

8. Sistemas de información Heredados y reingeniería de software

Son aquellos a los que es necesario aplicarles reingeniería. En el segundo subtema se emprende por completo el concepto de reingeniería de software y sus objetivos. En el siguiente subtema se menciona la importancia de aplicar reingeniería de software en los sistemas de información heredados. Al final de este Capítulo, se explica el modelo que propone Sneed para calcular los costes de un proyecto de reingeniería.

8.1 Sistemas de información heredados

Los sistemas de información heredados generalmente son la columna vertebral del flujo de información de las empresas y la principal forma de agruparla. Un Sistema de Información Heredado (LIS por sus siglas en inglés Legacy Information System) puede ser definido como "cualquier sistema de información que significativamente se resiste a la modificación y evolución" [Brod 94]. Tales LISs pueden causar serios problemas a la organización:

Los LISs casi siempre son ejecutados sobre hardware obsoleto que son lentos y caros de mantener.

El mantenimiento del software puede ser caro, porque carecen de la documentación necesaria para el entendimiento de los detalles del sistema y su seguimiento es costoso y consume mucho tiempo.

Una falta de interfaces limpias hace que la integración de los LISs con otros sistemas sea difícil.

Los LISs son también difíciles mas no imposibles ampliarlos.

La solución a estos problemas según Weiderman [Weid 97] cae en las siguientes categorías:

Mantenimiento: es un proceso incremental e iterativo en el cual se hacen pequeñas modificaciones al sistema.

Modernización: implica cambios más extensos que el mantenimiento pero conserva partes considerables del sistema existente.

Remplazarlo: el cual consiste en reconstruir desde los inicios al sistema. Esta solución consiste en aplicarle al sistema actividades de Reingeniería.

8.2 Reingeniería del Software

Reingeniería de Software es una forma de modernización para mejorar las capacidades y/o mantenibilidad de los sistemas de información heredados mediante la aplicación de tecnologías y prácticas modernas. La Reingeniería de Software ofrece una disciplina de preparación para migrar un sistema de información heredado hacia un sistema evolucionable. El proceso aplica principios de ingeniería para un sistema existente para encontrar nuevos requerimientos.

"El Instituto de Ingeniería de software (SEI) desarrollo una definición de Reingeniería como: Reingeniería es la transformación sistemática de un sistema existente dentro de una nueva forma de realizar mejoramientos de calidad en unas operaciones, capacidad del sistema, funcionabilidad, rendimiento o evolucionabilidad a bajo costo, agendas o riesgos para el cliente." [Till 95]

El propósito de la reingeniería es que los sistemas existentes tomen ventajas de las nuevas tecnologías y habilitar el nuevo esfuerzo de desarrollo para que aproveche las ventajas de reutilizar sistemas existentes. La reingeniería tiene el potencial de mejorar la productividad y calidad del software a través de todo el ciclo de vida.

La reingeniería casi siempre implica cambiar la forma de un programa y mejorar su documentación. En este caso, la funcionabilidad del programa no es cambiada; sólo su forma es modificada. En otros casos, la reingeniería va más allá de la forma e incluye rediseñar para cambiar la funcionabilidad del programa para buscar mejores requerimientos de usuario.

Los objetivos de la reingeniería son: [McCl 92]

Proporcionar asistencia automatizada para el mantenimiento.

Reducir los errores y costos del mantenimiento.

Incrementar la intercambiabilidad del grupo de mantenimiento.

Hacer sistemas fáciles de entender, cambiar y probar.

Habilitar la conversión y migración de sistemas.

Reforzar el apego a estándares.

Mejorar la respuesta a peticiones de mantenimiento.

Mejorar el estado de ánimo del grupo de mantenimiento.

Proteger y extender la vida del sistema.

Usar CASE para apoyar sistemas existentes

Re-usar componentes de sistema existentes.

La reingeniería puede ayudar a entender sistemas existentes y descubrir los componentes de software que son comunes en todo el sistema. Estos componentes comunes pueden ser re-usados en el desarrollo (o redesarrollo) de sistemas y así reducir el tiempo y riesgos del desarrollo de sistemas.

La reingeniería requiere tiempo; implica un coste de dinero enorme y absorbe recursos que de otro modo se podrían emplear en preocupaciones más inmediatas. Por todo esto, la reingeniería no se lleva a cabo en unos pocos meses, ni siquiera en unos pocos años. La reingeniería de sistemas de información es una actividad que absorberá recursos de las tecnologías de la información durante muchos años.

8.3 La importancia de aplicar la reingeniería de software

8.3.1 Las Viejas Aplicaciones

Mucha gente al ver las grandes y viejas mansiones queda asombrado de su belleza, pero no se preguntan que tan bien se puede vivir en ellas. Las personas que lo hacen dicen que es una pesadilla mantenerlas. Todas ellas fueron construidas con viejas tecnología estándar. Sus paredes externas no tienen aislamiento. El alambrado eléctrico tiene limitaciones y claramente es inadecuada para las necesidades de energía de hoy y su cableado decadente crea un severo peligro eléctrico.

