (Plantillas y actividades de remoción de defectos)
* Módulo de Defectos (Registro y seguimiento de defectos)
* Módulo de Reportes (Generación de reportes estadísticos personales, por proyecto, por equipo y de empresa)

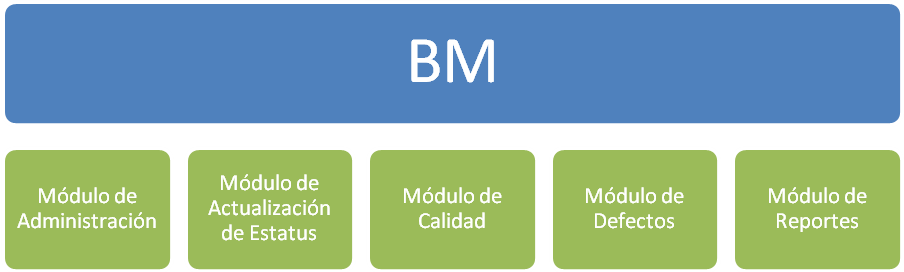


Ilustración 1. Módulos principales del sistema.

## Modos de Operación para el Sistema Propuesto

El BM será un sistema web el cual permitirá a los usuarios conectarse por medio de un navegador web a una aplicación montada en un servidor Apache Tomcat. Todos los usuarios contarán con un nombre de usuario y una contraseña las cuales le brindarán acceso al sistema. Dependiendo del tipo de usuario serán las distintas actividades que el usuario pueda realizar.

El sistema tiene como condición importante elegir el tipo de tamaño que se utilizará para medir los productos de software. Los tipos de tamaños más comunes son: Líneas de código (LOC), Puntos de Función, Número de Clases, párrafos y páginas entre otros. Este tipo de tamaño determinará la forma en que se generan las estadísticas y los reportes. Es importante mencionar que si el usuario desea cambiar el tipo de tamaño, debe de generar una instalación nueva del sistema, ya que los tipos de tamaño no son compatibles.

### Uso individual

Es la forma más común de utilizar el sistema. Los usuarios, ya sean desarrolladores, líderes de proyecto, gerentes o administradores, accederán al sistema vía un navegador web. Esto les permitirá desempeñar las actividades relacionadas con su tipo de usuario.

### Uso grupal

La aplicación soportará la conexión paralela de varios usuarios vía el servidor web en el cual residirá la aplicación principal.

El uso grupal está relacionado más con la forma en que se generan las estadísticas, que un propio trabajo en equipo. No existe un modo en el que dos o más usuarios utilicen el sistema en el mismo navegador. Los usuarios accederán paralelamente a sus distintas actividades por medio de los navegadores web de sus computadoras personales. Así pues, el trabajo en paralelo genera estadísticas grupales, las cuales pueden ser revisadas por proyecto, por conjunto de usuarios y por empresa.

## Tipos de Usuario y Actividades

A continuación se describe el uso de los distintos tipos de usuario: Desarrollador, Líder de Proyecto, Gerente y Administrador.

### Desarrollador

Es el usuario más básico del sistema. Sus principales actividades estarán relacionadas con el registro de datos acerca de: esfuerzo que le tomó realizar una tarea, tiempo que le tomó hacer una tarea, registro de defectos encontrados en el análisis y seguimiento de defectos encontrados.

#### Actividades de Desarrollo

El desarrollador podrá tener asignadas actividades directamente relacionadas con la elaboración del producto de software. Estas pueden ser: Documentos de desarrollo, productos de diseño, código fuente y cualquier otro artefacto determinada en el ciclo de vida del sistema. En estas actividades el desarrollador tendrá que registrar el tiempo, el esfuerzo que le tomó realizar la tarea y el tamaño de la tarea.

Las tareas pueden ser medidas de distintas formas dependiendo de su naturaleza. La medida recomendada para las actividades de requerimientos y diseño es número de páginas, mientras que para el código fuente se recomienda líneas de código.

Una vez que se define la medida que se va a utilizar para los distintos tipos de productos, esta no puede cambiar, debido a que no sería compatible con lo que se ha medido antes y las métricas generadas serían incorrectas.

#### Actividades de Calidad

Cada actividad de desarrollo tendrá ligada una actividad de calidad determinada por el líder de proyecto. Esta actividad de calidad deberá realizarse una vez terminada la actividad de desarrollo. Para la actividad de calidad el desarrollador tiene que registrar:

* Fecha de inicio y fecha de fin de la actividad.
* Esfuerzo requerido.
* Registro de defectos encontrados.

