

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MEXICO

Centro Universitario UAEM Texcoco



**Revisión y análisis al proceso de mantenimiento de software y
adición de nuevos requerimientos del sistema S336 de AFORE
BANAMEX**

TESINA

Que para obtener el título de:
Licenciado en Informática Administrativa

Presenta:
Armando Ivan Delgadillo De La O

Número de Cuenta
0823343

Generación
2008-2012

Director
M. en C. C. Rivera Villa Carlos Ocelotl

Texcoco, Estado de México, Octubre de 2018

Índice

Índice	2
Introducción	3
1. Software Legado en México	5
1.1 Concepto de software legado.....	5
1.2 Características de software legado.....	9
1.3 Software legado en la actualidad, en México	18
2. Mantenimiento y modificación de software hecho a la medida	22
2.1 Concepto de software hecho a la medida	22
2.2 Objetivos del mantenimiento de software hecho a la medida	39
2.3 Importancia del mantenimiento de software hecho a la medida.....	43
3. Revisión al caso S336.....	46
3.1 Antecedentes	46
3.2 Problemática	51
3.3 Propuesta.....	67
Conclusiones	84
Referencias.....	87

Introducción

La presente tesina tiene como objetivo principal la elaboración de una propuesta de solución a las problemáticas de desarrollo de software que se presentan en una de las instituciones bancarias más importantes de México.

Para lograr este objetivo primariamente es necesario identificar los aspectos más relevantes los cuales son causa principal de malos hábitos al momento de dar mantenimiento a sistemas legados. Para todos los sistemas que califican dentro de esta clasificación es posible observar las malas prácticas del personal encargado de dar mantenimiento por medio del historial de modificaciones que se encuentra registrado dentro de las piezas de software así como en la documentación de referencia que se tiene.

Como elemento principal de análisis para la ejecución de esta revisión se utilizó uno de los sistemas legados que aun brindan soporte en AFORE BANAMEX debido a su gran robustez, capacidad de procesamiento y relevancia que este tiene dentro de la empresa.

Sistemas como este poseen un nivel de complejidad muy elevado, de tal manera que conseguir un entendimiento claro y amplio sobre la manera en cómo opera cada uno de sus módulos es extremadamente complejo, sin embargo es necesario conocerlos muy bien para realizar el mantenimiento que requieren de la manera más óptima teniendo como resultado final una mejor planeación al momento de iniciar algún proyecto de desarrollo de software, proyectos

de optimización, mejoras, atención a fallas e incluso que la aplicación de una reingeniería total sea lo más sencillo posible.

Debido a que estos sistemas fueron diseñados y construidos con tecnologías que se encuentran casi en desuso a la fecha o bien, que son poco conocidas uno de los mayores problemas para esta rama del área de sistemas es el reclutamiento de personal calificado para llevar a cabo este trabajo debido a que el personal con mayor experiencia se encuentra ya en la etapa del retiro y el conocimiento que ellos poseen a la fecha es muy difícil de encontrarlo en la nueva generación de egresados por lo que además de lograr mejores prácticas en los procesos de desarrollo es necesaria una transferencia de conocimiento a las nuevas generaciones así como un resguardo claro y sencillo de toda la información histórica que se logre obtener a fin de que el personal que se incorpore a esta área de sistemas pueda comprender de manera sencilla como llevar a cabo su trabajo.

Se espera que este trabajo sea la base para lograr creación de un modelo que evite que este conocimiento se pierda ya que no solo es suficiente tener un concepto claro de estos sistemas a nivel técnico, sino que también se requiere conocer las reglas de negocio bajo las que operan estos sistemas, adicionalmente esto permitirá que el personal en la actualidad se encarga de dar mantenimiento a estos sistemas este mejor capacitado y el porcentaje de fallas se reduzca.

1. Software Legado en México

1.1 Concepto de software legado

Sommerville (2005) afirma: “Un sistema de software consiste en diversos programas independientes, archivos de configuración que se utilizan para ejecutar estos programas” (p 05). En cierto sentido esto nos brinda una idea de cómo se compone un sistema informático, algunos nuevos otros más antiguos pero todos con basados en la misma definición.

Laudon (1996) menciona: “Todos estos elementos interactúan para procesar los datos (incluidos los procesos manuales y automáticos) y dan lugar a información más elaborada, que se distribuye de la manera más adecuada posible en una determinada organización, en función de sus objetivos”

Los sistemas legados o sistemas heredados, son sistemas informáticos o aplicaciones de software que se caracterizan por estar contruidos con tecnología muy antigua, pero que actualmente sigue siendo utilizada por las organizaciones o empresas en las que están implementados, de tal manera que se han convertido en herramientas de suma importancia para el negocio lo que provoca que el poderlos remplazar sea una tarea bastante complicada y sus actualizaciones, modificaciones o adecuaciones cada vez resulten más costosas.

Es muy común que a la fecha los sistemas legados que aún existen en México se encuentren operando dentro de instituciones bancarias, gubernamentales o transaccionales, básicamente en aquellas empresas en las que el nivel de procesamiento de información es muy

alto y es por esta razón que este tipo de organizaciones son las que han venido haciendo uso de estos sistemas desde casi dos décadas atrás (esto se puede encontrar en el trabajo diario de dar mantenimiento a software heredado con programas fechados en la década de los 70) cuando la tecnología con la que fueron construidos aún era considerada reciente y estos sistemas aun eran simplemente sistemas hechos a la medida.

En las instituciones bancarias que aun operan se puede notar que la mayor parte del software del cual se componen su sistema son piezas fuente que fueron desarrolladas sin tener una especificación técnica debidamente diseñada sino que fueron construidas con base en documentos que establecían los reglamentos constitucionales y políticas bancarias definidas por instituciones ajenas que regularizan a este tipo de negocios, de tal forma que solo se tenía una ligera idea sobre las reglas de negocio con las que debía operar este sistema, así fue como estos sistemas nacieron y con el paso del tiempo se fueron puliendo. Como se menciona anteriormente la manera en que fueron diseñados la mayoría de estos sistemas provoco que al momento de operar por primera vez estos tuviesen muchas fallas, las cuales podían ir desde simples adecuaciones como por ejemplo el cambio de un campo o etiqueta en una pantalla hasta la modificación o corrección de cálculos financieros, por esta razón que tomara tantos años afinar todas estas fallas.

Estos sistemas frecuentemente no estaban bien documentados, lo que llevaba a una tarea de mantenimiento bastante compleja. Fueron construidos de forma rígida, con el objetivo específico de solucionar los requerimientos funcionales; dejando de lado aspectos importantes como la evolución, extensibilidad, portabilidad y compatibilidad entre otros. (Jaimes, 2013, p.01)

Normalmente este tipo de sistemas son considerados herramientas de negocio crítico, ya que el poderlos reemplazar suele ser bastante arriesgado. Un buen ejemplo son los sistemas contables de las instituciones bancarias ya que en estos se manejan políticas y procedimientos organizacionales, en algunos casos de nivel regulatorio de los cuales depende que las instituciones puedan operar o no.

Si el banco fuera a descartar y reemplazar el sistema contable de clientes (el cual es posible que se ejecute en un costoso hardware mainframe), entonces habría un serio riesgo de negocio si el sistema de recambio no funcionara adecuadamente. Además, los procedimientos existentes tendrían que cambiar, y esto puede molestar a los clientes de la organización y causar dificultades con los auditores del banco. (Sommerville, 2005, p. 35)

Regularmente todos los procedimientos que se realizan dentro de una empresa de tipo bancario para el manejo de la información se realizan de manera automatizada a través de un sistema (el cual inicia como un sistema hecho a la medida que con el paso de los años se convierte en un sistema heredado), todos estos procesos son ejecutados y monitoreados con un estricto nivel de seguridad tanto interno como externo ya que en este tipo de negocios se maneja información valiosa la cual requiere ciertos cuidados y niveles de seguridad que garanticen el resguardo de ella, por ejemplo si en un banco se tuviese fuga de información de los saldos o datos personales de todos los clientes podría incluso negársele al banco el permiso para operar.

Los sistemas legados actualmente se encuentran operando constantemente siendo adaptados no solamente a nivel software, si no que hoy en día se ha logrado adecuar estos sistemas con las nuevas tecnologías de software como son lenguajes de programación nuevos, metodologías de desarrollo más actuales y en general a plataformas que permiten la interacción entre sistemas en tiempo real así como a elementos hardware y se logra establecer una conexión entre los diferentes tipos de equipos que permiten el almacenamiento e intercambio de información.

El tiempo y esfuerzo requerido para desarrollar un sistema complejo (sistema heredado) es muy extenso y costoso, lo que provoca que estos comúnmente tengan un tiempo de vida muy largo. Su desarrollo continúa durante toda su vida productiva con distintas adecuaciones que tienen como fin el satisfacer los nuevos requerimientos de negocio y a su vez poder irse adaptando a las nuevas tecnologías operativas. (Sommerville, 2005).

Especificar el tiempo de vida de un sistema heredado es muy complejo. Vega (2011) afirma: “muchos sistemas grandes se pueden llegar a utilizar hasta por más de 20 años” (p.10). En cierto sentido esto confirma lo vitales e importantes que son estos sistemas para sus respectivos negocios que operan aun con este tipo de tecnologías.

Por lo menos en México este tipo de sistemas en su mayoría se encuentran implementados en instituciones en las que detener su producción por unos cuantos minutos puede significar millones de pesos en pérdidas lo que provoca que estos no puedan darse el lujo

de detenerse o consumir demasiado tiempo ejecutando una sola tarea. Lo que en realidad sucede es que estos sistemas se desarrollan pensando en que sus procesos sean lo más eficientemente posible y generen un buen producto con un bajo costo. Por esta razón es muy común que con el fin de agilizar todas sus tareas que la mayoría de los procesos y además los más complejos se ejecuten en modo *BATCH* es decir manera nocturna.

Siempre se ha buscado adaptar nuevas tecnologías y plataformas con todas estas tecnologías arraigadas que permitan procesos ágiles y más rápidos lo cual al 2015 ya es una realidad con variedad de propuestas y proyectos que han podido permitir que estos sistemas interactúen con el software más sofisticado y actual de hoy en día esto permite a los sistemas legados dar un paso más en la innovación e ir prolongando cada vez más su tiempo de vida.

Es necesario hacer cambios rápidos a los sistemas heredados para proveerlos de una interfaz de servicio web. Para tratar estos retos, necesitaremos nuevas herramientas y técnicas, así como formas innovadoras de combinación y uso de métodos de ingeniería del software existentes. (Sommerville, 2005, p.12).

1.2 Características de software legado

En la actualidad es posible ver que todos los sistemas heredados con los que nos encontramos en México tienen como de tres décadas de uso, esto se ha podido notar más a menudo en los sistemas de las instituciones bancarias en donde las fechas de creación de estos programas se remontan a los años 70. Debido a que la mayoría de los procesos en un banco se rigen a través de leyes constitucionales que se encuentran en cambio constantemente es muy común que estos sistemas también hayan sufrido cambios con el paso de los años.

Cuando alguien se da a la tarea de revisar y analizar todos esos programas se encuentra con distintas problemáticas, como por ejemplo, los distintos estilos para la solución problemas que se ven reflejados en el código fuente por parte de los desarrolladores a la hora de programar, o bien, la frecuencia con la que cada programa es modificado ya que aunque no lo parezca a simple vista es posible notar todas estas características nos proporciona la información suficiente para identificar fácilmente qué programas son de los más significativos dentro un sistema de estos y cuáles deben ser tratados con más sutileza a la hora de modificarlos.

Cuando revisamos estos programas uno de los factores principales que podemos determinar es la complejidad de este, en la mayoría de los sistemas se puede ver que todas las piezas cuentan con una bitácora de modificaciones y con una muy breve descripción de cada una de las modificaciones que se han llevado a cabo a lo largo de todo este tiempo, por ejemplo podemos encontrarnos con piezas muy pequeñas que se modifican eventualmente las cuales pueden realizar procesos tan simples como la generación de reportes sin una gran prioridad para el negocio que además se ejecutan dentro del sistema una o dos veces por mes y por otra parte tenemos programas bastante robustos y complejos que se modifican de manera constante, por lo menos una vez al mes y que realizan distintos procesos como pueden ser cálculos matemáticos a gran escala, transferencia de archivos a distintos equipos, lectura y escritura en base de datos, generación de archivos con manejo masivo de información o bien procesos en línea mejor conocidos como pantallas de consulta para el usuario final, adicional a esto se puede ver que este tipo de procesos pueden ejecutarse dentro del sistema más de cien o doscientas veces por día, lo que nos muestra un indicador muy alto en cuanto a su uso y a la importancia que este adquiere dentro del sistema.

