

Desarrollo de una herramienta Web para monitorización y control de proyectos de Software para empresas con nivel 5 de CMMI-DEV

D. A. Ramos Calvario¹, A. M. Chávez Trejo¹, Z.B. Osorio Martínez²

¹División de Estudios de Posgrado e Investigación Instituto Tecnológico de Orizaba,

²Intellego S.A. de C.V.

darc_b_30@hotmail.com, achavez@itorizaba.edu.mx,
elpkbj@gmail.com

Resumen. En este trabajo se exponen los avances en el desarrollo de la herramienta Web para la monitorización y control de los proyectos de software en empresas de alto nivel de madurez de CMMI-DEV, las cuales están invirtiendo tiempo y recursos financieros en la búsqueda de la mejora continua y en la calidad de sus procesos, desarrollando herramientas que ayuden a su monitoreo. El modelo CMMI-DEV cuenta con las áreas de "Monitorización y Control" así como "Desempeño de los Procesos Organizacionales", las cuales apoyan a la monitorización de las líneas base de desempeño de la empresa en sus subprocesos críticos, así como sus límites de control y confianza, a fin de tomar acciones correctivas apropiadas cuando el rendimiento se desvía significativamente. La herramienta que se propone en este trabajo, atenderá la inexistencia de herramientas documentadas que apoyen a las áreas antes mencionadas para monitorizar las líneas base en una empresa de alta madurez, esta herramienta será desarrollada para la empresa Intellego S.A. de C.V. la cual posee un nivel 5 de madurez [1]. La automatización pretende una reducción de hasta un 50% de esfuerzo para la recolección y análisis de información, adaptándola a la herramienta de producción diaria (Jira).

Palabras clave: Herramienta Web, Proceso de Desarrollo de Software, Modelo de Calidad, Monitorización y Control de Proyectos.

1. Introducción

Actualmente las empresas de desarrollo de software que desempeñan procesos de producción y que se encuentran evaluadas en modelos de calidad, buscan estar siempre a la vanguardia, por lo que están invirtiendo en automatizar sus procesos [2]. Las empresas relacionadas en la mejora continua, son aquellas con alto nivel de madurez en modelos de calidad como CMMI-DEV, el cual es un modelo de mejora de procesos para el desarrollo de software. Una alternativa para la mejora es la automatización de los procesos y subprocesos en el desarrollo de sus productos [3]. CMMI-DEV

establece 22 áreas clave de proceso en el nivel más alto de madurez [4], empresas con nivel 5 en este modelo, requieren monitorizar el comportamiento de sus subprocesos, es por ello que se requiere una comprensión del rendimiento de los proyectos en los sub procesos que afectan de forma significativa los objetivos del proyecto y/o su organización, de modo que se identifiquen causas profundas que originan las desviaciones y posterior a ellos se definan mejoras que establezcan el comportamiento del proyecto. El proceso a aplicar contempla actividades de verificación y validación mediante pruebas formales y de aceptación dentro del entorno real de trabajo, las cuales serán realizadas por los usuarios finales, y con ello tener la certeza de que la herramienta satisfaga los requerimientos funcionales, por otra parte se recopilará una muestra de 10 proyectos en la ejecución de la herramienta para tener una muestra que compruebe que se disminuyó el esfuerzo asociado a la recolección y análisis de mediciones realizado con anterioridad.

