UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

TIA
Tecnología, Investigación y Academia

Prototipo para la gestión de cambios con base en el marco de trabajo COBIT 5

Prototype for change management based on the COBIT 5 framework

Luis Alfredo González Riscanevo[1]

Para citar este artículo: : González, L. A. (2018). Prototipo para la gestión de cambios con base en el marco de trabajo COBIT 5. *TIA*, 6(1), pp. 47-53.

Artículo de reflexión

Fecha de recepción:
25-11-2014
Fecha de aceptación:
05-04-2017

ISSN: 2344-8288
Vol. 6 No. 1
Enero - junio 2018
Bogotá-Colombia

Resumen

La gestión de cambios es uno de los procesos clave para toda organización dedicada al desarrollo y mantenimiento de software, esta ayuda a una mejor organización, facilitando el resto de procesos dentro de la compañía. Los temas de gestión de cambios suelen tener como principal barrera la desconfianza, las consecuencias que puedan acarrear y la falta de motivación. La gestión efectiva incluye la participación de los stakeholders en este proceso; sin embargo, en el mercado de software no se encuentra una herramienta tecnológica que almacene la información relevante de la gestión de cambios con base en el marco de trabajo ITIL v. 5. De acuerdo con el problema identificado, se hace importante el desarrollo de una herramienta enfocada a brindar información confiable, completa, detallada y oportuna de la gestión de cambios.

Palabras clave: COBIT, diseño basado en componentes, gestión de cambios, OpenUP, programación.

Abstract

Change management is one of the key processes for any organization dedicated to the development and maintenance of software. It helps a better organization, facilitating the rest of processes within the company. Change management issues tend to have as main barrier distrust, consequences that may result and lack of motivation. Effective management includes participation of stakeholders in this process; however, in the software market there is not a technological tool that stores relevant information of change management based on the ITIL v. 5 framework. According to the problem identified, it's important to develop a tool focused on providing reliable, complete, detailed and timely information on change management.

Keywords: COBIT, component-based design, change management, OpenUP, programming.

[1] Ingeniero de Sistemas de la Universidad Distrital Francisco José de Caldas. Correo electrónico: lagonzalezr77@gmail.com

INTRODUCCIÓN

En un mundo globalizado y tan competitivo como el actual, la forma para continuar en el mercado es adaptarse a las nuevas tendencias. Cada vez más, los negocios tienen que aumentar su agilidad para lanzar y posicionar productos y servicios en el mercado antes que la competencia. Las organizaciones, impulsadas por la tecnología, tienen que asegurarse de que sus sistemas de información puedan adaptarse a las necesidades del negocio rápida y eficientemente.

La gestión de cambios basada en el marco de trabajo COBIT 5 permite que el área de tecnología sea más estable y pueda responder mejor a las necesidades del negocio; el área de tecnología debe ser un aliado estratégico de cara a los objetivos con que cuenta una empresa.

El actual proyecto pretende ser un soporte y herramienta útil para la gestión de cambios por medio del uso de instrumentos tecnológicos de *software*, a fin de lograr agilidad, oportunidad y eficacia en el proceso.

HIPÓTESIS

La gestión de cambios para la producción de las aplicaciones en una compañía no cuenta con un aplicativo que conlleve a realizar un control de actividades derivadas de dicho proceso, no es posible gestionar eficientemente los indicadores de evaluación de coste y de tiempo, además de no monitorear cuándo se producen cambios sin controlar los incidentes producidos y su clasificación; teniendo en cuenta lo anterior, se hace necesario crear un prototipo para la gestión cambios tomando como referencia el marco de trabajo COBIT 5, el cual permite controlar el ciclo de vida de todos los cambios midiendo el impacto en el alcance, tiempo y coste del proyecto o requerimiento según sea definido en la fase de planificación. El propósito de este prototipo es asegurar que los cambios se registren, evalúen, prioricen, planifiquen, queden documentados y se aprueben de forma controlada.

ANTECEDENTES

Software Assyst de gestión de cambios basado en ITIL [1]

Assyst incluye en una sola aplicación funciones de gestión de cambios, entregas y configuraciones basadas en mejores prácticas. Ofrece soporte para procesos de cambio, evaluación del impacto sobre el negocio y control de autorizaciones, garantizando de esta forma una mayor tasa de éxito en la implementación de cambios; integra la función de gestión de cambios con la automatización de todos los procesos ITIL en una sola aplicación, lo que permite implementarla en entregas y configuraciones sin necesidad de realizar proyectos de integración complejos.