Los viejos sistemas son muy similares a los grandes y viejos edificios. Ellos tienen los mismos problemas de mantenimiento, un hecho en gran parte irreconocible por parte de la comunidad corporativa. Muchos de esos edificios son demolidos por que no son mantenibles y ya no sirven para las necesidades de sus ocupantes.

Las viejas computadoras tal vez se puedan ver solamente en museos. Pero en muchos casos, software escrito para viejos modelos de computadora están ejecutándose hoy en día. Un caso extremo es el de un software escrito para una IBM 1401 Autocoder. Cuando la compañía reemplazó la 1401 con una IBM 360/40, compraron un emulador de la 1401 para poder ejecutar el software. Esa aplicación hoy día corre en una PC – la compañía compro otro emulador.

Los clientes demandan que las nuevas capacidades sean agregadas al código escrito en sus viejos sistemas. Casi siempre, las empresas encuentran que no pueden

modificar su código – el programador que lo mantenía murió recientemente o nadie sabe programar en el lenguaje en el que fue escrito. Por lo que la funcionalidad de ese programa quedará así para siempre.

La siguiente lista son las razones por las que es aplicable la reingeniería a los sistemas de información heredados:

- Frecuentes fallas de producción (fiabilidad cuestionable).
- Problemas de rendimiento.
- Tecnología obsoleta.
- Problemas de integración del sistema.
- Código de calidad pobre.
- Dificultad (peligroso) al cambio.
- Dificultad para probar.
- Mantenimiento caro.
- Incremento de problemas del sistema.

Estas razones pueden ser solucionadas al aplicar un proceso de mantenimiento de software, pero cuando dicho mantenimiento deja de ser viable, entonces se toma la decisión de aplicar reingeniería.

8.3.2 Nuevas Ideas, Nuevo Software

Aunque la reingeniería se usa principalmente durante el mantenimiento del software, va más allá de una simple ayuda para el mantenimiento. La reingeniería es el puente desde viejas tecnologías hacia nuevas tecnologías que las organizaciones deben usar en la actualidad para responder al cambio de requerimientos del negocio.

Los viejos programas representan la tecnología del ayer. Ahora sabemos que los años tienen cuatro dígitos y no dos, que los datos pueden ser manejados mejor en bases de datos y que tenemos nuevos diseños de construcción y lenguajes de programación que permiten diseñar programas notablemente mantenibles.

Cuando el costo de mantener viejos edificios es altamente excesivo, se reemplazan estos edificios. Nosotros deberíamos hacer lo mismo con los programas. Los programas no se hacen obsoletos al paso del tiempo ya que fueron escritos para hardware y sistemas operativos que ya no existen, muchos están llenos de características y parches no documentados.

Sólo cuando hayamos aprendido a que es mejor invertir en nuevo hardware y nuevos edificios podremos reconocer el valor de reemplazar los viejos sistemas raquíticos.

8.4 Costos y beneficios de la reingeniería

Antes de reconstruir un sistema en uso, es altamente recomendable analizar las diversas alternativas disponibles:

- Dejar el producto como está.
- Adquirir uno en el mercado que realice la misma función.
- Reconstruirlo.

Evidentemente, elegiremos la opción que mejor relación coste/beneficio nos ofrezca. Para calcular los costes de un proyecto de reingeniería, Harry Sneed [Snee 95] propone un modelo basado en cuatro etapas:

- Justificación del proyecto de reingeniería.
- Análisis de la cartera de aplicaciones.
- Estimación de costes.
- Análisis de costes / beneficios.

8.4.1 Justificación Del Proyecto De Reingeniería.

Para justificar un proyecto de reingeniería se requiere de un análisis del software existente, de los procesos de mantenimiento actuales y del valor de negocio que tienen las aplicaciones; todo esto con el objeto de hacer una evaluación en posibles aumentos de valores sobre estos tres factores.

La mayoría de las organizaciones sólo toman en consideración los procesos de reingeniería cuando el coste de un nuevo desarrollo es demasiado alto. En cualquier caso, y aunque a primera vista parezca la única o la mejor alternativa, es necesario confirmar la necesidad de reconstruir el sistema.

Existen cuatro operaciones que nos pueden dar una idea de los costes del proyecto y del valor del software actual dentro del negocio:

Introducción de un sistema de evaluación de los costes del mantenimiento. Es recomendable que esta tarea la lleve a cabo la organización anticipándose con suficiente anticipación al momento en que se percibe la necesidad de aplicar reingeniería.

Análisis de la calidad del software actual, para lo cual pueden utilizarse auditores de código automáticos que proporcionan datos del tamaño, complejidad y métricas de calidad del código fuente. Estos valores son incorporados a una base de datos que es utilizada por otra herramienta para realizar comparaciones y obtener resultados.

Análisis de los costes de mantenimiento: Se proponen tres métricas para medir los procesos de mantenimiento: "Dominio del impacto" o proporción de instrucciones y elementos de datos afectados por una tarea de mantenimiento con respecto al total de instrucciones y elementos de datos del sistema; "Esfuerzo empleado", que es el número de horas dedicadas a tareas de mantenimiento, con lo que se puede obtener una media del número de horas por tarea de mantenimiento; y "Tasa de errores de segundo nivel", que es el número de errores causados por acciones de mantenimiento. Si se observa que estas tres medidas se incrementan, es muy probable que los costes de mantenimiento se incrementen con el tiempo.