Como anteriormente se mencionó, estos sistemas tiene un periodo de vida bastante largo en el cual han sufrido bastantes cambios y se les han adicionado muchísimas más funcionalidades, lo cual ha provocado que estos programas hayan crecido desmedidamente cada vez que se modifican por simple que sea el cambio, es por esta razón que se ha vuelto muy común que algunos de los procesos principales de estos sistemas cuando se encuentran en modificación a la hora de compilarlos o bien en tiempo de ejecución envíen mensajes de advertencia debido a que han excedido el tamaño apropiado para una pieza de software de estas características, aun así, aunque ya se sabe de la advertencia simplemente es ignorada al no encontrar solución a este problema. En algunos de estos programas se recurre al uso de *copy* que ayudan a que sea posible que algunos de estos programas dependiendo la tecnología con la que estén hechos se puedan modificar y sigan creciendo de acuerdo a las necesidades del negocio.

Si se analiza esto desde la perspectiva de un programador se convierte en un problema mayor, como ya sabemos estos sistemas tienen muchos años de vida por esta razón es cien por ciento seguro que distintos desarrolladores hayan intervenido a la hora de modificar estas piezas de software, de tal manera que en ellas se puede ver que cada programador realiza su trabajo de distinta manera, esto se ha podido ver en las piezas al momento de tomarlas cuando requieren ser modificadas.

En los programas actuales se puede ver que algunos programadores tratan de generar el menor número de líneas de código posible, tratando siempre de aprovechar todo lo existente y reutilizar aquellas variables o rutinas que les puedan ser útiles, esto probablemente por varios podría ser una buena técnica a la hora de desarrollar pero por otro lado podría no serlo ya que

para la gente nueva que se integra a las actividades de mantenimiento de este tipo de sistemas seguido le causa confusiones y dudas referentes al diseño de estos sistemas. Por el contrario tenemos a todos aquellos desarrolladores que por miedo a adentrarse a realizar un análisis profundo de estas piezas y averiguar parte de su funcionamiento optan por adicionar sus procedimientos de manera independiente dentro del programa, es decir, en ocasiones se duplican rutinas existentes, se adicionan nuevos e innecesarios archivos y en general se agrega exceso de código fuente el cual la funcionalidad que tiene es probable que ya se pueda encontrar repetida en algunas muchas otras rutinas existentes dentro del programa. La mayoría de las veces esto es provocado por la inexperiencia de los programadores aunque esta práctica suele tener beneficios al momento de estar realizando las pruebas a las piezas modificadas ya que el que la lógica que se adiciona se encuentre aislada a lo que ya se tiene es garantía de que la afectación a la funcionalidad actual disminuya casi al cien por ciento y a su vez con esto se reduce al mínimo que una pieza modificada de esta manera requiera inversión de recursos en pruebas de no impacto.

Todos los procesos y la forma de funcionar de los sistemas legados comúnmente esta enlazada con diferentes sistemas de sus mismas características pero con ciertas y funcionalidades y aplicaciones en específico lo que hace que estos sistemas sean bastante complejos. Si alguna se requiere remplazar un sistema de estos, además de considerar toda la parte del software también se tendría que estar tomando en cuenta la parte física de estos sistemas.

La cantidad de información que procesan los sistemas heredados de los bancos es incalculable, a diario se llevan a cabo distintos tipos de procesos los cuales requieren de un potencial muy alto en cuanto a hardware ya que la mayoría de los procesos que se ejecutan pueden estar corriendo en paralelo, es decir varios procesos pueden estarse ejecutando al mismo tiempo realizando diferentes tipos de tareas y consumiendo un porcentaje muy alto en el uso del procesador, por esta razón es muy importante que el equipo hardware en el que residan cuente con características de procesamiento lo bastante robustas para cumplir los requerimientos que estos sistemas tienen por lo que estas organizaciones tienen que hacer uso de un equipo Mainframe. Hasta la fecha no ha existido ningún otro equipo que pueda superar las capacidades de manejo de datos de un equipo de este tipo.

Los *mainframes*, conocidos coloquialmente como “*big irons*”, son computadoras con 40 años de antigüedad, las cuales se distinguen por las grandes dimensiones, ya que están orientadas al procesamiento de grandes volúmenes de datos, estas computadoras son usadas principalmente por empresas grandes para aplicaciones críticas, procesamiento de grandes volúmenes como censos, estadísticas de la industria y de consumo, aplicaciones bancarias, tributarias, compañías telefónicas o de telecomunicaciones y mercado de valores, aerolíneas y tráfico aéreo, en general como de centro principal de procesamiento de grandes empresas con un volumen de facturación elevado. La capacidad de un mainframe se define en términos de los siguientes factores: la velocidad de su CPU, su memoria interna, su capacidad de almacenamiento externo, sus resultados en los dispositivos E/S rápidos y considerables, la calidad de su ingeniería interna que tiene como consecuencia una alta fiabilidad y soporte técnico caro pero de alta calidad. Es común que los mainframes soporten miles de usuarios. (Vega, 2011, p.12).

En México las marcas que predominan el mercado son los equipos IBM que se encuentra en instituciones como BBVA Bancomer, Santander Serfin, HSBC, Banco Azteca entre otros y BANAMEX a partir de 1990 adquiere equipos UNISYS que tiempo después soportaría entre otros sistemas al S336 de AFORE BANAMEX (Morillon, 2003, p.06).

Pero no es suficiente tener un nivel de procesamiento de datos poderoso, también se requiere una capacidad de almacenamiento de datos lo bastante robusta y que sea capaz de adaptarse a distintas tecnologías. Estos sistemas ejecutan cientos de procesos diariamente por esta razón es que deben ser capaces de almacenar todos los datos que reciben y mantener un registro histórico de todos los que envían, pero adicionalmente deben contar con ciertos niveles de seguridad para garantizar que la información histórica con que la cuenta sea íntegra.

La información procesada por estos sistemas comúnmente es conservada en distintos tipos de formatos como pueden ser archivos planos, bases de datos, reportes, listados, entre otros. Así, la información de los sistemas legados se recolecta, se depura e integra a fin de ser almacenada (Gutiérrez, 2001). Esto provoca que en algunos casos pueda existir infinidad de información duplicada e innecesaria, esto es debido a que este tipo de sistemas no fueron construidos sobre una metodología en específico, lo que provoca que al no conocer procesos para el manejo de la información se genere información basura.

Haciendo referencia al tipo de información que almacenan estos sistemas por la naturaleza propia de estos así como del giro en el que están implementados los datos que

procesan son considerados información sensible, la cual debe ser manejada con altos niveles de seguridad tanto internos como externos para toda aquella persona relacionada con el sistema de información (Bertolín, 2008).

Un claro ejemplo de la seguridad interna se puede ver actualmente en los sistemas legados de instituciones bancarias es cuando un recurso interno (denominado usuario o aquellos que hacen uso del sistema) del banco requiere hacer uso de este, es necesario que tenga clave de acceso para que pueda ingresar al sistema y a su vez tenga acceso a toda la información que contiene, estas claves de acceso se asigna dependiendo el puesto funcional en el que se encuentre y siempre con la validación de un superior esto permite evitar que se haga un mal uso de estos sistemas, así como de la información que contienen y se tenga un control de quien, puede o no, tener acceso a esta herramienta fundamental y critica para el negocio. Poco o nulo es el personal externo que puede tener acceso a todos los niveles de información del sistema debido a que casi en todos los bancos la información de estos se encuentra con niveles muy altos de seguridad.

No solamente se requieren tener medidas internas como claves de acceso para el personal interno del negocio, sino que, como ya se mencionó con anterioridad es muy común que casi todos estos sistemas interactúen con otros sistemas externos, lo cual no es la parte crítica, sino más bien nos referimos al cuidado que se debe tener del camino que los datos recorren del lugar de donde origen y hasta el lugar destino, y garantizar principalmente que no exista fuga de información y datos confidenciales que puedan llegar a manos equivocadas, lo cual podría ocasionar un problema muy grave para el negocio.

Se ha podido ver en los sistemas bancarios de hoy en día la implementación de software para seguridad muy sofisticado que va del uso de antivirus en los equipos PC que actualmente se utilizan, hasta aplicaciones de autenticación interconectadas remotamente con servidores ubicados en el extranjero los cuales se encuentran monitoreando los equipos las 24 horas del día los 365 días del año. Pero eso no es suficiente ya que también se debe asegurar que la transmisión y recepción de información se realice de la manera segura y confiable tanto para el negocio como para los clientes de este, es por eso que son implementados elementos como *routers*, *firewalls* y software para la cifrado de datos como un complemento funcional de estos sistemas y así puedan mantener el nivel de seguridad requerido por los estándares establecidos de cada negocio.

Es muy común dentro de la documentación que se ha podido ver en estos sistemas no existan especificaciones o documentos que describan la funcionalidad de cada uno de los módulos que componen dicho sistema, o si existen comúnmente no son los adecuados, solo en algunos programas que son los que se utilizan y modifican con mayor frecuencia llega a existir algún documento bastante básico que comúnmente solo da una leve idea de la funcionalidad que actualmente se le está adicionando a una pieza de software en específico, pero no proporciona un conocimiento amplio de todo lo que cada uno de estos pequeños programas realiza, es disminuye la posibilidad de que puedan conocer de manera más detallada el funcionamiento a detalle de todos estos sistemas. En la mayoría de este tipo de sistemas los analistas y desarrolladores con más años de experiencia son los que han logrado tener un conocimiento lo suficientemente apto para identificar detalles más específicos de sobre cómo operan el sistema, la mayoría de estos personajes son aquellos que colaboraron en la definición y diseño de estos sistemas casi desde

sus nacimiento. De tal manera que para los ingenieros encargados del mantenimiento de un sistema de este tipo que comienzan involucrarse en esta área es probable que esto dificulte al realizar su trabajo y pueda ser un poco más complejo adquirir cierto nivel de conocimiento detallado sobre los alcances y capacidades del sistema.

Las reglas de negocio establecidas dentro de un sistema legado en un banco comúnmente se definen día a día, estas no son publicadas de manera formal ni documentadas sino que son creadas con base en leyes constitucionales y con base en las políticas de trabajo de cada institución las cuales se encuentran constantemente siendo modificadas sin previo aviso y no existe algún documento que permita tener un historial de la parte del negocio que cambio de tal manera que estos sistemas deben irse adaptando a la par de cómo cambian estas reglas, a su vez esto dificulta el tener sistemas simples, fáciles de entender y sobre todo organizados.

Uno de los problemas más frecuentes para estos es que ya no existe gente que conozca el tipo de tecnología con la que fueron desarrollados, como son los lenguajes de programación, equipo en el que residen y sobre todo la metodología no formal bajo la que nacieron y fueron tomando forma, esto provoca que los proyectos de modificación de software legado lleven más tiempo de lo normal ya que se cuentan con menos manos para realizar todo ese trabajo, adicional a esto se sabe que ya desde hace muchos años se viene diciendo por algunos expertos en tecnologías de la información que este tipo de tecnologías van a desaparecer lo que ha provocado que en las instituciones de educación se haya eliminado del plan de estudios actual y este haya

sido remplazado por lenguajes de programación y tecnologías actuales como lo es Java o .NET o lenguajes para el desarrollo de aplicaciones móviles.

Esto provoca que el para una institución el mantener este tipo de sistemas se bastante costos y cada vez menos rentable, el que no exista personal capacitado en este tipo de tecnologías tiene como consecuencia que las instituciones globales establecidas en otros países pero que sus sistemas residen en México busquen migrar todos sus sistemas y aplicaciones a países del extranjero lo cual suele tener mayores beneficios para la empresa, como puede ser una disminución en costos de mantenimiento, reducción de personal, centralización de sus procesos y en general beneficios económicos que permiten a la empresa reducir sus gastos al máximo.

1.3 Software legado en la actualidad, en México

Es muy común que los grandes sistemas heredados que se encuentran en México, no sean los mismos que originalmente se empezaron a utilizar. Vega (2011) afirma: “Muchos factores externos e internos, como el estado de las economías nacional internacional, los mercados cambiantes, los cambios en las leyes, los cambios de administración o la reorganización estructural, conducen a que los negocios experimenten cambios continuos” (p 10). De tal manera que estos sistemas se ven en la necesidad de ir incorporando actualizaciones durante toda su vida útil, estas adecuaciones regularmente se hacen por varias personas debido a que este tipo de negocios requieren de servicio de *outsourcing* para realizar las adecuaciones requeridas a lo largo de todos estos años.