2. Trabajos relacionados

En el ámbito de desarrollo de software Westfall Linda en [5] propone una serie de pasos para la selección, diseño e implementación de métricas de software posibles de aplicar en un proceso de desarrollo de software de calidad. Estos pasos se centran en la elección de diferentes entidades y atributos que ayuden a medir la calidad de los procesos de desarrollo de software los cuales se dividen en 3 fases, la primera fase indica cómo elegir las diferentes métricas y entidades posibles a tomar en cuenta, la segunda fase describe cómo diseñar y adaptar los indicadores seleccionados a este proceso y por último la tercera fase indica cómo elegir a las personas encargadas de la recolección de datos. En este trabajo relacionado cómo en el que se propone en este artículo son adaptativos al proceso actual de la empresa pero el trabajo de Linda Westfall carece de sustento en algún modelo de procesos y de una herramienta que apoye a su implementación. Por otro lado Pete Rotella y Sunita Chulani en [6], presentan la experiencia adquirida en Cisco System, Inc. en la adopción de un proceso de desarrollo de software de calidad unificado, para controlar sus departamentos y proyectos. Este control se observaba casi imposible ya que cada uno de los departamentos tiene su método de trabajo, para solucionar esto se desarrolló una herramienta llamada SWQD (Software Quality Dashboard, Panel de Calidad de Software), el cual inició con 13 indicadores comunes entre departamentos en el año 2011, y para el año 2012 se integraron entre 16 y 18 indicadores. Esta integración logró aumentar el porcentaje de aceptación entre los clientes de un 30% aumento a un 70%. Si bien este trabajo está enfocado al monitoreo de los diferentes departamentos, no se encuentra basado en modelos de calidad sino en similitudes entre cada uno de los departamentos, es por ello que este control solo es para prácticas específicas de la empresa dejando a un lado el propósito que CMMI-DEV provee para esta área. Otro trabajo interesante es el de Sutherland, J., et al [7], donde se plantea un estudio centrado en empresas de nivel 5 de madurez en CMMI, se propone institucionalizar métodos ágiles en su proceso de desarrollo de software, indicando que el uso selectivo de estos métodos disminuye la indisciplina dentro de las instituciones de desarrollo de software. En este

trabajo se proponen mezclar CMMI-DEV con SCRUM, mediante 12 prácticas genéricas relacionadas con los niveles 2 y 3 de CMMI, esta unión fortalece la adaptabilidad y la previsibilidad de los procesos. El propósito final de este trabajo es detectar errores por medio de una metodología, pero solo se logra adaptando nuevos procesos a lo actual, por lo cual esto genera un trabajo extra además de que quizá sea necesario un departamento especializado en estas prácticas, además de carecer de una herramienta que apoye al funcionamiento esperado, caso contrario al trabajo que se propone en el desarrollo de este artículo el cual es adaptado al proceso actual sin generar trabajo extra. Por otro lado Ferreira, A.I.F., et al en [8], presentan cómo una empresa logra evolucionar a niveles altos de madurez en CMMI, indicando que: La satisfacción del cliente, la mejora de la calidad y la adopción de la reutilización entre otros aspectos, son problemáticas comunes en la búsqueda de esta mejora continua. Se presenta el caso de la empresa BL Informatic's, donde se puso en marcha un plan de mejora que atiende la eliminación de la resistencia al cambio, la reducción de costos en los distintos procesos y áreas de producción, así como la estrategia para aumentar la motivación de sus colaboradores directos e indirectos, todo en base a diferentes normativas cómo CMMI nivel 3, MPS.BR Nivel F e ISO 9001:2000. Teniendo cómo resultado al finalizar el proyecto de BL Informatic's, una reducción en sus pérdidas económicas de manera significativa, esto debido a que la identificación de fallas y errores ahora se logra desde la fase de planeación, facilitando así la toma de decisiones con anticipación. Viendo este trabajo se llega a la conclusión que el propósito del proyecto de BL Informatic's está basado en lo esperado con la implementación del área de "Monitorización y Control de Proyectos" de CMMI-DEV pero dejando a un lado el trabajo mediante sus líneas base y límites de confianza. Por otro lado J. Pino, et al [2], muestran que en el mundo existen muchas empresas de desarrollo de software, las cuales son denominadas PyMES (Pequeñas y Medianas Empresas) con no más de 25 empleados, las cuales representan un ingreso económico alto en cualquier país, por ello es necesario que estas empresas tengan una comprensión y amplio conocimiento, en cómo implementar los diferentes estándares propuestos por los Indicadores de Editoriales Académicas (Scholarly Publishers Indicators) correctamente, tanto a nivel internacional cómo regional, en el desarrollo de proyectos de software. Para esto este trabajo propone una estrategia de selección y priorización de procesos, listando en primera instancia todos los procesos detectados en la empresa, teniendo como objetivo priorizar los procesos mediante las diferentes normativas de CMMI, ISO/IEC 15504, ISO / IEC 12207, entre otras. De igual manera se propone un método que permite la priorización de procesos, este método proporciona diferentes opciones para priorizar, cómo lo es ordenado por: Mejora de grupos de procesos, procesos de gestión, procesos de soporte y por último los procesos de ingeniería. Los autores exponen que la buena práctica de esta metodología, implica una mejora en la calidad de sus procesos de desarrollo de software en las PyMES, y que con el tiempo la productividad en los países que lo implementen logre un aumento significativo. Si bien este trabajo está enfocado en la búsqueda de la mejora continua, no está fielmente basada en una práctica que ayude a este propósito, al contrario del propuesto en este artículo que se basa en prácticas probadas y adaptadas al proceso implementado actualmente,