En Assyst, la gestión de entregas se encuentra estrechamente integrada con la gestión de cambios, esto permite controlar fácilmente todo el proceso, desde la petición del cambio (RFC) hasta la implementación como parte de un único proceso definido, sin integrar herramientas diferentes de cambio y lanzamiento. El diseñador de procesos de Assyst permite crear procesos para responder a las necesidades del negocio mediante el método de "desplazar y soltar", creando procesos nuevos desde cero o modificando los modelos que ya vienen con Assyst.

GMF Opensource [2]

Es un *software* que implementa las recomendaciones ITIL (*IT Infrastructure Library*) para la gestión de servicios de TI (*IT Service Management* o ITSM). GMF es un producto de *software* libre distribuido bajo licencia GPL e incluye módulos de gestión de incidencias

(*trouble ticketing*), gestión de inventario, gestión del cambio (*change management*), SLA y *reporting*.

ITIL es una serie de mejores prácticas (*best practices*) orientadas a ayudar a las organizaciones con la implementación de ITSM. Se trata de una orientación

a negocio de la gestión de las TI para ofrecer un nivel de servicio elevado y ser capaz de cuantificar el valor generado por la infraestructura de TI. Para ello, se centra en la gestión de los procesos y de las personas.

La ventaja de la implantación de una herramienta ITSM es la de disponer de un método sistemático y profesional para la gestión de los servicios, sus beneficios son múltiples:

- Reducción de costes.
- Mejora de la calidad.
- Mejora de la satisfacción del cliente (interno o externo).
- Incremento de la productividad.
- Definición de procesos estándares.

GMF dispone actualmente de los siguientes módulos:

- *Incident management* o módulo de gestión de incidencias. Gestión y control de incidencias.
- *Change management*. Gestión y control de peticiones de cambio (*Request for Changes* o RFC).
- *Service Level Management*. Gestiona la calidad de los servicios ofrecidos al cliente y busca su mejora constante mediante la revisión de los parámetros de control.
- *Reporting*. *Reports* en formato HTML y PDF de la evolución de distintas variables relacionadas con los SLA, las incidencias y los cambios.
- También incorpora un módulo básico para mantener inventario *hardware*.

MARCO CONCEPTUAL

COBIT 5 [3]

Una sólida plataforma que subyacente a la norma ISO 38500 es COBIT, es una buena referencia para las políticas, los procesos, las estructuras y los controles necesarios para implementar el sistema de gestión que ayuden a la gobernabilidad. Con COBIT se puede lograr la alineación de TI al negocio y utilizarlo como punto de partida para la adaptación de los procedimientos específicos.

ISACA publicó en el 2012 una nueva versión del ya conocido estándar COBIT —para el cumplimiento de objetivos de control— para el CIO y su área; esta versión, profundamente revisada y mejorada, provee un marco de referencia integral que contribuye en la organización al logro de los objetivos y entrega de valor a través de un efectivo gobierno y gestión de la TI empresarial.

COBIT 5 provee de un marco de trabajo integral que ayuda a las empresas a alcanzar sus objetivos para el gobierno y la gestión de las TI corporativas; dicho de una manera sencilla, ayuda a las empresas a crear el valor óptimo desde TI manteniendo el equilibrio entre la generación de beneficios y la optimización de los niveles de riesgo y el uso de recursos.

COBIT 5 permite a las TI ser gobernadas y gestionadas de un modo holístico para toda la empresa, abarcando al negocio completo de principio a fin y las áreas funcionales de responsabilidad de TI, considerando los intereses relacionados con TI de las partes interesadas internas y externas. COBIT 5 es genérico y útil para empresas de todos los tamaños, tanto comerciales como sin ánimo de lucro o del sector público.

El marco COBIT 5 se construye sobre cinco principios básicos y siete catalizadores o habilitadores para el gobierno y la gestión de TI de la empresa.

Gestión de cambios [4]

La gestión de cambios es hacer que los productos y servicios se integren en el entorno de producción y sean accesibles a los clientes y usuarios autorizados.

Sus principales objetivos se resumen en:

- Supervisar y dar soporte a todo el proceso de cambio del nuevo (o modificado) servicio.
- Garantizar que los nuevos servicios cumplann los requisitos y estándares de calidad estipulados en las fases de estrategia y la de diseño.
- Minimizar los riesgos intrínsecos asociados al cambio reduciendo el posible impacto sobre los servicios ya existentes.
- Mejorar la satisfacción del cliente respecto a los servicios prestados.
- Comunicar el cambio a todos los agentes implicados.