Evaluación del valor de negocio del sistema actual, que es realizado por la dirección de la organización.

8.4.2 Análisis De La Cartera De Aplicaciones.

En esta etapa se cotejan la calidad técnica y el valor de negocio de cada aplicación, con el objetivo de construir una lista de aplicaciones, ordenada según sus prioridades en el proceso de reingeniería.

La calidad técnica de un producto es una medida relativa, dependiente de cada organización, que se calcula en función de diversas características (complejidad ciclomática o errores/KLDC, por ejemplo).

Para cada variable que interviene en la calidad técnica e fijan unos límites inferior y superior (que representan los valores máximos y mínimo de calidad). Para hallar el nivel de calidad de la variable considerada se puede utilizar la siguiente formula:

Por ejemplo, si establecemos los valores mínimo y máximo de calidad en 0 y 7 errores por KLDC, y actualmente hay 3, $C_i = 0.571$.

Asociando un punto de un plano para cada aplicación, e interpretando el valor de negocio y la calidad técnica como coordenadas de estos puntos, se puede representar como en el diagrama de la siguiente figura [Piat 00]:

Las aplicaciones situadas en el cuadrante superior izquierdo tienen alta calidad y bajo valor de negocio, por lo que no requieren reingeniería; las situadas en el cuadrante inferior izquierdo tienen poco valor en ambos parámetros, por lo que pueden ser desarrolladas de nuevo o remplazadas por productos comerciales; las del superior derecho tienen un gran valor de negocio y alta calidad: se les puede aplicar reingeniería, pero sin excesiva prioridad; las del inferior derecho tienen alto valor de

negocio y baja calidad técnica, por lo que serán las primeras candidatas a la reingeniería.

8.4.3 Estimación De Costes.

Se realiza identificando y ponderando, mediante métricas adecuadas, todos los componentes del software que se van a modificar.

Se deben considerar los costes de cada proyecto de reingeniería: si éstos son superiores a los beneficios, la reingeniería no será una alternativa viable y la aplicación deberá ser desarrollada de nuevo o bien adquirirse en el mercado.

Para estimar los costes de la reingeniería, se tienen ciertas ventajas respecto a la misma estimación en proyectos de ingeniería directa: no se debe calcular factores influyentes como el número de líneas de código, sentencias ejecutables, elementos de datos, accesos a archivos, etc., ya que son medidas que se pueden tomar directamente de la aplicación.

Se aconseja utilizar como variables para calcular los costes las que se ofrecen a continuación, y que deben ser debidamente ponderadas en función de su influencia en el coste total:

Número de líneas de código no comentadas.

Coste de los casos de prueba, que se calcula multiplicando el coste medio de cada caso de prueba por el número de éstos, que es función de la complejidad ciclomática del problema.

Número de accesos a archivos, bases de datos y campos. En la ponderación de estas entradas/salidas consideramos la complejidad de las estructuras de información y el grado de independencia de la aplicación respecto de los datos.

Número de operaciones que realizan los usuarios de la aplicación, número de ventanas, número de informes, etc., para el caso de las interfaces de usuario.

8.4.4 Análisis De Costes/Beneficios.

Una vez que se ha calculado el coste de la reingeniería, la última etapa es comparar los costes con los beneficios esperados (no es suficiente con examinar los beneficios que aporte la reingeniería).

El beneficio proporcionado por continuar manteniendo el producto sin reingeniería es el siguiente:

$$BM = [P3 - (P1 + P2)] * P16$$

Deberá retocarse la fórmula cuando los diversos costes varíen de un año para otro.

Si se desarrolla de nuevo el sistema, se obtiene este beneficio:

$$BD = [(P12 - (P10 + P11)) * (P16 - P14) - (P13 * P15)] - BM$$

El beneficio producido por la reingeniería es:

$$BR = [(P6 - (P4 + P5)) * (P16 - P8) - (P7 * P9)] - BM$$

Donde:

P1 = Coste de mantenimiento actual para una aplicación (anual).

P2 = Coste de operación de una aplicación (anual).

P3 = Valor del negocio actual (anual).

P4 = Coste previsto de mantenimiento tras la reingeniería (anual).

P5 = Coste previsto de operaciones tras la reingeniería (anual).

P6 = Valor de negocio previsto tras la reingeniería (anual).

P7 = Coste estimado de la reingeniería.

P8 = Duración estimada de la reingeniería.

P9 = Factor de riesgo de la reingeniería.

P10 = Coste previsto de mantenimiento tras el redesarrollo (anual).

P11 = Coste previsto de operaciones tras el redesarrollo (anual).

P12 = Valor de negocio previsto el nuevo sistema (anual).

P13 = Coste estimado del redesarrollo.

P14 = Duración estimada del redesarrollo.

P15 = Factor de riesgo del redesarrollo.

P16 = Vida esperada del sistema.