además de carecer de herramientas que apoyen este funcionamiento de manera automatizada.

Cómo se puede apreciar en los trabajos antes citados se menciona el desarrollo de herramientas de soporte al proceso de desarrollo, sin embargo no se menciona una herramienta específica para la monitorización y control de límites de control de proyectos en una empresa de nivel 5 de madurez del modelo CMMI-DEV. Hasta el momento con la investigación realizada no se conoce alguna herramienta semejante, que sea capaz de llevar este proceso con métricas probadas y establecidas.

3. Descripción de la herramienta propuesta

La herramienta propuesta establece los requerimientos mediante los casos de uso numerados a continuación:

1. Consultar Panel de Control
2. Crear Panel de Control
3. Generar Reporte de Gráfico de Control SPI, CPI, Esfuerzo
4. Verificar Productividad de Subprocesos Críticos
5. Generar Reporte de Revisión por Pares
6. Generar Reporte de Productividad y Defectos en Revisiones por Pares
7. Generar Reporte de Variación de Margen Financiero
8. Establecer Límites

En la figura 1 se representa mediante un diagrama de casos de uso, la interacción que tienen los dos actores identificados con cada uno de los casos de uso, estos actores son: “Intellego” y los “Líder de Proyecto”, ambos tienen acceso a los casos de uso de: “Consultar Panel de Control”, el cual proporciona al usuario la posibilidad de verificar o modificar los datos del panel de control que estén asignados a este usuario, su funcionamiento consiste en autocompletar los valores previamente registrados de todos los módulos del sistema. Por otro lado el caso de uso de “Crear Panel de Control” genera espacio de almacenamiento en la base de datos del sistema para un proyecto nuevo, esto para que posterior a ellos se visualicen los indicadores con la información de Jira en cada uno de los módulos para ese panel de control. El caso de uso “Generar Reporte de Gráfico de Control SPI, CPI, Esfuerzo”, da a conocer la tendencia de los indicadores mediante gráficos de control, dependiendo de las revisiones realizadas semanalmente, y en caso de encontrar desviaciones por fuera de los límites de control se lleva a cabo un análisis causal para posterior a ello identificar mejoras que estabilicen el comportamiento y el desempeño. Con respecto al caso de uso de “Verificar Productividad de Subprocesos Críticos”, se obtiene la productividad (horas por puntos de función) en operaciones críticas en el desarrollo de software definidas por Intellego y se calcula con base a los datos de esfuerzo en Jira y del modelo de estimación del proyecto para conocer el tamaño del módulo, después se visualizan los datos mediante gráficos de control donde se observa si el comportamiento del proyecto está dentro de los límites de control. Por otro lado en el caso de uso “Generar Re-