El hecho de que distintos sistemas legados sigan creciendo sin metodologías de desarrollo establecidas de manera formal es muy normal dentro de los procesos de desarrollo de software que se llevan a cabo hoy en día dentro de estas instituciones, como se menciona con anterioridad estos sistemas nacieron sin el uso de un modelado formal de software y es por esta razón que difícilmente podrían adaptarse a uno, en realidad lo que se pretende realizar por el departamento de sistemas de estas empresas es dar una vida más larga a estos sistemas.

Muchos de los sistemas heredados con los que actualmente son utilizados en las organizaciones en México, afirma Vega (2011) “no fueron diseñados ni construidos con estricto apego a una metodología, esta es una característica inherente a sistemas con 10 o más años de antigüedad, donde la ingeniería de software y metodología estaban en etapas de crecimiento e implementación” (p.14).

Como consecuencia de lo de lo que se menciona en el párrafo anterior los sistemas han ido creciendo de manera descontrolada y desorganizada lo que ocasiona que el lograr un entendimiento de estos sistemas sea bastante complicado, este tipo de contrariedades son las más comunes a las que se enfrentan los ingenieros que se incorporan a las empresas de *outsourcig* encargadas de dar mantenimiento a estos sistemas, esto tiene como resultado que exista cada vez menos personal que tenga el conocimiento de estas tecnologías. Aunque otra causa se debe a la falta de escuelas que manejen en su plan de estudios tecnologías como esta, que aunque para muchos parezcan arraigadas en realidad son las herramientas que se utilizan de hoy en día.

La mayoría de las organizaciones se encuentran tratando de migrar sus sistemas a países del extranjero en donde es mucho más fácil encontrar al personal apropiado para realizar los trabajos de mantenimiento a sus sistemas y sobre todo mucho más barata que en México, esto disminuye la posibilidad de que más gente se pueda incorporar a esta área de las tecnologías de información y se pierdan fuentes de empleo tan valiosas y necesarias en este país.

Actualmente en México las organizaciones que operan con sistemas legados o heredados se encuentra implementando nuevas tecnologías y lenguajes de nueva generación siempre buscando lograr una interacción entre sus sistemas a través de distintos medios de comunicación como por ejemplo el intercambio de archivos con información entre sí o la conectividad entre plataformas *web* a través de *WebService*, mediante las que se ha podido comprobar que aunque antiguos, los equipos *mainframe* tienen un nivel muy alto de procesamiento de datos quizás a la par de los sistemas de última generación que implementan una gran cantidad de innovaciones como interacción con aplicaciones móviles las cuales permiten agilizar y hacer más rápidos los procesos actuales, de alguna manera esto abre las puertas y logra que estos sistemas puedan adaptarse e ir avanzado de la misma forma que todas estas nuevas de tecnologías de la información.

Una de las posibles alternativas para garantizar la sobrevivencia de estos sistemas es la implementación de la arquitectura SOA o lo que su siglas significan “arquitectura orientada a servicios” esto consta de publicar las aplicaciones de estos sistemas como servicios los cuales

constan de llamadas o peticiones de a través de un proveedor y un consumidor en donde las peticiones pueden ser solicitudes de información en donde se requiere de una respuesta.

No solamente la parte software de estos sistemas cuenta con innovaciones a la fecha, también se encuentran remplazando algunos de sus equipos actuales los cuales incorporan características como el uso de arquitecturas x86 la cual vuelve aún más rápidos estos equipos, adicionalmente se tiene un notable aumento en las capacidades de almacenamiento de información lo que da como resultado niveles de procesamiento de datos más altos a los de sus antecesores.

La plataforma mainframe está presente en organizaciones grandes (empresas de telecomunicaciones, gobierno, bancos, etcétera) con departamentos de informática con tamaño del orden de miles de empleados y la misma se caracteriza por la presencia dominante de sistemas informáticos heredados en las funciones principales del núcleo organizacional. (Vega, 2011, p. 02).

El que en estos sistemas se busque actualizar sus plataformas de alguna manera garantiza que el uso de los mismos se asegure por una o dos décadas mas siempre y cuando no sean migrados a otros países hay garantía de que se generen nuevas fuentes de empleo y nuevas oportunidades para la gente joven que trabaja con sistemas legados.

Por suerte para algunos la adecuación en estos sistemas no se detiene, siguen siendo actualizados en cada empresa en la que se siguen funcionando bajo las metodologías y procedimientos establecidos los cuales cada vez se globalizan más tratando de adaptarse a nuevas formas de trabajo que intentan volver más ágiles los procesos actuales, esto en ocasiones dificulta que el personal que interviene en estos procesos logre comprender en el tiempo esperado como hacer su trabajo a pesar de que son las mismas tareas y el mismo producto a generar debe realizarse de distinta manera.

Los niveles de conocimiento para estos actores cada vez son más exigentes, requieren de personal altamente capacitado en diferentes áreas de trabajo no solo en una. Para lograr esto se requiere de una inversión de horas hombre más alta de lo normal, a lo cual se le conoce como curva de aprendizaje, entre más larga sea esta más alto es el tiempo que demorara un persona en adquirir este aprendizaje, en otros términos esto se traduce para cualquier compañía en altos costos de inversión para capacitar a sus empleados, lo cual casi nunca sucede por eso depende de cada profesionista el prepararse por su cuenta sin depender de su empresa con el fin de irse actualizando ya que esta es una de las áreas con más demanda y un con un avance tecnológico muy rápido.

2. Mantenimiento y modificación de software hecho a la medida

2.1 Concepto de software hecho a la medida

El software hecho a la medida es aquel que se diseña y construye de acuerdo a las especificaciones definidas por el negocio o cliente en base a una necesidad, por esta razón este tipo de software es el que más se asemeja a lo que exactamente el cliente desea, pero aunque este software se considera personalizado no siempre cumplen su totalidad con lo que el cliente desea debido a que no siempre se puede conseguir la funcionalidad que se requiere sin tener que involucrar algunos otros componentes que van desde algunos otros lenguajes de programación o incluso algunos otros subsistemas construidos en plataformas totalmente distintas pero que proporcionan cierta funcionalidad la cual un solo sistema hecho a la medida no puede ofrecer.

El software a la medida se adapta a las necesidades del cliente y, por tanto, tiene características únicas. Sin embargo, las diferencias en la práctica no resultan tan nítidas ya que, por una parte, los desarrollos a la medida constituyen un producto terminado y, por otra, los productos empaquetados necesitan personalización, actualización y mantenimiento, lo cual representa oportunidades para empresas que no necesariamente son las que desarrollan. (Hualde & Gomis, 2007, p. 198)

En la actualidad existen diferentes tipos de software los cuales son utilizados en distintas empresas como herramienta principal para la ejecución de sus tareas de manera eficiente entre los cuales destacan algunos programas de uso comercial los cuales en algunos casos no cubren en su totalidad las necesidades de procesamiento de datos que se requieren en algunos tipos de negocios. Picazo (2012) menciona: “Por ejemplo, tiendas comerciales que tienen numerosas operaciones con muchos artículos deben tener un programa que les permita controlar todo en un tiempo real, maximizando todos sus recursos humanos, materiales, financieros, etc.” (p.15).

Las grandes empresas transnacionales que hacen uso de sistemas comerciales, por cuestiones de pago de licencias así como algunos otros impuestos de manera usual se ven en la necesidad de invertir porcentajes considerables en procesos de mantenimiento y actualización de sus sistemas los cuales deben ser capaces de cubrir las necesidades requeridas. Estos sistemas al ser de código patentado y poco modificables cuando son liberados no pueden sufrir ajustes por parte de la empresa que hace uso de ellos, sino que deben esperar hasta el momento en que la empresa dueña de este software aplique las correcciones y actualizaciones pertinentes, esto en la mayoría de los casos mejora el funcionamiento de estos sistemas pero no cubre en su totalidad las necesidades de funcionalidad requeridas, de tal modo que algunos negocios tienen serias dificultades para llevar a cabo de manera correcta su procesos.

A diferencia de los sistemas con pago de licencia conocidos como *shareware*, los sistemas hechos a la medida suelen ofrecer más posibilidades de configuración y adaptación a diferentes tipos de negocio, dependiendo el tipo organización en la que se encuentren implementados pueden o no considerarse como sistemas terminados debido a que existen negocios los cuales se encuentran en constante cambio y por ende estos sistemas requieren de actualizaciones, adecuaciones y mantenimiento casi durante su periodo total de vida para poderlos operar.

En proyectos de mantenimiento de software hecho a la medida la inversión monetaria casi siempre suele ser la necesaria debido a que los requerimientos que se tienen estén correctamente definidos, esto garantiza que el impacto al sistema será el adecuado y el requerido

por parte del usuario final, esto sobre todo reducen al mínimo el exceso de gasto en la compra de los paquetes de actualización que se requieren en un *shareware* los cuales la mayoría de las veces son bastante costosos. Como ejemplo cuando se requiere actualizar un sistema con pago de licencia (*shareware*) es muy común que se deba adquirir todo el paquete de actualizaciones que el proveedor ofrece sin importar si son o no requeridos por el negocio ya que de no implementarlos en el sistema implica riesgos de compatibilidad con otras aplicaciones internas o externas de tal forma que no se tiene ninguna otra alternativa por parte de las empresas y se ven en la necesidad de realizar una inversión en actualizaciones de software que no en todos los casos son de mucha utilidad.

Características de software hecho a la medida:

Cuando se adquiere un *shareware* por determinada empresa las actualizaciones que se aplican a este se hacen a través de la compra de paquetes, los cuales únicamente son desarrollados por parte de la empresa dueña dicho software, esto casi siempre es una gran ventaja para algunos negocios que hacen uso de sistemas con pago de licencia ya que con esto evitan la necesidad de invertir grandes cantidades en procesos relacionados con el desarrollo de software. Lo mismo sucede cuando llega a existir un error en el software la marca dueña de éste tiene la obligación de corregirlo siendo transparente para el cliente cualquier modificación al software.

Las desventajas para las empresas que hacen uso de sistemas con pago de licencia o *shareware* al momento de requerir algún tipo de ajuste en sus sistemas son mayores ya que para

aplicar estos cambios es necesario adquirir los productos que la marca propietaria del software ofrece. Algunos de estos ni si quiera son necesarios para el negocio, pero con el fin de obtener más ganancias estas compañías ofrecen grandes paquetes de software que incluyen algunos componentes innecesarios los cuales se deben adquirir de manera casi obligatoria ya que de no hacerlo la actualización no es aplicada.

Pero también existen diversas ventajas al usar sistemas de licencia, entre las que destaca la aplicación de actualizaciones, las cuales comúnmente suelen ser más baratas e incluso en algunos casos gratuitas ya que se considera como soporte al sistema que algunos software de licencia ofrecen sin costo adicional para el usuario.

Es muy común que en cada que son liberadas al mercado nuevas actualizaciones en ellas se adicione complementos que proporcionan posibilidad de tener compatibilidad con algunos otros sistemas lo cual obliga al negocio a actualizando sus herramientas de software de manera casi obligatoria ya que de no hacerlo podría provocar fallas en sus sistemas. Otro factor importante el cual provoca que el negocio dependa de los proveedores de software es cuando se requiere de soporte técnico se da por ejemplo cuando la compañía dueña del software decide discontinuarlo y reemplazarlo por otro muy similar pero con algunas mejoras. Para los negocios que utilizaban el software reemplazado surgiría la necesidad de migrar sus sistemas a la nueva versión, esto debido a que el soporte técnico brindado a los aplicativos discontinuados no suele durar por más de 5 años.

Para brindar mantenimiento a un sistema hecho a la medida se requiere de los servicios de empresas que ofrecen servicio de *outsourcing* las cuales en la mayor parte de los casos son las dueñas de estos sistemas, por esta razón cuentan con el personal que tiene los conocimientos necesarios para realizar estas adecuaciones. Conforme va pasando el tiempo es muy común que estos sistemas pasen a ser sistemas legados debido a que hasta la fecha siguen siendo utilizados y actualizados, por lo que algunos de estos se han pasado ser considerados sistemas de nivel críticos para algunos negocios.

En las instituciones bancarias una de los principales herramientas para llevar a cabo sus procesos son sus sistemas de información, por esta razón las instituciones manejan sus procesos de modificación de software de manera muy estricta haciendo uso de metodologías para proyectos de modificación de software muy sofisticadas las cuales incluyen certificaciones sobre distintos estándares utilizados a nivel mundial.