Para cumplir adecuadamente estos objetivos es necesario que durante la gestión de cambios se realicen las siguientes acciones:

- Se planifique todo el proceso de cambio.
- Se creen los entornos de pruebas y preproducción necesarios.
- Se realicen todas las pruebas necesarias para asegurar la adecuación del nuevo servicio a los requisitos predefinidos.
- Se establezcan planes de *roll-out* (despliegue) y *roll-back* (retorno a la última versión estable).
- Se cierre el proceso de cambio con una detallada revisión post-implementación.

Como resultado de una correcta gestión de cambio se tiene que:

- Los clientes disponen de servicios mejor alineados con sus necesidades de negocio.
- La implementación de nuevos servicios es más eficiente.
- Los servicios responden mejor a los cambios del mercado y a los requisitos de los clientes.
- Se controlan los riesgos y se dispone de planes de contingencia que eviten una degradación prolongada del servicio.
- Se mantienen correctamente actualizadas las bases de datos de configuración y activos del servicio.
- Se dispone de una base de conocimiento actualizada a disposición del personal responsable de la operación del servicio y sus usuarios.

Proceso de desarrollo OpenUP [5]

El proceso OpenUP es un método de trabajo que involucra un conjunto mínimo de prácticas tendientes a guiar a un equipo de trabajo pequeño en el análisis, diseño, desarrollo y despliegue de un producto de *software*. Los objetivos que persigue son:

- Promover la colaboración y compartir conocimientos alineando intereses del equipo de trabajo y los usuarios.
- Ayudar al equipo a enfocarse en la arquitectura de forma rápida de tal forma que se minimicen los riesgos y se organice el desarrollo.
- Ayudar al equipo a balancear prioridades en conflicto para maximizar el valor obtenido por los interesados en el proyecto.
- Ayudar al equipo en la evolución continua del producto para obtener retroalimentación continua y fomentar el mejoramiento.
- Permitir a los administradores del proyecto realizar seguimiento a los avances basados en metas e indicadores.
- Permitir que los integrantes del equipo entiendan rápidamente cómo realizar el trabajo para alcanzar los objetivos y metas proyectadas.

Los principios en que se enmarcará el método de trabajo OpenUP/OAS son mencionados a continuación.

- Conocer a los interesados: se deben identificar a los grupos de interés y trabajar de cerca con ellos para asegurarse de que sus necesidades son claramente definidas y satisfechas a medida que se evoluciona en el desarrollo de la solución. Debe mantenerse una comunicación abierta y frecuente, además de una colaboración entre ellos y el equipo de trabajo.
- Separar el problema de la solución: se debe estar seguro de conocer el problema (o una parte de él) antes de definir una solución (o una parte de ella). Al separar claramente el problema (qué necesita el cliente, no qué necesita el equipo de desarrollo) de la solución (el sistema que tiene que hacer), es fácil mantener un enfoque y encontrar vías alternativas para solucionar el problema.
- Crear un conocimiento compartido del dominio: se debe fomentar un ambiente de intercambio y trabajo en el que todos los involucrados puedan obtener constantemente la información adecuada para lograr tener una visión compartida de lo que se debe hacer, el por qué hacerlo y cómo se está haciendo.
- Usar escenarios y casos de uso para capturar requerimientos: permite que los interesados alcancen rápidamente un consenso acerca de sus necesidades e intereses.
- Establecer y mantener contratos de prioridades: se deben priorizar los requisitos y requerimientos de implementación basados en un trabajo continuo con los grupos de interés, también tomar decisiones que lleven a que el sistema siempre incremente los beneficios ofrecidos y reduzca los riesgos.
- Realizar negociaciones que maximicen el beneficio obtenido: las negociaciones costo-beneficio dentro del proyecto no pueden ser independientes de la arquitectura. Los requisitos y requerimientos establecen los beneficios que se deben alcanzar al implementar el sistema mientras que la arquitectura es una medida base para calcular su costo. El costo asociado con un beneficio puede influenciar en gran medida la percepción del usuario acerca del valor real obtenido.
- Gestionar el entorno: el cambio es inevitable, y aunque presenta oportunidades para mejorar los beneficios dados a los grupos de interés, un entorno incontrolado de cambios fácilmente decantará en sistemas deficientes, sobredimensionados y que no satisfacen las necesidades reales de los clientes. Se deben gestionar los cambios manteniendo contratos específicos con los grupos de interés.
- Conocer cuándo se debe parar de cargar características :no solo es una pérdida de tiempo y recursos, sino que conduce a sistemas innecesariamente complejos. El desarrollo debe parar cuando la calidad esperada del sistema se alcanza.
- Mantenga un entendimiento común: sea proactivo comunicando y compartiendo información con los participantes del proyecto y no asuma que todos y cada uno encontrarán justo lo que ellos necesitan saber o que cada persona tiene la misma comprensión del proyecto que todos los demás.
- Aprender continuamente: desarrolle continuamente sus habilidades técnicas e interpersonales, aprenda de los ejemplos de sus colegas, aproveche la oportunidad, tanto de ser un estudiante de sus colegas como maestro de ellos. Siempre incremente su habilidad personal para sobrellevar su propio antagonismo hacia otros miembros del equipo.
- Organice alrededor de la arquitectura: la comunicación entre los miembros del equipo empieza a ser compleja; por consiguiente, organice el equipo alrededor de la arquitectura, el vocabulario y el modelo mental compartido del sistema.
- Desarrolle su proyecto en iteraciones: divida su proyecto en una serie de iteraciones encajadas en el tiempo y planee su proyecto iterativamente.