porte de Revisión por Pares” se obtienen los datos de los tipos de revisiones realizadas durante el desarrollo de los proyectos, así como defectos, esfuerzo asociado, producto revisado, entre otros y obtener el valor de re trabajo de cada una de ellas. El caso de uso de “Reporte de Productividad y Defectos en Revisiones por Pares”, se da a conocer la productividad en los siguientes tipos de revisiones: 1. Email, 2. Walkthrough, 3. Inspecciones, las cuales se efectúan en los proyectos, así como la densidad de defectos (defectos por cada punto de función), calculado mediante datos de Jira y del modelo de estimación del proyecto. Y por último dentro de los casos de uso compartidos para a los dos tipos de usuarios se encuentra el caso de uso “Generar Reporte de Variación de Margen Financiero” presenta el costos devengados, ingresos a facturar, margen financiero y acumulado financieros generados en periodos semanales. Por último existe un caso de uso que solo el actor "Intellego" tiene acceso, el cual es el de “Establecer Límites” donde se podrán modificar los límites de control y de confianza para cada uno de los proyectos, ya que de inicio se creará el panel de control con valores calculados automáticamente de acuerdo a datos históricos de la empresa. Estos casos de uso en conjunto forman la herramienta propuesta, la cual proporciona indicadores y apoya visualizando tendencias de los proyectos en desarrollo.

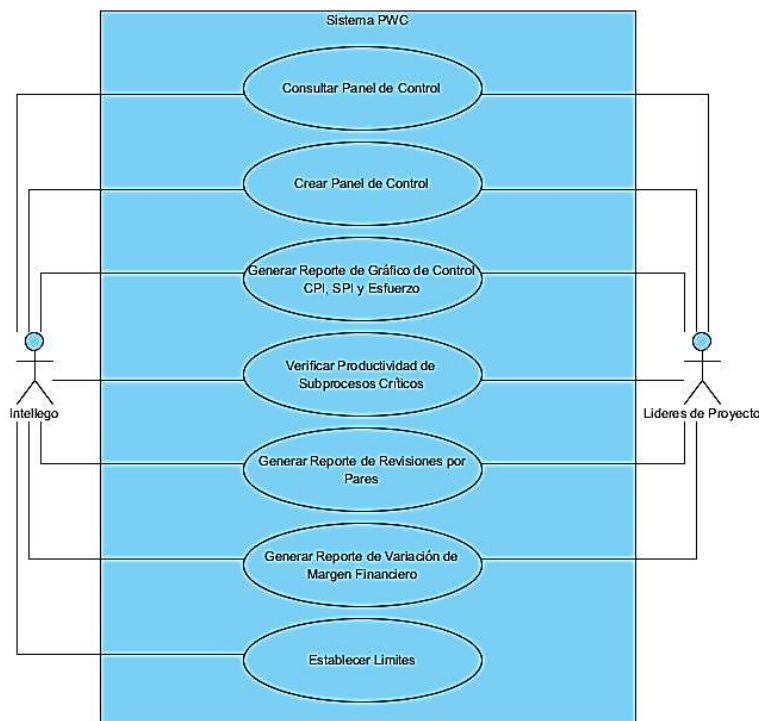


Figura 1. Diagrama de casos de uso – Interacción sobre la herramienta.

3.1 Arquitectura propuesta

La figura 2 muestra la propuesta de arquitectura describiendo la estructura del software a desarrollar, la cual se basa en el patrón arquitectónico en capas. A continuación se describen cada uno de los elementos que se encuentran en la arquitectura:

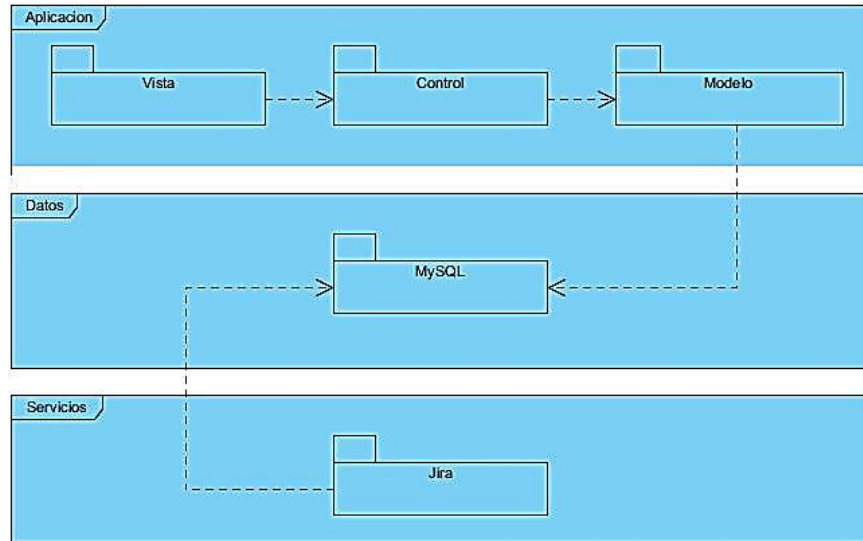


Figura 2. Arquitectura Propuesta.

Capa de Servicios: En esta capa se encuentra instalada la herramienta Jira la cual es un sistema de producción diaria donde se registra el esfuerzo en los procesos y sub procesos de la empresa, esta herramienta se encuentran alineada al nivel 5 del modelo CMMI-DEV, y es gestionada por los ingenieros de Intellego encargados de los proyectos en desarrollo. Además de que esta herramienta se encuentra interconectada a la capa de datos donde almacena la información registrada.

Capa de Datos: Contiene el gestor de base de datos donde se aloja el conjunto de datos generados por el cliente JIRA de cada uno de los proyectos desarrollados y en desarrollo por Intellego, los cuales son gestionados mediante el gestor de base de datos de MySQL. Cabe mencionar que dentro de este gestor existe un componente que contiene los indicadores para satisfacer la herramienta.

Capa de Aplicación: Contiene el código fuente de la herramienta, el cual adopta el patrón arquitectónico MVC (Modelo Vista Controlador). Esta capa depende directamente de la capa de datos, para el consumo de los indicadores necesarios.

En la figura 3 se muestra el diagrama de secuencia UML correspondiente al proceso general de la aplicación propuesta para Intellego, en el cual se identifica el siguiente flujo de trabajo:

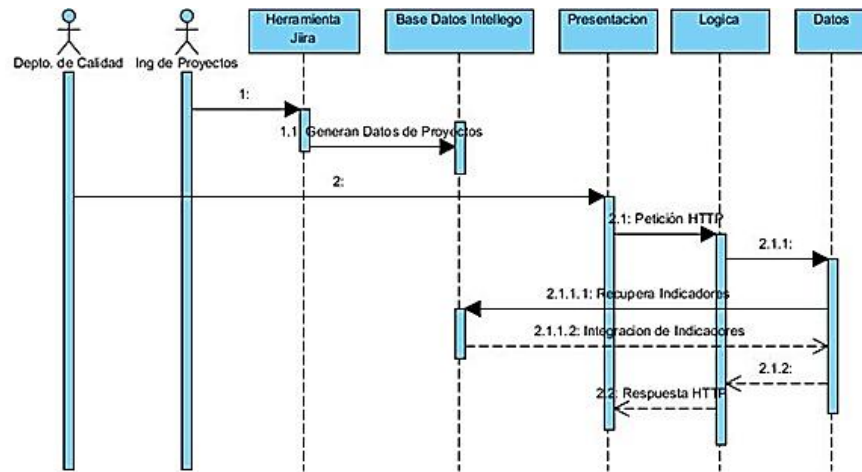


Figura 3. Diagrama de secuencia – Secuencia del proceso general de la aplicación.