Para llevar a cabo un proceso de modificación de software se necesita de un requerimiento por parte del negocio o bien del usuario final (quien hace uso del sistema), quien es la persona que solicita se adicione una nueva funcionalidad a lo que se encuentra funcionando actualmente, este requerimiento debe estar muy bien fundamentado, sustentado y aprobado por la gente responsable de autorizar proyectos así como el presupuesto necesario para la ejecución de este. Una vez que se cuenta con los requisitos es necesario formar un equipo de trabajo que se compone de expertos en diferentes áreas, uno de los principales actores en este proceso son los analistas de negocio así como el personal encargado de la gestión administrativa del proyecto. Es al inicio del proyecto en donde se consume el mayor tiempo de este, ya que se determina el

impacto que se tendrá en el sistema y se realiza una estimación sobre la duración de todo el proyecto incluyendo todos los procesos necesarios para su liberación a un ambiente productivo, adicional a esto se debe realizar los cálculos de capital e insumos que se requieren para cumplir para que el ciclo de vida de proyecto camine de acuerdo al camino trazado. Entre los elementos que son requeridos para la ejecución de un proyecto de modificación de software se encuentran el personal de desarrollo, personal de pruebas, analistas de negocio, integradores tecnológicos, lugares físicos de trabajo, equipos de cómputo, claves de acceso a todas las herramientas que se utilizan los cuales pueden ser requeridos o no conforme va avanzando el proyecto.

En la mayoría de los casos en los cuales una empresa hace uso de un *shareware* este casi siempre cubre la mayor parte de las necesidades que la empresa tiene, pero algunas veces suele suceder que se requiera de algún ajuste a estos programas, por concepto de derechos de autor y aunque sea muy necesario el cambio no es del todo legal que la empresa que lo utiliza realice alguna modificación debido a que todos estos programas se encuentran patentados. Los sistemas comerciales suelen ser muy cuadrados y pocas veces se puede modificar o adaptar a lo que el negocio requiere lo que se convierte en un problema mayor para los tipos de negocio que dependen en su totalidad de un sistema de información ya que se ven en la necesidad de hacer uso de otro software adicional que cubra las nuevas necesidades o bien recurrir a uno hecho a la medida, lo cual en ambos casos suele generar gastos que podrían haberse evitado de haber hecho uso de un software a la medida al inicio de sus operaciones.

Por ejemplo, si tenemos un sistema con pago de licencia para la administración de clientes de un determinado negocio que realiza distintas funciones como la impresión de

comprobantes, consulta en pantallas de la información de los registros dados de alta, generación de reportes para el negocio y demás, suele ocurrir que con las actualizaciones que se van liberando durante su uso adquiriera elementos que sean muy útiles pero que cualquier otro negocio pueda adquirir lo cual no le adiciona una ventaja competitiva al negocio. Por ejemplo, si a este mismo software se le quisiera adicionar una mejora a la funcionalidad con la que ya cuenta, el negocio dependería de la marca encargada de las actualizaciones para obtener esta mejora siempre y cuando los nuevos paquetes de mejoras consideren todos estos cambios que el usuario de este software requiere, esto casi nunca pasa por lo que el negocio se ve en la necesidad de tener que adquirir otro aplicativo.

Una gran ventaja del software hecho a la medida bien hecho es la flexibilidad y nivel de modelado que este tiene, aunque puede ser más costoso que un software comercial este tipo de programas permite que con el paso del tiempo se pueda ir adaptando de acuerdo a las necesidades que el negocio tiene y lo más importante conforme este lo va requiriendo sin tener ni más ni menos funcionalidad de la que se necesita. Para el negocio resulta más sencillo modificar su software, si éste es hecho a la medida debido a que casi siempre el negocio suele ser dueño del sistema, esto ofrece la posibilidad de modificarlo dentro de sus áreas de desarrollo de software (si es que tiene) o bien contratar empresas externas que ofrezcan servicio de *outsourcing* para realizar el trabajo teniendo como resultado la adición de solo la funcionalidad necesaria, nada extra que pueda consumir recursos innecesarios en sus equipos o la generación información basura que entorpezca la funcionalidad con la que ya se cuenta, es decir, beneficios que aporten mejoras notables con las cuales la competencia no cuente permitiendo al negocio obtener ventajas que el resto negocio no tengan.

Otro factor importante cuando hablamos de software hecho a la medida es la capacidad de conectividad con otros sistemas cuando un mismo negocio cuenta con varias herramientas de software que trabajan en diferentes plataformas ya que al ser modificables es posible adaptarlos y lograr así una interacción entre estos. Algunas veces cuando se requiere comunicar sistemas a la medida con algunos otros sistemas externos incluso de diferente tipo casi siempre es más sencillo realizarlo debido a que cualquier componente de este puede ser modificado y configurado según las necesidades de comunicación que se requieran.

Cuando una empresa adquiere un software hecho a la medida conoce de manera exacta las capacidades con la que este cuenta así como el producto que debe generar y las condiciones bajo las que debe realizarlo, algunos de los módulos de estos sistemas pueden pasar muchos meses o incluso años sin requerir ninguna modificación, aunque estos contienen fallas o errores que hasta la fecha no se hayan detectado. La mayoría de estos sistemas requieren de adecuaciones después de haber sido liberados y entregados al cliente, esto no siempre es provocado por una falla técnica si no que en ocasiones se requieren de algunos ajustes a fin de poderlos adaptar a las características de los equipos en los que estarán siendo ejecutados, otra causa muy común es una falla en el diseño que casi en todos los casos está relacionada a una mala definición por parte del cliente o falta de entendimiento de lo que se requería (finalmente a un conocimiento insuficiente de los procesos y la problemática a resolver). Por ejemplo si se tiene un sistema el cual imprime algunos recibos o comprobantes con un formato establecido el cual no es el que el cliente definió puede estar operando pero aun así necesita corregir estos errores o deficiencias con las que cuenta tiempo después de haber sido desarrollado.

Como se menciona anteriormente realizar una modificación o adecuación se requiere de la creación de proyectos de software bastante robustos los cuales van corrigiendo los distintos fallos del sistema con el paso de los años, algunas veces cuando se libera nueva funcionalidad suele ocurrir que algunas de las modificaciones aplicadas sin querer afecten otros módulos del sistema por lo que se deban realizar correcciones de manera constante y a lo largo de vida del sistema. Las modificaciones de los sistemas hechos a la medida son parte de los procesos diarios de la organización y aunque no siempre se afectan los mismos módulos constantemente se pueden ver cambios nuevos en estos sistemas.

Cuando se adquiere un software hecho a la medida normalmente este no recibe actualizaciones ni soporte por parte de la empresa que lo creó debido a que los alcances que se definen cuando se inicia el desarrollo de un sistema de estos no incluyen actualizaciones ni correcciones a posibles errores o si los incluye únicamente contemplan la parte funcional del sistema, de tal manera que la única solución para que estos sistemas se puedan ir actualizando es mediante el departamento de sistemas de cada institución si es que lo tiene de lo contrario esto debe realizarse a través de empresas que ofrezcan los servicios de *outsourcing*, donde constantemente son requeridos estos ajustes a sus sistemas.

Situaciones de este tipo se dan más frecuentemente en instituciones bancarias o de gobierno debido a que estas instituciones funcionan en base a lo establecido en algunas leyes gubernamentales por esta razón cualquier cambio por mínimo que sea debe verse reflejado en sus sistemas o en todo aquello que utilizan para llevar a cabo las actividades diarias de negocio ya

que de no hacerlo estas organizaciones podrían ser sancionadas con multas millonarias por no cumplir con los lineamientos establecidos.

Por ejemplo en un software comercial es muy común que una vez que se adquiere se adicionen algunos complementos o se liberen de manera gratuita y automática para todos los usuarios paquetes para la corrección de pequeños fallos que podría tener la herramienta, esto suele ser transparente para el usuario ya que todos estos complementos y actualizaciones se aplican de manera automática y sin costo adicional para el cliente (dada las tecnologías de redes y desarrollo existentes en la actualidad) ya que se encuentran dentro del soporte y actualizaciones de software que este tipo de sistemas incluye desde el momento de su adquisición.

Por otro lado para que un sistema hecho a la medida pueda recibir todas estas actualizaciones o mejoras es necesario mandarlas a hacer a la medida ya sea con la empresa que desarrollo el sistema que en la mayoría de los casos es quien conoce el sistema a detalle o como se mencionó anteriormente contratar algún *outsourcing* para que se apliquen estas correcciones o actualizaciones.

Los sistemas hechos a la medida nacen siendo pequeños programas que realizan actividades muy sencillas por esta razón las piezas de código fuente que componen estos programas en sus inicios suelen ser muy pequeñas. Con el paso de los años y conforme lo va requiriendo el negocio estas piezas comienzan a tener más y más cambios debido a que cada actualización que se les es aplicada afecta de una u otra manera estas piezas, ya sea que se les adicione nuevo código fuente a esas mismas piezas o se vayan generando nuevas, de cualquier

manera estos sistemas comienzan a crecer cada vez más conforme va pasando el tiempo hasta convertirse en unos verdaderos gigantes con un nivel de complejidad más alto cada que son modificados lo que dificulta en gran medida al momento de realizar un análisis para tratar de comprender la manera en que estos operan y cualquier modificación pueda ser más sencilla.

Toda la funcionalidad y tamaño que estos sistemas van adquiriendo con el paso de los años los convierte en herramientas de negocio críticas, por esta razón el poderlos remplazar por alguno otro sistema suele ser bastante complicado y de alto riesgo para el negocio debido a que cualquier error o falla a la hora de modificarlos puede resultar bastante costosa para la empresa sin mencionar el impacto a nivel negocio que esto podría provocar como puede ser el que sus operaciones diarias queden detenidas.

La tecnología con la que fueron construidos la mayoría de los sistemas legados para muchos parece ser obsoleta, sin embargo es muy robusta y poderosa con un potencial muy alto para procesar miles de transacciones en muy poco tiempo es por eso que para algunos tipos de negocios los cuales manejan un nivel muy elevado de transacciones por día este tipo de tecnologías suelen ser las más apropiadas.

Por ejemplo, para las instituciones de tipo bancario y de gobierno sus sistemas legados llevan varias décadas brindando un excelente y eficaz servicio. La toma de decisiones al momento de diseñar todos estos sistemas estuvo fundamentada con base al poco rendimiento y potencia que brindaban en aquellos tiempos los equipos cliente-servidor para almacenar volúmenes de información tan altos como los que se requerían, pero sobre todo la poca

capacidad para procesar de forma rápida y exacta toda la información con la que se estaría operando.

De esta forma y como una opción más rentable aunque un poco más costosa se optó por mantener todos sus sistemas de información en equipos *Mainframe*.

Stephens (2008) menciona: “Mainframe, es el término que se les daba a las computadoras entre la década de los 70’s a 80’ debido a que en esta época los equipos existentes era demasiado grandes” (p.11).

En aquellos ayeres estos equipos eran el único tipo de hardware que era capaz de proveer un nivel superior de procesamiento, almacenaje superior al resto de las opciones que existían en aquel entonces.

Aunque en la actualidad existen equipos más sofisticados y modernos en los cuales podrían residir estos sistemas la mayoría de las empresas prefieren mantener los equipos *Mainframe* con los que cuentan y siguen utilizando e ir adquiriendo equipos de la misma familia pero con tecnología más avanzada, debido a que las empresas que desarrollan estos equipos se encuentran mejorando las características sus modelos, una de las principales mejoras realizadas es la implementación de arquitecturas x86 fiel representante de RISC *reduced instruction set computer* (Computadoras con un conjunto de instrucciones reducido), dejando atrás a su rival CISC *complex instruction set computer* (Computadoras con un conjunto de instrucciones complejo) que fue una de las arquitecturas más sofisticadas desarrollada para equipos *Mainframe*

así como el incremento en la capacidad de almacenamiento con la que estos equipos cuentan a través de actualizaciones y adaptaciones con nuevas tecnologías.

Algunos de los proveedores más importantes en México de equipos *Mainframe* quienes abarcan la mayoría del mercado de negocios con sistemas legados son IBM con equipos como el IBM S/390 y UNISYS con el LIBRA 680, estas marcas se encuentran brindando mejoras e innovaciones a sus equipos con la finalidad de que sus equipos sigan siendo los más utilizados como hasta ahora se ha venido haciendo principalmente por las grandes empresas del sector financiero.

Como en la mayoría de los casos cuando alguien adquiere un equipo de cómputo nuevo ya sea una laptop, una PC de escritorio o incluso un servidor con determinado proveedor este ofrece cierta garantía o soporte técnico según sea el caso. Por ejemplo podemos mencionar el caso de los equipos de uso personal, los cuales al momento de su adquisición cuentan con un determinado tiempo de garantía por si algún componente ya sea hardware o software llega a fallar, entonces el proveedor de dicho equipo se vería en la necesidad de reparar o reemplazar el equipo si fuese necesario. Pero no en todos los casos la necesidad del usuario suele ser la misma, por ejemplo si fuese el caso en el que el usuario quisiera aumentar la capacidad en disco duro esto sería una mejora al equipo pero que es poco frecuente encontrarla dentro del soporte técnico que ofrecen estas compañías de tal forma que el usuario debe buscar alguna otra alternativa para realizar la actualización necesaria a su equipo.