Esta estrategia iterativa lo habilita para entregar capacidades como un conjunto ejecutable, subconjunto utilizable de requisitos y requerimientos probados e implementados que pueden ser evaluados por los interesados al final de cada iteración.

- Gestione los riesgos: ataque tempranamente los riesgos que viciarán el proyecto. Continuamente identifique y priorice los riesgos y entonces idee estrategias para mitigarlos.
- Adopte y gestione el cambio: la adopción ayuda a construir un sistema que se encamina a las necesidades de los interesados y el manejo permite reducir costos y mejorar la predicción de estos. Los cambios hechos tempranamente en el proyecto se pueden hacer usualmente a bajo costo; sin embargo, a media que se avanza en el proyecto, pueden empezar a incrementarse en términos de costos.
- Mida el progreso objetivamente: si no conoce objetivamente cómo el proyecto está progresando, no sabe si este falla o tiene éxito. La incertidumbre y los cambios a un proyecto de *software* en progreso dificultan medirlo objetivamente, en tanto que las personas tienen una habilidad muy asombrosa para creer que todo está bien ante la catástrofe.

METODOLOGÍA

Enfoque empírico-analítico

La presente investigación es un estudio empírico-analítico en la que se establecerá el diseño de un prototipo para la gestión de cambios tomando como base el marco de trabajo COBIT 5.

Desarrollo ingenieril

La metodología ingenieril que se utilizará para el desarrollo es la metodología OpenUP, la cual es un modelo de desarrollo de *software* que se centra en la producción y mantenimiento de modelos del sistema más que en producir documentos; consta de cuatro fases que se aplicarán en el proyecto de la siguiente manera [6]:

- Fase de inicio: se establecen los objetivos, límites y alcances del prototipo, se especifica claramente lo que se desea realizar con esta propuesta. Esencialmente esta fase responde a las preguntas ¿cuáles son las principales funciones del sistema para sus usuarios más importantes?, ¿cómo podría ser la arquitectura del sistema? y ¿cuál es el plan del proyecto y cuánto costara desarrollar el producto?
- Fase de elaboración: se establecen los programas y arquitecturas a utilizar, se desarrollan casos de uso que ilustrarán mejor el modelo generando ideas claras sobre los usuarios, actividades y tareas a desarrollar. Se detectarán las principales fuentes de riesgos y su probable impacto sobre el proyecto buscando su disminución y mejor tratamiento.
- Fase de construcción: se busca alcanzar la capacidad operacional del producto, se implementan, integran y prueban todos los componentes, características y requisitos del sistema. Se busca que la base de cumpla con los requerimientos y expectativas realizados por los usuarios.
- Fase de transición: se realizan pruebas durante todo el proyecto pero en esta etapa se deben realizar con todos los aplicativos funcionando y buscando detectar errores finales. Se realiza la socialización del aplicativo a los integrantes del departamento de ajustes y de los centros de servicio al cliente realizando últimos ajustes y resolviendo las dudas que se puedan generar; sin embargo, para esta propuesta no se llegará hasta esta etapa debido a que es un prototipo.

REFERENCIAS

[1] Assyst. (2014). Recuperado de https://www.axiossystems.com/itsm-solutions

[2] GMF (2014). *Gestión de incidencias*. Recuperado de http://www.genos.es/gestor-de-incidencias

[3] COBIT. (2014). Recuperado de http://www.isaca.org/COBIT/pages/default.aspx

[4] Osiatis. (2014). Recuperado de http://faquinones.com/gestiondeserviciosit/itilv3/transicion_servicios_TI.php

[5] OpenUp. (2014). Recuperado de https://portalws.udistrital.edu.co/CIT/documentos/NORMATIVIDAD/openupoas/archivos/GuiaRapidaOpenUPOAS.pdf

[6] Larman, C. (2006). *UML y patrones, introducción al análisis y diseño orientado a objetos*. Madrid: Prentice Hall.