1. Los ingenieros de proyectos acceden a la herramienta Jira, para ingresar los datos del proyecto en el que se encuentren laborando en ese momento, esto se lleva a cabo de acuerdo a sus fechas estipuladas.
2. Se genera dentro de la base de datos de Intellego, el registro de los datos del proyecto gestionado.
3. Los encargados de “Intellego” o los “Líderes del proyecto”, podrán ver la información mediante un navegador Web, ingresando a la herramienta en la capa de presentación.
4. El navegador realiza una petición HTTP al controlador de la herramienta, solicitando un proyecto específico, esto dentro de la capa lógica de la herramienta.
5. La capa lógica realiza la petición de los indicadores a la capa de datos, para que esta última retorne a la capa lógica los Indicadores del proyecto seleccionado.
6. Teniendo los indicadores disponibles del proyecto, la capa lógica retorna a la capa de presentación mediante una respuesta.
7. La capa lógica emite una respuesta HTTP con los valores, y con ellos mostrarlos en el navegador Web.

En la figura 4 se muestra un diagrama despliegue representando la integración de la herramienta dentro de la infraestructura actual de Intellego, donde el propósito es observar la distribución de los 3 elementos involucrados que a continuación se describen: Uno de ellos es el “Servidor de datos”, que es donde se encuentra el gestor de base de datos administrando los Indicadores de la empresa generados por JIRA, y

conectado a él mediante un protocolo TCP/IP se encuentra el “Servidor de aplicación”, donde se instalara la herramienta a desarrollar en el lenguaje ASP.Net y por último está el “Cliente”, el cual es considerado cualquier dispositivo que mediante el protocolo TCP/IP tenga acceso a la red de Intellego y que además posea un browser (Navegador Web) instalado, solo con estas características podrá hacer una petición HTTP al servidor de aplicación y así ingresar a la herramienta.

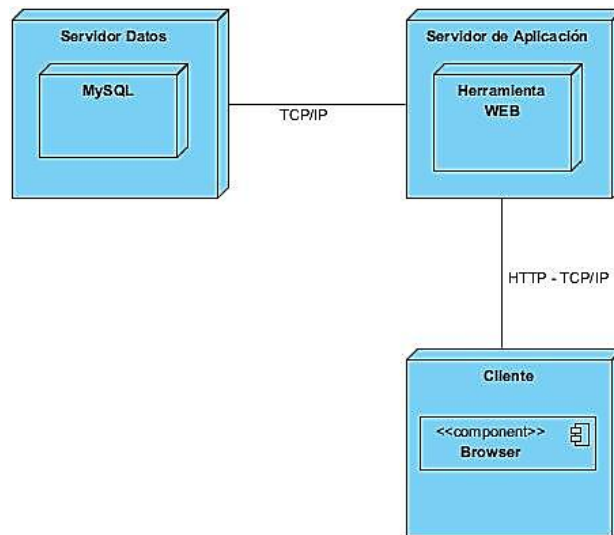


Figura 4. Despliegue de la Herramienta.

3.2 Tecnología a utilizar

La herramienta se desarrollará en el lenguaje ASP.Net, para integrarla adecuadamente a las tecnologías que maneja Intellego, esta tecnología es la más estable en sus servidores. Además, actualmente ASP.Net soporta 3 modelos de programación: ASP.Net Web Forms, ASP.Net MVC y ASP.Net Web Pages, de los cuales el modelo ASP.Net MVC al implementarse en el desarrollo, separa claramente la lógica de presentación, de negocios y la de acceso a datos.

En el caso del gestor de la base de datos MySQL se establece cómo repositorio propio de la herramienta, además de ser este el consumido por la herramienta JIRA y así facilitar la compatibilidad en la integración de ambas herramientas.

Para el desarrollo de la herramienta se implementa el ciclo de vida iterativo incremental, debido a que permite realizar el ciclo completo del proyecto, y en cada iteración se permite añadir funcionalidad o refinar los requerimientos, además se implementara la metodología Scrum que permite priorizar funcionalidades y resolver dudas del alcance con colaboración del "Product Owner", permitiendo así un desarrollo ágil.