Lo mismo sucede con estos sistemas ya que a pesar de que incluyen cierto nivel y tiempo de soporte este no siempre satisface las necesidades del cliente, en estas situaciones cuando se ven en la necesidad de solicitar los servicios de una empresa externa o en algunos casos la misma pero con costos extras a la adquisición de una mejora al sistema.

Una de las características principales de estos sistemas es que se encuentran implementados en negocios en los que las modificaciones a los procesos que ejecutan a diario son constantes, por esta razón estos sistemas tiene la necesidad de irse adaptando a las nuevas reglas tanto de negocio como tecnológicas. Por ejemplo hablemos de las instituciones de banca pública o privada, para que estos negocios puedan operar deben ser regulados por leyes u organizaciones gubernamentales las cuales son las que establecen bajo que términos deben operar, de tal manera que si estas organizaciones realizan cambios en cuando a políticas o alguna ley relacionada al sector financiero es modificada, estos cambios deberán verse reflejados en el funcionamiento del sistema a fin de que se logre operar de acuerdo a lo establecido por la ley.

Existen dos tipos de requerimientos para llevar a cabo una modificación a determinado sistemas de información, una de estas considerada como prioritaria en todo el sector financiero es el requerimiento regulatorio, que son requisitos legales para que la institución pueda llevar a cabo sus procesos de negocio, estos deben aplicarse en tiempo y forma establecido de acuerdo a la ley, de tal forma que todas las instituciones deben tener implementadas estos requerimientos en sus sistemas sin excepción, ya que de no cumplir con estos requisitos provocaría que la empresa se enfrente a serios problemas legales que podrían tener como resultado la suspensión de sus actividades de manera definitiva. Por otro lado existen las adecuaciones no regulatorias,

las cuales son aquellas que sólo adicionan mejoras al negocio propio y pueden no tenerlas implementadas el resto de las empresas, lo cual depende de cada una de acuerdo a su estrategia de negocio por lo que son libres de aplicarlas o no a la funcionalidad de sus sistemas.

En la actualidad los sistemas legados se encuentran en un proceso de migración a nuevas plataformas este proceso consiste en lograr una integración con sistemas de plataformas cliente-servidor y lenguajes más modernos de manera que esto permita que se logre una interacción entre ambos y se logre tener sistemas con lo más nuevo en cuanto a tecnologías de información sin perder la capacidad de procesamiento y estabilidad en la ejecución de sus procesos con la que ya cuentan los antiguos sistemas legados.

La idea primordial de esta migración es no perder la funcionalidad que ya se tiene en los diferentes sistemas y que difícilmente se podría duplicar en algún otro, pero que a su vez se implementen tecnologías innovadoras como los *Web Services* (Servicios Web). ¿Cómo?, Muy sencillo, la idea principal es seguir utilizando todos los procesos *BATCH* que se ejecutan actualmente en un sistema legado los cuales son procesos que por el número de transacciones a procesar y lo complejo de estos se tiene que ejecutar durante la noche y es hasta uno o varios días después cuando se tiene una respuesta del resultado del proceso básicamente lo que pretenden con estos servicios web es en esencia seguir manteniendo los procesos *BATCH* que ya conocemos pero ofreciendo una respuesta en línea a los usuarios mediante una consulta en internet sin necesidad de esperar varios días, es decir simular que esta comunicación con el negocio se hace de manera directa y a través de nuevas tecnologías como lo es un portal en internet o aplicaciones móviles que permitan tener esa misma funcionalidad.

Estos *Web Services* ya se encuentran implementados en algunos negocios permitiendo una interacción y convivencia entre los sistemas legados y sistemas de última generación de manera eficaz lo que trae grandes beneficios para el negocio como por ejemplo evitarse la necesidad de correr los altos riesgo en proyectos que logren migrar sus sistemas a otros con tecnología más reciente así como la reducción de gastos en actualizaciones de software y la posibilidad de reutilizar todos los elementos con los que ya cuenta el negocio.

Para cualquier negocio, tener herramientas a la medida adiciona ventajas competitivas contra sus similares ya que este puede ser modificado o mejorado cada que el negocio lo requiere, esto como resultado tiene mejoras en los procesos de negocio y da como resultado la preferencia de los clientes. En el caso de los sistemas comerciales la situación se torna diferente ya que los negocios se deben adaptar a las actualizaciones y mejoras que el proveedor de ese software ofrece creando una limitante la mejorar sus procesos.

La mayoría de los negocios aplican encuestas a sus usuarios con el fin de obtener estadísticas e información que les permita conocer los gustos y preferencias de sus usuarios a la hora de consumir un servicio o producto. Esto les permite tener las bases necesarias para ir mejorando sus procesos actuales de negocio y con esto ofrecer un servicio de mejor calidad. Para modificar los procesos de un negocio determinado se necesita que este cuente con procesos flexibles y fáciles de adaptar a diferentes metodologías, esto es debido a que los gustos de los usuarios cambian conforme pasa el tiempo y las exigencias en los distintos mercados están cada vez más orientadas a las tecnologías de información.

Es por esto que tener un sistema legado permite a los negocios modificar la mayoría de sus procesos a su conveniencia y de la forma en que ellos lo requieran adicionando ventajas competitivas al mejorar sus procesos actuales y modernizándolos usando medios electrónicos. Un claro ejemplo del uso de estas tecnologías son los servicios que ofrecen la mayoría de las instituciones bancarias como son consultas de saldos a través de internet, transferencias electrónicas, estados de cuenta en línea, pago de servicios vía internet entre otros servicios que son exclusivos de cada tipo de banca a fin de ir manteniéndose en la preferencia de sus usuarios.

De tal forma que si todas estas instituciones no contaran con sistemas de información hechos a la medida estarían limitadas y dependerían de las únicas herramientas y posibilidades que ofrece un sistema comercial tradicional las cuales en la mayoría de los casos son bastante limitadas y costosas.

2.2 Objetivos del mantenimiento de software hecho a la medida

Objetivo principal de acuerdo con Sommerville:

El mantenimiento del software es el proceso general de cambiar un sistema después de que éste ha sido entregado. El término se aplica normalmente a software a medida en donde grupos de desarrollo distintos están implicados antes y después de la entrega. Los cambios realizados al software pueden ser cambios sencillos para corregir errores de código, cambios más extensos para corregir errores de diseño o mejoras significativas para corregir errores de especificación o acomodar nuevos requerimientos. (2005, p.451)

Es muy común que distintos grupos de personas estén involucrados en este proceso debido a que no siempre es el proveedor del sistema quien realiza el mantenimiento, hoy existen infinidad de empresas que realizan estas actividades con distintas tecnologías y plataformas.

Debido a que el área de las tecnologías de información es una área muy demandada en la actualidad se requiere de mucho personal para atender todas estas solicitudes por parte de las consultorías que ofrecen el servicio de *outsourcing*, causa de esto es que los programadores tengan que volverse más flexibles y deban tener conocimiento en distintos lenguajes y plataformas ya que su estancia en un lugar depende de lo demandante que sea este, y en mucho de los casos los programadores tienen que estar migrando de lugar en lugar empezando de cero cada que lo hacen.

El nivel de conocimiento y experiencia en resolver problemas a nivel sistemas requerido por parte del personal que realiza los trabajos de mantenimiento sólo es posible conseguirlo a través de muchos años de experiencia, la cual en la mayoría de los casos pocos programadores tienen, esto es un gran problema a la hora realizar los trabajos de mantenimiento ya que estos sistemas requieren ser manejados con delicadeza debido a que la mayoría de estos dejaron de ser sistemas hechos a la medida para convertirse en grandes sistemas legados de carácter crítico para sus organizaciones y la forma en la que son adecuados, actualizados y corregidos debe ser de manera precisa tratando de afectar de manera mínima la funcionalidad que ya tienen, y esto únicamente se puede lograr con personal que sepa exactamente que está modificando y como debe realizarlo.

Con esto es posible realizar trabajos de mantenimiento de software de la manera más limpia como resultado se consigue incrementar el tiempo de vida útil de estos grandes sistemas a través de actualizaciones que permitan a estos programas cumplir de la manera más confiable con los requerimientos del negocio sin que haya necesidad de corregirlos una vez que son liberados a un ambiente productivo.

Además de cubrir reglas y políticas establecidas por parte de negocio estos sistemas deben de tener la capacidad de interactuar con las nuevas tecnologías o plataformas a fin de permitirles estar a la vanguardia e ir a la par de las nuevas tecnologías de información esto con el fin de evitar que estos sistemas se vuelvan obsoletos.

Las modificaciones de mantenimiento de software se aplican a componentes del sistema ya existente o recientemente liberados a un ambiente productivo en algunos de los casos añadiendo uno o varios componentes que sean necesarios.

Se ha podido ver en distintas ocasiones cuando se liberan adecuaciones o incluso nueva funcionalidad a un ambiente productivo que algunos de estos cambios fallan y es necesario corregirlos de manera inmediata, para este tipo de hallazgos existe una área especializada en fallas que todo departamento en sistemas debe tener mejor conocida como soporte a la producción, la función principal es la de corregir de manera casi inmediata todos estos incidentes los cuales en la mayoría de los casos son errores de programación pero sin generar costos adicionales al negocio.

En el peor de los casos cuando se es detectado un error de diseño es necesaria la apertura de todo un proyecto de modificación de software en el cual se requiere mucho más personal entre los cuales están los analistas de negocio, integradores tecnológicos, analistas de sistemas, personal *testing* así como todos los programadores que tendrán que realizar las modificaciones a nivel código, lo que cual genera grandes costos para el negocio así como un impacto muy grande en la funcionalidad del sistema, esto repercute de manera grave en el nivel de preferencia que los negocios pueden perder.

Mantenimiento para adaptar el software a diferentes entornos operativos. Este tipo de mantenimiento se requiere cuando cambia algún aspecto del entorno del sistema, como por ejemplo el hardware, la plataforma del sistema operativo u otro software de soporte. El sistema de aplicaciones debe modificarse para adaptarse a estos cambios en el entorno.

Este tipo de mantenimiento cada vez es más común verlo hoy en día, debido a que desde hace más de una década surgió la necesidad de lograr una interconexión entre tecnologías más recientes con sistemas hechos a la medida que ya se han convertido en sistemas legados con las nuevas plataformas de comunicación. El que estos viejos sistemas se encuentran construidos con tecnologías antiguas no ha sido una limitante para que estos se hayan podido adaptar a las nuevas plataforma por el contrario ha sido un reto para ir renovando estos sistemas y así prolongar cada vez más sus tiempo de vida sin que los negocios se tengan la necesidad de tenerlos que reemplazar por algún otro.

Los avances en la parte hardware de estos sistemas también ha sido notables en épocas actuales, esta parte también es fundamental para que estos puedan funcionar y subsistir. Se han implementado mejoras en el nivel de procesamiento que estos equipos tienen mejorando la arquitectura de estos equipos permitiendo que puedan ser reemplazados por otros nuevos y más sofisticados pero con las mismas compatibilidades, es decir que puedan seguir alojando en ellos el sistema actual pero con un mejor rendimiento.

Otra mejora se refiere al almacenamiento de datos y a la calidad con la que este proceso se realiza, permitiendo a estos sistemas poder adaptarse a metodologías de bases de datos más recientes y eficientes que adicionan un nivel muy alto de compatibilidad hacia otras aplicativos, plataformas y lenguajes de programación, esto permite realizar la exportación de archivos de datos hacia distintos sistemas sin necesidad de aplicar procesos complejos para poder compartir información entre estos sistemas.

Todas estas mejoras garantizan la vida útil de todos estos sistemas por mucho más tiempo sin perder la estabilidad, exactitud y robustez que todos estos sistemas pueden ofrecer.

2.3 Importancia del mantenimiento de software hecho a la medida

Un factor importante en una empresa a la hora de adquirir un software hecho a la medida es la posibilidad de prolongar el periodo de vida útil de este, con el paso del tiempo y conforme va creciendo este tipo de sistemas se convierten en un elemento fundamental para cualquier tipo

de negocio, es por esta razón que el mantenimiento a este tipo de software se convierte en una actividad de negocio recurrente y muy importante.

Como se menciona en párrafos anteriores el mantenimiento de software es una actividad constante que abarca tanto aspectos administrativos como técnicos, de tal manera que cualquier tipo de trabajo que tenga como fin modificar software se considera trabajo de mantenimiento.