Durante cada actividad de calidad, todos los defectos encontrados quedarán registrados en el sistema. A cada defecto se le tendrá que dar un seguimiento específico el cual consta de los siguientes pasos:

* Registro ortogonal del defecto:
  + Apertura del defecto:
    - Actividad: Se refiere a la actividad que se estaba realizando cuando se encontró el defecto. Un ejemplo de estas actividades sería inspecciones de código, pruebas unitarias, pruebas de sistema, etc.
    - Catalizador: Se refiere al ambiente o condición existente que ocasiona que el defecto aparezca, aquella condición necesaria para reproducir el defecto. El método ya tiene algunos activadores definidos para cierto tipo de actividades, aunque también se pueden usar los propios.
    - Impacto: Aquí se selecciona el impacto que el defecto podría haber tenido en el cliente final si no se arreglara. En caso de que el defecto lo haya detectado un cliente, se pone el impacto actual que el defecto tuvo en el cliente. Determinar en base a su prioridad e impacto si el defecto se corrige o no.
  + Cierre del defecto:
    - Target: Representa la entidad de alto nivel que se arregló. Por ejemplo, el diseño o el código del programa.
    - Tipo de defecto: Representa la naturaleza de la corrección que se realizó. Para este campo existen valores predeterminados, por ejemplo: asignación o inicialización, que se refiere a que el valor de una variable fue corregido; algoritmo/método, significa que se implementó o corrigió un algoritmo; tiempo/serialización, que nos indica que se tuvo que implementar métodos de serialización en un recurso compartido.
    - Calificador: Aplica al tipo de defecto, y captura el calificativo que describe al defecto. Para este atributo existen 3 valores posibles:
      * Omisión: Indica que el defecto se debió a una omisión.
      * Incorrecto: Indica que valores incorrectos se usaron.
      * Extraño: Indica que el defecto se debe a algo no relevante al código.
    - Fuente: Identifica el origen del target que tenía el defecto. Para este atributo también existen ciertos valores predeterminados:
      * Desarrollado internamente: Nos dice que el defecto fue introducido por el equipo de desarrollo
      * Reusado de una librería: Nos indica que el defecto pertenecía a una librería de re uso.
      * Outsourced: Nos indica que el defecto fue introducido por un prestador de servicios.
      * Portado: Nos indica que el defecto tiene que ver con una parte que fue validada en un ambiente diferente.
    - Edad: Identifica el historial del target que tenía el defecto. Para este campo existen los siguientes valores:
      * Base: Nos indica que el defecto está en un aparte del producto que no ha sido modificada en el proyecto actual, y no es parte de una librería de re uso. Es un defecto latente.
      * Nuevo: El defecto fue introducido en el proyecto actual.
      * Re-escrito: El defecto fue introducido por re-diseñar o re-escribir una función con el objetivo de mejorar su diseño o calidad.
      * Re-arreglado: El defecto fue introducido al proveer una solución a un defecto previo.
* Corrección del defecto en caso que se haya decidido.
* Revisión de la corrección.
* Registro final del defecto.

El desarrollador contará con una plantilla para cada actividad de calidad, la cual servirá como guía y formato de registro.

#### Revisión de Métricas

Finalmente el desarrollador será capaz de revisar sus estadísticas de desempeño personal, ya sea por proyecto o por el conjunto de los proyectos en los que ha estado involucrado. Entre los datos que podrá revisar estarán:

• Productividad, que se mide en unidad de tamaño sobre tiempo.

• Tasa de inyección de defectos, que es el número de defectos sobre unidad de tamaño.

• Porcentaje de re trabajo, que es la cantidad de trabajo que se tuvo que realizar a causa de los defectos inyectados.

• Productividad real, que es la productividad ponderada con el porcentaje de re trabajo.

### Líder de Proyecto

El líder de proyecto tendrá como actividades principales:

• Determinar el ciclo de vida del proyecto.

• Realizar el plan de proyecto.

• Realizar el plan de calidad de proyecto.

• Asignar actividades de trabajo y calidad a los desarrolladores.

• Generar reportes estadísticos del equipo.

• Generar reportes estadísticos de los proyectos a su cargo.