Por otra parte en cada iteración se hará un caso de uso de la herramienta, y al finalizar cada uno de ellos se realizará las pruebas, además en el proyecto se elaborarán todos los entregables estipulados por la empresa Intellego.

4. Trabajo a futuro

Para la continuación del desarrollo de la herramienta, es necesario analizar la estructura de la base de datos de Intellego para recuperar la información requerida para visualizar los indicadores que provee el sistema. Teniendo este análisis inicia la construcción de los módulos que sustentara las especificaciones del área, mediante los requerimientos funcionales: 1. Consultar Panel de Control, 2. Crear Panel de Control, 3. Generar Reporte de Gráfico de Control SPI, CPI, Esfuerzo, 4. Verificar Productividad de Subprocesos Críticos, 5. Generar Reporte de Revisión por Pares, 6. Generar Reporte de Productividad y Defectos en Revisiones por Pares, 7. Generar Reporte de Variación de Margen Financiero y 8. Establecer Límites, además conforme se reporten los avances de la herramienta generar los entregables enfocados al nivel de madurez de la empresa Intellego.

Una vez construida la herramienta se procederá al pilotaje en su totalidad de la herramienta a fin de tener una muestra significativa de proyectos que permitan analizar el ahorro de esfuerzo obtenido y comparar respecto a la meta planteada.

5. Conclusiones

Finalizando esta etapa de desarrollo de la herramienta, se observa que el monitoreo de los proyectos que se tiene actualmente mediante plantillas en Excel, si bien abarca el funcionamiento requerido para llevar a cabo esta actividad, no disminuye el trabajo realizado, ya que la recolección de los diferentes indicadores de la interfaz de JIRA genera un trabajo extra en sus actividades normales, por ello la herramienta se considera viable para apoyar el funcionamiento del área de CMMI-DEV llamada “Monitorización y Control de Proyectos” y con ello reducir el esfuerzo en un 50% de actual.

Agradecimientos

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por la beca otorgada para la realización de los estudios de Maestría, y a la empresa Intellego por su disponibilidad en información y por el apoyo brindado para la publicación de este artículo.

Referencias

1. Intellego. <http://www.intellego.com.mx/country-selector>. 09-Abril-2013.
2. Francisco J. Pino, Félix García, Mario Piattin. Key processes to start software process improvement in small companies. Proceeding SAC '09 Proceedings of the 2009 ACM symposium on Applied Computing, 2009, PP: 509-516.
3. Olga Liliana Castrillón Giraldo, Gustavo Adolfo Mina Valencia. Framework para gestión de conocimiento de equipos de desarrollo de software basado el modelo CMMI DEV 1.3. Colombia, 2011.
4. Mike Konrad, Sandy Shrum. CMMI (2a. Ed.): Guía para la integración de procesos y la mejora de productos, Addison-Wesley, 2009.
5. Linda Westfall. 12 Steps to Useful Software Metrics. The Westfall Team, 2005.
6. Pete Rotella, Sunita Chulani. Implementing quality metrics and goals at the corporate level. Proceeding MSR '11 Proceedings of the 8th Working Conference on Mining Software Repositories, 2011, PP: 113 - 122.
7. Jeff Sutherland, Carsten Ruseng Jakobsen, Kent Johnson. Scrum and CMMI Level 5: The Magic Potion for Code Warriors. Proceeding HICSS '08 Proceedings of the Proceedings of the 41st Annual Hawaii International Conference on System Sciences, 2008, PP: 466.
8. Analia Irigoyen Ferreiro Ferreira, Gleison Santos, Roberta Cerqueira, Mariano Montoni, Ahilton Barreto, Andrea O. Soares Barreto, Ana Regina Rocha. Applying ISO 9001: 2000, MPS.BR and CMMI to Achieve Software Process Maturity: BL Informatica's Pathway. Proceeding ICSE '07 Proceedings of the 29th international conference on Software Engineering, 2007, PP: 642 - 651.