Estos negocios no están libres de tener fallas en sus sistemas así que cuando estas suelen ser detectadas siempre se busca la manera de atenderlas de manera casi inmediata de no ser así implicaría pérdidas muy fuertes para el negocio, como un ejemplo de este tipo negocios podemos referirnos a las instituciones bancarias, de gobierno y en general negocio que manejen un flujo muy alto de transacciones por día.

Adicional a las mejoras y correcciones a los sistemas de información hechos a la medida que se van aplicando conforme va pasando el tiempo otro punto relevante que vale la pena mencionar es la importancia que tienen estos procesos en la manera en la que el sistema evoluciona. Se ha podido ver en algunos sistemas que las funcionalidades que son implementadas no sólo adicionan nuevos servicios, si no que permiten que esas aplicaciones sean capaces de interactuar con muchos otros sistemas de información que no necesariamente se encuentran desarrollados con la misma tecnología, esto es de suma importancia para que cualquier software pueda seguir subsistiendo y brindando un servicio de calidad que supere cada vez más las expectativas de sus usuarios.

Con el paso de los años el mantenimiento realizado a estos sistemas permite ir perfeccionando cada vez más la manera de operar de estas aplicaciones haciendo que los procesos que ya ejecutan cada vez se hagan de manera más exacta y confiable reduciendo al mínimo los fallos. Para lograr todo esto también es necesario que el personal que lleva a cabo estos trabajos se especialice en distintos estándares de desarrollo mejorando cada vez más los procesos de mantenimiento de sistemas empleando metodologías de ingeniería de software más sofisticadas que garanticen un trabajo de calidad.

3. Revisión al caso S336

3.1 Antecedentes

Con base en la entrada en vigor de la nueva ley del seguro social del Instituto Mexicano del Seguro Social el primero de julio de 1997, inicia operaciones en este nuevo sistema de pensiones el cual tiene como principal objetivo el retirar todos los fondos de pensiones del IMSS eliminando por completo el control administrativo que esta tenía sobre los mismos. Todos los fondos recaudados hasta ese momento fueron entregados a empresas privadas denominadas Administradoras de Fondos para el Retiro (AFORE), que son instituciones financieras privadas de México, que administran fondos de retiro y ahorro de los trabajadores afiliados al Instituto Mexicano del Seguro Social y recientemente de los afiliados al Instituto de Seguridad y Servicios Sociales de los Trabajadores al Servicio del Estado. Su funcionamiento está regulado por la Comisión Nacional del Sistema de Ahorro para el Retiro, (CONSAR) y autorizado por la Secretaría de Hacienda y Crédito Público. (Unla.mx, 2017).

En mayo de 1996, la LSS creó las AFORES que iniciaron operaciones el 1 de julio de 1997, con la finalidad de ofrecer cuentas personales a los trabajadores y para que el ahorro. Generado a lo largo de su vida laboral, pudiera crecer con los rendimientos generados y, de esta forma, asegurar un retiro digno para los trabajadores y un manejo transparente de los recursos. (Contreras, 2011, p.45)

Estas instituciones están dedicadas a administrar las cuentas individuales de los trabajadores canalizando los recursos todas las subcuentas a las diferentes SIEFORES a fin de

que estos recursos sean invertidos con el propósito de obtener mejores rendimientos y así ofrecer a sus cuenta habientes o trabajadores una pensión más digna al momento de su retiro.

Algunas de las funciones principales de la AFORE son:

- Crear las cuentas individuales a petición de los trabajadores.
- Reciben las cuotas y aportaciones que los trabajadores, los patrones y el gobierno les entregan y, las depositan, en las cuentas individuales.
- Reciben y registran las aportaciones voluntarias de los trabajadores y en su caso, de los patrones.
- Administran e invierten el ahorro para el retiro a través de cada SIEFORE para obtener rendimientos.
- Registran en la cuenta individual, los rendimientos que vaya generando el ahorro para el retiro del trabajador.
- También registran en la cuenta individual las aportaciones para vivienda, que hace el patrón al INFONAVIT, que el propio Instituto administra, así como los rendimientos que generen.
- Envían un estado de cuenta cuando menos tres veces al año al domicilio que el trabajador indique, para que conozca los movimientos llevados a cabo en su cuenta y lleve el control de la misma.
- Entregan a los trabajadores los recursos por las prestaciones a que tienen derecho durante su vida laboral, por concepto de ayuda para gastos de matrimonio y desempleo temporal.
- Al momento del retiro, entregan al trabajador el monto correspondiente a su pensión.

Existen muchos tipos de AFORE entre las cuales destacan:

- Banamex
- Invercap
- SURA
- Profuturo GNP
- Principal
- Azteca
- Inbursa
- Pensión ISSSTE
- Coppel
- XXI Banorte

Existen notables diferencias en porcentaje de rendimientos que estas generan y la calidad de los servicios ofrecidos a sus usuarios. Esto se debe a que algunas de las AFORE son hijas de instituciones financieras con muchos más años de experiencia en el sector financiero las cuales ya cuentan con cierta preferencia en el mercado. AFORE BANAMEX es una de las AFORE más fuertes en el mercado, nacida ya hace poco más de 15 años bajo el respaldo del Banco Nacional de México lo que rápidamente la posiciono entre algunas de las preferidas por los ya clientes de este.

El 2 de junio de 1884, la fusión del Banco Nacional Mexicano y el Banco Mercantil Mexicano dio vida al Banco Nacional de México (Banamex) y el 6 de enero de 1997 se

constituyó la Afore Banamex, S.A. de C.V. la cual logro obtener el 27 del mismo mes la autorización para operar como Administradora de Fondos para el Retiro.

Pero el lograr esta autorización no fue sencillo, se necesitó cumplir con una cantidad inmensa de requisitos legales debido a que el tipo de información que se estaría manejando es considerada sensible y altamente confidencial, adicional a esto su sistema de información requirió de la aplicación de diferentes metodologías de desarrollo y estándares de seguridad utilizados a nivel mundial a fin de que su sistema de información cumpliera con los requisitos necesarios para procesar e intercambiar con otros sistemas externos información de la manera más segura, confiable y exacta.

Como toda organización de tipo financiero esta depende principalmente de sus sistemas de información para llevar a cabo sus operaciones diarias. AFORE BANAMEX como muchas otras es dueña de muchos de estos sistemas entre los que destacan algunos ‘legados’ construidos en distintas plataformas los cuales constituyen de alguna manera la mayor parte de su funcionalidad diaria.

Uno de los sistemas legados con mayor prioridad para el negocio es el sistema S336 Sistema de clientes AFORE, este es un sistema desarrollado ya hace más de 15 años con tecnologías que para el día de hoy son consideradas antiguas pero lo suficientemente robustas para brindar el soporte necesario al negocio de manera eficiente cabe mencionar que para que este sistema pueda funcionar necesita alimentarse de algunos otros sistemas más pequeños pero igual de importantes que éste.

El sistema digital de AFORE BANAMEX esta soportado en un mainframe UNISYS, el manejador de la base de datos es DMSII (*Data Management System II*) la cual es una base de datos de tipo jerárquica y el lenguaje de programación principal es COBOL74 (*COmmon Business Oriented Language*), aunque también podemos ver algunos programas que corren junto a los procesos principales del sistema contruidos en lenguaje ALGOL y WFL (*Work Flow Language*).

Actualmente existen 6 equipos de trabajo:

ACYPBETA	Para pruebas unitarias e integrales (SIT por sus siglas en inglés).
VDMBETA	Para pruebas de integrales y de usuario (UAT por sus siglas en inglés)
ACYPGAMA	Para pruebas de usuario y ambiente
MONALFA	Para validación de seguridad
VDMALFA	Para producción Valle de México
MONBETA	Para producción Monterrey

El sistema de la AFORE BANAMEX está dividido en tres grandes sub-sistemas:

S336	Administrador de la operación y de las cuentas de los clientes
S584	Saldos y movimientos de las cuentas individuales
S550	Estado de cuenta en línea y papel

En el sistema los programas se denominan pasos, existen tres maneras de procesarlos:

En línea (On line)	Programas en COBOL controlados a través de la aplicación ADMWIN (Pantallas)
BATCH (Por lotes)	A través de programas COBOL, controlados por <i>WFL (Work Flow Language)</i> los cuales son ejecutados al final del día (durante la noche mejor conocidos como procesos <i>BATCH</i>)
Combinado	Mejor conocido como ‘MiniBATCH’ es un <i>WFL</i> que ejecuta programas <i>BATCH</i> lanzados desde procesos <i>En línea</i> .

3.2 Problemática

El sistema S336 de AFORE propiedad Grupo Financiero BANAMEX es un sistema que se ha distinguido desde su nacimiento por tener un crecimiento constante de manera poco organizada. Existen infinidad de factores involucrados en este crecimiento descontrolado, pero, uno de los principales se deriva a raíz del negocio en el que este sistema se encuentra implementado.

AFORE BANAMEX así como el resto de las administradoras son negocios estratégicos que se encuentran normados y vigilado por las autoridades, de forma que si alguna de estas no cumple debidamente con lo establecido por la ley podría sufrir multas multimillonarias o incluso perder la licencia para seguir operando.

El S336 de entre otros tantos sistemas legados propios de BANAMEX es la herramienta principal de AFORE BANAMEX para llevar a cabo las operaciones diarias de negocio, por lo que es muy común que año con año se lleven a cabo más de 50 proyectos de modificación de software en los cuales se realizan modificaciones a gran parte de componentes propios de este sistema.

Ramírez comentó recientemente que este sistema nació bajo una planeación muy precipitada por lo que fue necesario ser diseñado bajo los cimientos del viejo sistema existente del SAR, varias de las piezas que componen el actual sistemas son copias casi idénticas que conservan además de una cierta similitud en la funcionalidad la mayoría de errores lógicos y de programación que se tenían en el anterior sistema. Con el paso de los años y de acuerdo a los comentarios de algunos de los ingenieros con más años de experiencia y conocimiento dando mantenimiento al S336 estos errores han tratado de corregidos, la mayoría con éxito, sin embargo siguen existiendo algunos otros que son casi imposibles de solucionar debido al alto nivel de complejidad que tiene las piezas por lo que realizar modificaciones tan abruptas podría ocasionar fallas críticas en el sistema por lo que estas deben hacerse de manera progresiva.

Durante la transición de lo que anteriormente era el SAR a las AFORE la necesidad para el negocio de contar con un nuevo sistema que fuera de capaz de cubrir las necesidades de procesamiento de información eran demasiada si quería operar al mismo tiempo en que lo haría la competencia. El tiempo disponible para llevar esto a cabo era demasiado limitado por lo que el grupo de desarrollo tuvo que optar por una alternativa que dadas las circunstancias era la mejor

opción en ese momento para cumplir en tiempo con lo requerido. (P. Ramírez Grande, Comunicación personal, Diciembre de 2015)

Esta solución constaba de realizar una copia de algunos de los programas que componían sistema del SAR debido a que compartía una filosofía que se asemejaba casi en su totalidad a lo que se requería para el sistema del nuevo modelo de pensiones. A pesar de que estos programas compartían gran parte de su funcionalidad fue necesario modificar casi la mayoría de los piezas de software que habían sido tomadas como base para generar lo nuevo, se manejó un formato de piezas casi idéntico al del sistema original a fin de respetar la mayoría de reglas de negocio con las que se debía operar. Esto es observable aún en el código fuente por los vestigios que hoy en día se pueden encontrar como comentarios de programa.

El equipo de ingeniería en sistemas decidió llevar a cabo el desarrollo del sistema de esta manera al ser la única opción que permitiría cumplir en el tiempo establecido la entrega de un sistema lo suficientemente capaz de cumplir de manera total con los requisitos establecidos por los reguladores y así obtener la certificación necesaria para iniciar operaciones en la fecha definida.

Con el paso de los años surgió la necesidad de realizar varias modificaciones al sistema algunas para aplicar ajustes al proceso y otras para adicionar alguna funcionalidad de tal forma que iba creciendo de manera rápida hasta que se llegó a un punto en el que se detectó que la solución de tomar como base el sistema anterior en lugar de diseñar una nueva solución terminó perjudicando de manera grave al S336, debido a que heredaba además de la funcionalidad los

errores que ya se tenían en el sistema anterior. Estos defectos podían describirse desde errores muy simples de funcionalidad hasta problemas de diseños bastante complejos, para los cuales era necesaria una gran inversión en la apertura de proyectos para modificación de software.

Conforme ha pasado el tiempo se ha podido ver a la hora de dar mantenimiento a estos programas que algunos de los errores antes mencionados aún se siguen persistiendo en las piezas de software más antiguas o en aquellos que nacieron junto con el sistema debido a que fue en estos en los que más errores se heredaron. A pesar de los proyectos de modificación de software que consideran entre sus objetivos principales eliminar en su totalidad todos estos fallos no ha sido suficiente durante todos estos años y hasta el día de hoy se sigue trabajando para pulir en su totalidad el funcionamiento de este.