#### Planeación de Proyecto

El líder de proyecto inicia con una serie de proyectos y recursos asignados por el gerente. Al inicio de cada proyecto el líder de proyecto debe de determinar su ciclo de vida. Esto significa decidir las fases que comprenderán el desarrollo, por ejemplo: Requerimientos, diseño, codificación y pruebas. El líder de proyecto tendrá la libertad de definir el modelo de ciclo de vida que más le agrade.

Una vez decidido el ciclo de vida se pasa a la elaboración del plan de proyecto y plan de calidad. En el plan de proyecto se determinaran todas las actividades a realizarse, así como sus fechas de inicio y fin, y su responsable. El líder de proyecto será capaz de hacer proyecciones de cómo funcionaría el proyecto con distintos usuarios. En el plan de calidad se determina que actividad de calidad realizar por cada actividad de desarrollo, y se fijan los objetivos de calidad. También se determina las plantillas a utilizar para cada actividad de calidad.

#### Revisión de métricas

El líder de proyecto podrá revisar las métricas de desempeño de algún desarrollador en particular, de un proyecto en específico o un conjunto de proyectos. Las métricas principales serán:

• Productividad, que se mide en unidad de tamaño sobre tiempo.

• Tasa de inyección de defectos, que es el número de defectos sobre unidad de tamaño.

• Porcentaje de re trabajo, que es la cantidad de trabajo que se tuvo que realizar a causa de los defectos inyectados.

• Productividad real, que es la productividad ponderada con el porcentaje de re trabajo.

• Entre muchas otras más.

### Gerente

El gerente es el encargado de crear nuevos proyectos, de asignar líderes de proyecto a los distintos proyectos y de asignar recursos humanos a los líderes de proyecto. También podrá tener acceso a los reportes estadísticos por desarrollador, líder de proyecto, proyecto y conjunto de proyectos.

#### Actividades de Planeación

El gerente es el que inicia con el funcionamiento de sistema. Primero debe de determinar la unidad de medida para los proyectos, por ejemplo líneas de código, puntos de función, clases, etc. Después tendrá que crear un proyecto. Los proyectos serán asignados a los distintos líderes de proyecto. También asignará los recursos a los distintos líderes de proyecto.

#### Revisión de Métricas

El gerente podrá revisar las métricas de desempeño de algún desarrollador en particular, de un proyecto en específico, de un líder de proyecto o un conjunto de proyectos. Las métricas principales serán:

• Productividad, +que se mide en unidad de tamaño sobre tiempo.

• Tasa de inyección de defectos, que es el número de defectos sobre unidad de tamaño.

• Porcentaje de re trabajo, que es la cantidad de trabajo que se tuvo que realizar a causa de los defectos inyectados.

• Productividad real, que es la productividad ponderada con el porcentaje de re trabajo.

• Entre muchas otras más.

### Administrador

Es la persona encargada de configurar y dar mantenimiento al sistema. Es capaz de dar de alta y baja usuarios y proyectos, así como dar permisos a los diferentes usuarios.

## Ambiente para el sistema propuesto

El sistema será desarrollado de manera tal que se pueda acceder a él mediante un navegador web. Esto nos dará la ventaja que, sin importar la plataforma que se vaya a acceder, se podrá ver y editar los datos sin dificultad.

# Escenarios de Operación

En esta sección se explican las diferentes funciones a las cuales se tendrá acceso.

## Ingreso al sistema

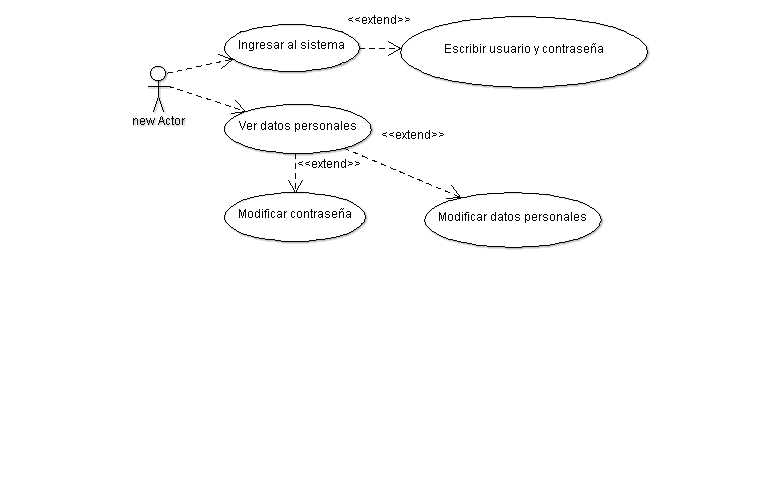


Ilustración 2. Ingreso al sistema y modificación de datos personales

## Funcionalidades Administrador



Ilustración 3. Alta, baja y modificación de usuarios



Ilustración 4. Alta, baja y cambio en proyectos

## Funcionalidades Gerente



Ilustración 5. Asignación de proyectos y de recursos.



Ilustración 6. Generación de estadísticas.

## Funcionalidades Líder de Proyecto



Ilustración 7. Plan de calidad y ciclo de vida.



Ilustración 8. Asignación de actividades de desarrollo y de calidad

.



Ilustración 9. Generación de estadísticas.



Ilustración 10. Alta, baja y cambio de plantillas públicas.

## Funcionalidades Desarrollador



Ilustración 11. Actualización de actividades de desarrollo.



Ilustración 12. Actualización de actividades de calidad.



Ilustración 13. Alta, baja y cambio de plantillas privadas.



Ilustración 14. Generación de estadísticas



Ilustración 15. Registro y seguimiento de defectos

# Impacto

El impacto que se pretende lograr involucra a los diferentes niveles de cualquier organización que se dedique al desarrollo de software.

* A nivel de alta gerencia provocará una mayor formalidad en la manera de controlar y asignar recursos a nuevos y existentes proyectos.
* A nivel de líder de proyecto provocará una mayor formalidad y disciplina para la realización de actividades de planeación referentes al ciclo de vida y a la calidad del producto.
* El mayor impacto se dará en la manera en que los desarrolladores involucrados en el proyecto trabajan día con día. Implicará que los desarrolladores cuenten con la suficiente disciplina para realizar las actividades de calidad de la mejor manera posible, para alimentar el sistema con información verídica que posteriormente permita mostrar estadísticas y métricas reales, que logren que la empresa y los desarrolladores estén conscientes de lo que realmente está pasando en el proceso de desarrollo.

# Análisis del sistema propuesto

## Nuevas Capacidades

Dentro de las capacidades del sistema BM se encuentran:

* Distintos modos de operación del sistema de acuerdo al tipo de usuario que se firme en el sistema.
* Seguridad, en términos de restricciones y privilegios, para cada tipo de usuario.
* Acceso simultáneo por parte de múltiples usuarios.
* Se evita la instalación de cualquier tipo de API o DLL en los clientes que accederán al sistema.
* Guía en el proceso de definición del plan de calidad y del ciclo de vida de desarrollo del producto.
* Generación de actividades específicas y puntuales de desarrollo del sistema.
* Generación de actividades de aseguramiento de calidad relacionadas específicamente con ciertas actividades de desarrollo.
* Registro de defectos en cualquier fase del proceso, así como seguimiento sobre la corrección y/o remoción de defectos en cualquier fase.
* Flexibilidad en la definición de los tipos de defectos a identificar en el proyecto.
* Flexibilidad en la definición de plantillas y listas para las actividades de aseguramiento de la calidad en las distintas fases del ciclo de vida.
* Generación de estadísticas y métricas de desempeño personales, de equipo, por proyecto y por empresa.

## Desventajas y limitaciones

Dentro de las desventajas y/o limitaciones identificadas para el sistema BM se encuentran:

* Si bien se permite el acceso a múltiples usuarios de manera simultánea, la concurrencia al momento de edición no está permitida.
* La generación de estadísticas y métricas se basa en la información y los datos introducidos por los distintos usuarios, por lo que en caso de que esta información no sea adecuada, las estadísticas y métricas generadas por el sistema tampoco lo serán.
* El acceso al sistema depende de la correcta operación de la red local de la empresa o del proveedor de servicios de Internet.
* La asignación de restricciones y privilegios sobre el uso de la aplicación para los diferentes tipos de usuarios está preestablecida, por lo que el cliente no podrá configurar estos permisos al momento de la instalación del sistema.
* Las estadísticas y métricas generadas por el sistema fueron determinadas con anterioridad, por lo que el cliente no tendrá la posibilidad de agregar, modificar o eliminar estadísticas o métricas.

Las desventajas y limitaciones listadas en esta sección fueron las identificadas al momento de la realización del documento. Nuevas desventajas o limitaciones pueden surgir durante el desarrollo del proyecto.