Algunos de los problemas más comunes a las que se enfrentan los desarrolladores se describen a continuación con un detalle más claro.

A raíz de la falta de experiencia para conceptualizar, planear y definir un proyecto de software por parte de los líderes es muy común que al momento de llevar a cabo la planeación un proyecto nuevo fallen. Las personas involucradas o bien a quien se considera líderes de proyecto son el usuario final del sistema, Líder del proyecto (quien lo administra), Analista de negocio, Integrador Tecnológico, que son los responsables de definir de manera exacta el alcance del proyecto así como todos los requerimientos del negocio. Es necesario que estos actores trabajen en conjunto ya que son los que tienen el mayor conocimiento en cuanto a reglas de negocio y un panorama bastante completo sobre la manera en la que opera actualmente el sistema. Pero todo

se complica cuando alguno de estos no comprende exactamente lo que se requiere o no tiene el conocimiento necesario para aportar lo que se requiere a la planeación del proyecto. Esto sucede casi en el 95% de los proyectos es lo que provoca que casi siempre desde sus inicios los proyectos de software ya presenten atrasos y de ahí en adelante tiempos cortos en las etapas subsecuentes lo que impide en la mayoría de los casos que se pueda lograr que el proyecto concluya en los tiempos establecidos sin necesidad de alargar más estos y sobre todo que se pueda entregar un trabajo con calidad que se requiere.

Ambigüedades por parte del analista de negocio y usuario final al momento de definir los requerimientos. Es muy común que algunas de las partes que intervienen en la planeación de un proyecto no comprenda claramente lo que se requiere o quizás que el usuario no lo haya definido correctamente. estas definiciones que quedan pendientes aun después de la etapa de planeación en ocasiones suelen ser resueltas ya avanzado el proyecto, esto casi siempre tiene como resultado que en los componentes que son modificados casi siempre presenten errores de funcionalidad, errores de código o más grave aún, que impacten de manera directa a otro modulo que ya operaba correctamente debido a que cuando estas ambigüedades son resueltas se cuenta con un tiempo bastante reducido para aplicar los cambios y las pruebas que se realizan a los componente no son lo bastante completas para identificar errores de este tipo.

Falta de experiencia técnica de los programadores. La plataforma bajo la que se ejecuta el S336 a la fecha es totalmente desconocida para muchos y la posibilidad de que esta sea impartida como materia opcional en las instituciones educativas actuales es totalmente nula. por esta razón la mayoría de los ingenieros egresados o no que se incorpora a esta área de sistemas en donde las

herramientas que se utilizan para dar mantenimiento a sistemas legados así como para desarrollar software nuevo para un sistema de estos suelen desconocer la mayoría de estos componentes, de tal forma que la curva de aprendizaje por la que deben de pasar esta bastante limitada no permite que adquieran la experiencia suficiente pero si incluye la construcción y liberación de software que es requerido por el proyecto, estos componentes liberados a raíz de la inexperiencia no considera el uso de estándares o todas aquellas reglas de desarrollo definidas. Como resultado al final del proyecto y en la etapa de implementación se tienen infinidad de fallas en el sistema que afectan de manera significativa al negocio debido a que estas suelen relucir en el ambiente productivo cuando es necesaria la operación del sistema. Estas incidencias suelen convertirlos en defectos graves conforme va pasando el tiempo afectando la funcionalidad del resto de los módulos que componen el sistema o peor dejando inconsistencias en la información que se almacena y transaccional con el resto de las entidades.

Falta de experiencia en el área de negocio de los programadores. Además de la experiencia técnica que se requiere tenga el personal de desarrollo de software es necesario que estos conozcan la manera exacta en cómo opera el negocio, esto puede permitir que se comprenda de una mejor manera las necesidades que el negocio tiene y así establecer un canal de comunicación más claro entre ambas partes y así resolver dudas de una manera más ágil pero sobre todo que se pueda ofrecer un diseño de solución eficiente. Aunque en la intranet de la organización existen algunos manuales de operación no son lo suficientemente útiles para el personal de desarrollo, esto se debe a que van dirigidos al personal de operaciones y utilizan terminologías distintas lo cual limita que un programador se capaz de comprender todo el documento. De igual forma la inexperiencia en la parte técnica el desconocer el tipo de negocio

en el que operan todos estos sistemas provoca retrasos, fallas y tiempos demasiado prolongados en los proyectos de mantenimiento de software en sistemas legados, por eso es necesario que los programadores tengan una capacidad de análisis lo suficientemente alta para lograr una comprensión total de estos sistemas tanto a nivel técnico como es la comprensión del código fuente de las piezas como de las reglas de negocio implementadas en la lógica de programación de estos programas garantizando que el nivel de incidencias será mínimo.

No existe documentación sobre la manera en la que opera actualmente el software. Aunque el personal de desarrollo tenga un conocimiento bastante amplio sobre la mayor parte del sistema, en ocasiones se les complica realizar de manera ágil el análisis de los componentes de software a modificar, esto se debe a que no existe información histórica sobre la funcionalidad de cada uno, ni se cuenta con información completa sobre los antecedentes de modificación que ha sufrido cada componente. Para lograr comprender lo que hace cada pieza de software los tiempos para el análisis puede llevarse semanas dependiendo de la capacidad del analista, lo que casi siempre provoca retrasos en proyecto y en algunas ocasiones fallas en el sistema al momento de liberar el nuevo software al ambiente productivo debido a que al no conocer totalmente la manera en que ejecutan los procesos suelen excluirse algunos que se pensaba no eran necesarios para la nueva funcionalidad.

Falta de personal capacitado en áreas administrativas. Así como el personal de desarrollo es de suma importancia en el área de sistemas también se requiere de especialistas para llevar ciertas actividades que requieren de la gestión de distintos componentes para que un proyecto de mantenimiento de software se concluya de manera satisfactoria, pero en la mayoría de las

ocasiones los especialistas encargados de estos procesos no cuentan con la experiencia necesaria en la metodología para administración de proyectos que es usada en AFORE. Uno de los principales problemas generados por estos actores son retrasos en el proyecto en su mayoría debido a que casi siempre existen documentos entregables que no son considerados o son entregados de manera tardía.

Requerimientos de negocios bastante confusos o no bien definidos. Cuando se está diseñando un nuevo módulo o alguna componente nuevo existen definiciones que quedan pendientes por definir, la mayoría de las veces el usuario final a pesar de haber analizado y definido 'claramente' sus necesidades no lo hace de la manera correcta, lo que sucede es que por falta de conocimiento o inexperiencia omite definiciones que son necesarias para que su nueva funcionalidad sea capaz de operar como él la requiere, detalles que él y sólo él puede definir lo cual como en los casos anteriores provoca retrasos en el proyecto, tiempos más cortos para desarrollo, diseño y construcción para nuevos componentes con tiempos bastante reducidos y con una posibilidad muy alta de que contenga fallas al momento de ser implementados.

No existe documentación sobre el software al que se estará dando mantenimiento. Al momento de analizar de manera específica cada una de las piezas que requieren ser modificadas no se tiene la información histórica sobre sus funcionalidad, o las modificaciones que se han aplicado a estas, aunque es necesario casi nunca se cuenta con esta información ya que la persona que modifica cada pieza no genera información sobre los cambios que está aplicando, y cuando la generan si no está incompleta o muy austera casi nunca suele ser compartida al resto del equipo y se queda individualizada sin que le pueda servir a alguien. El que la información no

sea compartida la mayoría de las veces se debe a que no existe una persona encargada de recopilar toda esta información, no se tiene un método para concentrar los documentos de este tipo que se genera por cada programador, que aunque estos quieran compartir tiene que hacerlo de manera individual ya que no existe un medio para la difusión de este tipo de información.

Lógica confusa, en modificaciones realizadas al código fuente. Debido a que no se ha hecho uso de una metodología de desarrollo cada programador piensa de diferente manera, de tal forma que cada uno modifica los componentes de software como mejor le parece sin respetar estándares y agregando lógica de programación bastante 'rebuscada' que además de complicar el análisis suele dificultar las modificaciones que se deben hacer a determinada pieza de software, esto dificulta tener una garantía total de que los cambios aplicados no tenga afectación a algún otro modulo del sistema ya que como se sabe existen piezas que corren en casi todos los módulos por lo que el cuidado que se debe tener al momento de modificarlas es muy elevado, casi siempre tienen defectos que son detectados cuando se encuentra en ejecución en el ambiente productivo y deben ser corregidos al instante.

Colisión con otros proyectos. Varios proyectos de distintos módulos se llevan a cabo en paralelo la mayoría de las veces se da la situación de que uno o más se encuentre modificando el mismo programa y existan afectaciones a causa del tipo de cambio que se encuentra realizando, por ejemplo, si dos proyectos se encuentran modificando la misma pantalla la manera en que desarrolla un programador a otro es totalmente diferente y es muy complicado lograr acuerdos para la manera en que se deberán hacer las modificaciones para no afectar al otro. Además de lo anterior tenemos otra situación que se da al momento de la liberación cuando pareciera que no

existe conflictos de ningún tipo, pero justo antes de enviar las piezas a producción se requiere nivelar las piezas con los cambios de los proyectos involucrados si es que se instalan en la misma fecha, y es ahí cuando sobre salen los conflictos de modificación por lo que el tiempo que tienen para resolverlos es muy limitado, y casi siempre se tiene errores.

Falta de experiencia de los líderes técnicos e integradores tecnológicos. Los responsables de coordinar al personal de desarrollo y resolver dudas técnicas, de negocio y de definición de requerimientos son los integradores tecnológicos apoyados de los líderes técnicos, ya que son las personas con más experiencia y conocimiento en el sistema, estos se clasifican por determinados módulos como lo es retiros, separación de cuentas, unificación de cuentas, trasposos, afiliación, entre otros. cuando una de estas dos partes no cuenta con el conocimiento necesario o no tiene la capacidad de diseñar o proponer soluciones alternativas el grupo que depende de él no lleva a cabo sus actividades de la manera correcta de tal forma que la mayor parte del software construido o modificado presenta errores de diseño graves que son detectados hasta la etapa de pruebas en donde deben ser corregido con un límite de tiempo muy limitado, y por consiguiente con el riesgo de que la corrección presente errores.

Tiempos asignados para desarrollo de software muy limitados. Como se mencionó anteriormente existen muchos factores que provocan que un proyecto de software presente retrasos y tiempos para construcción bastante reducidos, esto exige que los programadores trabajen en algunos casos a marchas forzadas poniendo en riesgo la calidad del trabajo. Debido a lo normado que se encuentran este tipo de negocios el incumplir con algunos de los requerimientos es impensable y muchas de las veces aun con el riesgo de fallas los proyectos son

instalados, asignando tiempo adicional para su corrección lo que implica re trabajo para todo el equipo, y calificaciones negativas en la métricas establecidas por la institución para control de calidad de los productos. la finalidad primordial de los proyectos es que sean instalados sea como sea, y aunque casi siempre se cumpla el objetivo, durante el proceso de corrección de fallas no todos los errores son debidamente corregidos y estos pueden permanecer por muchos años sin ser detectados hasta que sobresalen en algún otro proyecto, lo que provoca que la responsabilidad de corregirlo recaiga en ese nuevo equipo que no fue el que modifco los componentes que presentan la falla y se compliquen mucho más llevar a cabo su construcción.

Falta de uso de estándares de desarrollo. Cada que una pieza se modifica, es probada y cumple con la funcionalidad requerida, lo siguiente que se requiere es la validación por parte del *checker*, quien es la persona encargada de validar que las modificaciones a las piezas cumplan con los estándares básicos establecidos, la mayoría de las veces en la primera revisión las piezas son rechazadas ya que incumplen con las reglas establecidas, los errores persisten y el tiempo que se lleva en corregir las modificaciones es en algunos casos excesivo debido a que no son considerados durante la etapa de construcción por falta de conocimiento del personal de desarrollo, en situaciones más graves es hasta este punto en donde se detecta impactos a otros módulos por lo que es necesario un rediseño de la solución el cual provoca casi siempre la prolongación del proyecto por dos o tres meses más.

Ausencia metodología de desarrollo bien establecida. Aunque si exista una metodología orientada a la gestión de proyectos que permita que este se logre de manera satisfactoria para área de desarrollo no es así, ya que no se cuenta con una metodología para el desarrollo de

software establecida que permita la clasificación por etapas en esta área. La mayoría del personal trabaja de la manera en que mejor se adapta sin tomar en cuenta como deberían llevarse a cabo un desarrollo de software de manera ordenada lo que provoca que al momento de realizar una prueba integrando todos los componentes, algunos aun no estén listos o estén aún en proceso siendo primordiales para el ciclo de pruebas que se debería estar llevando a cabo. de acuerdo a la experiencia de cada persona realiza sus tareas aplicando distintas clasificaciones, como es por nivel de complejidad, por tipo de proceso (*BATCH*, *en línea* o Base de datos), por tamaño de pieza, por sistema (cuando se requiere la modificación de componentes que pertenecen a un sistema distinto pero que interactúan con el nuestro), en algunos casos si son piezas nuevas o solo se requiere modificación, por modulo (cuando se modifican más de dos módulos), entre otras, esto provoca una desincronización de los componentes que se encuentran en modificación y una gran dificultad para cuando se requiere estar en etapa de pruebas. Generalmente este tipo de situaciones se pueden notar en los programadores egresados que se incorporan al área de los sistemas legados, los cuales traen consigo el uso de otras formas de desarrollo distintas que no son adaptables en su totalidad a los viejos sistemas legados como el S336.

Errores de programación en el nuevo software. Actualmente se siguen creando programas nuevos para el viejo sistema legado S336, sin embargo estos siguen manteniendo los mismos errores que los programas más antiguos ya que la mayoría de los programadores toman como base los programas anteriores y simplemente realiza adaptaciones para que cumplan con la nueva funcionalidad que se requiere, esto es un gran problema ya que el software nuevo no es depurado debidamente y se recae en lugar de minimizar los problemas que ya se tienen, se siguen haciendo cada vez más notorios conforme crece el sistema.

Ambiente de desarrollo inestable. El ambiente de desarrollo en este tipo de sistemas es un entorno compartido en el que todos los usuarios que se conectan tiene acceso a casi todos los directorios y cuentas con acceso para ejecutar instrucciones de cualquier tipo sin restricción alguna, esto origina distintas problemáticas como son el uso de instrucciones de borrado de directorios, archivos, programas, el planchado de componentes como archivos planos, programas fuente y ejecutables, espacio de almacenamiento saturado constantemente, generación de información y archivos basura, saturación del procesador lo que provoca la lentitud en el equipo, dependencia de la base de datos y etapas de pruebas que limitan a muchos usuarios a ciertos horarios para continuar con sus actividades.

Ambiente asignado para pruebas inestables. Una de las etapas más complejas de llevar a cabo dentro de la vida de un proyecto de software es la de pruebas, hoy en día resulta ser bastante complicado realizar pruebas sin tener retrasos o incidencias en el ambiente, esto debido a que el equipo en el que se realizan las pruebas del sistema es el mismo en el que se lleva a cabo el desarrollo de software de tal forma que mucha gente se encuentra haciendo distintas actividades y casi siempre existe colisión en el usos de programas, información de la base de datos, fechas del sistema, versiones de programas, pantallas, entre otros. no se cuenta con una persona que administre el ambiente entre otros, por lo que se vuelve bastante complicado concluir con este proceso en tiempo y formas establecidos, mucho menos existe una educación por parte del personal para el uso del ambiente lo que impacta gravemente a todos los proyectos que se encuentran en ejecución. Adicionalmente todas las fallas productivas que surgen a diario tiene un nivel de prioridad alto por lo que si se requiere realizar alguna maniobra en el ambiente deben detener a todos los que se encuentre realizando alguna actividad en el equipo de desarrollo

para que se lleve a cabo la atención de estas fallas que afectan de igual manera los tiempos y fechas establecidos por los demás proyectos.

Se heredan los mismos errores en software nuevo. El S336 es un sistema que hasta la fecha sigue creciendo con piezas nuevas de software que son copias similares de otros pero adaptados de acuerdo a lo que se requiere, esto implica que además de cierta funcionalidad también se estén heredando ciertos defectos que en ese momento podrían ser corregidos o simplemente mejoras que podrían hacerse para que estos nuevos componentes funcionen de una manera más óptima. Esta recomendación pocas veces se aplica ya que no existe un compromiso propio del personal de desarrollo por entregar un producto de calidad si no que solo se limitan a que estos productos cumplan con lo mínimamente requerido son consideran un esfuerzo extra en implementar mejoras.

Defectos de programación dentro de todas las piezas de software del sistema. De acuerdo a lo que se ha podido ver al momento de dar mantenimiento al sistema S336 que falta puntos importantes a nivel código fuente que deberías ser clasificados y validados antes de realizar el envío de alguna pieza a producción los cuales se clasifican de la siguiente manera:

Código fuente que nunca se ejecuta

- Variables
- Procedimientos

Código fuente comentado

- Densidad de comentarios
- Rutinas
- Variables
- Instrucciones

Nombres fuera de estándar

- Rutinas
- Variables

Código ejecutable “destrutivo”

- Instrucciones ALTER
- Instrucciones ENTRY
- Instrucciones DIVIDE BY 0

Código duplicado

- Rutinas con funcionalidad similar
- Rutinas con la misma funcionalidad

Alineación de código fuente

- Instrucciones
- Variables
- Rutinas

Incumplimiento de metas de los programadores. Aun cuando no es bueno, es normal que cuando los tiempos de construcción de un proyecto sean limitados existan retrasos en cuanto al desarrollo; pero en ocasiones suele suceder que aunque se tengan tiempos de construcción considerables haya retrasos por falta de experiencia de los programadores, y a pesar de que diariamente se de una seguimiento al avance que llevan esto no sea cierto y al momento de entregar el producto se encuentre incompleto o no tenga la funcionalidad que se requiere, esto es seguro que retrasara al resto del equipo y provoca retrasos y errores graves en el desarrollo de los programas.

Reemplazo de piezas a liberar (código fuente). En el S336 las piezas residen en el mismo equipo, tanto la última modificada como las anteriores. De acuerdo al proceso de liberación cuando se va a realizar un envío de software al ambiente productivo la pieza a enviar debe residir en el mismo equipo, por lo que es muy común que gente con o sin experiencia por alguna situación remplace esas piezas y se haga él envío del nuevo software pero de versiones incorrectas lo que tiene como resultado final fallas en el sistema, trueno de programas y que la ejecución de los procesos quede incompleta dejando inconsistencias en la información.

Adición de atención a fallas en liberaciones de proyectos planeados y fallas al momento del envío de piezas a producción. Existen diversas actividades administrativas que se requieren hacer en conjunto con el personal de desarrollo para llevar a cabo una liberación exitosa de estos proyectos, entre estas tareas se encuentran la generación de manuales de instalación que indican paso a paso como se debe realizar esta tarea incluye además de software algunos otros componentes que algunas veces fallan, a pesar de que pasaron por la validación del *checker*.

Cuando sucede esto es necesario rehacer la mayor parte de los documentos que ya se habían entregado algo similar sucede cuando se da prioridad a la instalación de alguna corrección causada por falla en el ambiente productivo por lo que es necesario actualizar los documentos que ya se tenían agregando el software extra que de igual forma debe ser instalando, esto incrementa el riesgo de impacto a la funcionalidad que ya se traía ya que no se valida que las correcciones por falla que se van a liberar no tengan colisiones con lo nuevo.

3.3 Propuesta

De manera general considerando las problemáticas actuales del sistema S336 se requiere aplicar un proceso de reingeniería para el cual es necesario realizar una revisión de los siguientes elementos:

- Código fuente
- Datos del sistema (información en base de datos)
- Documentación existente del sistema (Especificaciones técnicas, manuales de operación, documentos de diseño)
- Manuales de negocio (manera en la que se encuentra operando el negocio o bien reglas de negocio oficiales documentadas)
- Manuales técnicos (manuales de programación, estándares o metodologías utilizadas)

Autores como Sommerville la definen de la siguiente manera:

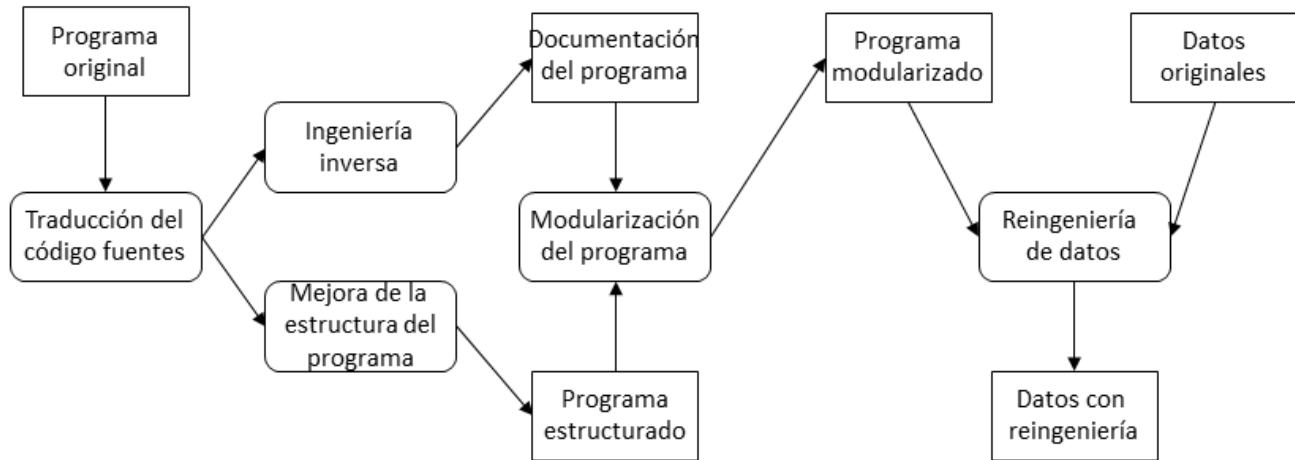


Figura 1. El proceso de reingeniería Tomado de Sommerville (2009).

Casi en todas las definiciones el aplicar una reingeniería de software implica la reconstrucción total del sistema en lenguajes más modernos, lo que significa algunos conceptos de un proceso de reingeniería de software pueden ser tomados en cuenta pero sin cambiar de plataforma o lenguaje. Basándonos en el modelo propuesto por Sommerville en la figura 2, la idea principal para ser aplicada el S336 es más bien la de crear un modelo que considere una serie de pasos que permitan logra una extracción de información de todas las piezas del sistema tanto a nivel técnico como de negocio y así lograra que poderle dar mantenimiento sea más sencillo.

Partiendo de los principios básicos de reingeniería de software a continuación se describen los puntos que se proponen se lleven a cabo en caso particular del S336.

1. Revisión a código fuente
 - a. Traducción del código fuente
 - b. Ingeniería Inversa
 - c. Mejora en la estructura de los programas
 - d. Modularización de los programas
 - e. Documentación del programa
2. Revisión documentación del sistema
 - a. Especificaciones técnicas
 - b. Modelos de datos
 - c. Documentos de diseño técnico
 - d. Código fuente
 - e. Documentación del usuario (negocio)
 - f. Documentación técnica
 - g. Diagramas funcionales
 - h. Interfaces o pantallas actuales
 - i. Información del sistema (datos almacenados en el sistema)
 - j. Documentos de prueba (guiones de prueba y casos de prueba)
3. Revisión de datos del sistema
 - a. Reingeniería de datos

Revisión a código fuente

Según García, un proceso básico de reingeniería a piezas de software se define de la siguiente manera:

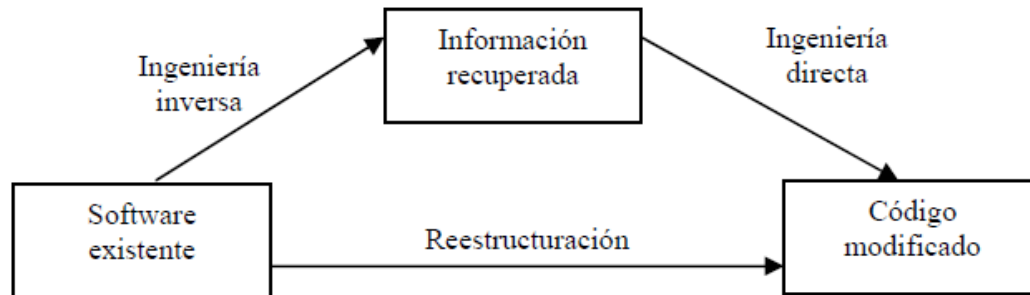


Figura 2. Proceso básico de reingeniería. García (2004).

El modelo que se observa en la figura 1 considera básicamente una extracción de la funcionalidad principal de cada uno de los componentes de software a fin de depurar las piezas originales y generar software nuevo.

Para la ejecución de este punto es necesario revisar inicialmente las piezas de software más críticas, considerando como crítico aquellas piezas que cumplan con los siguientes criterios:

- Piezas con más fallos reportados
- Piezas que se ejecutan más veces dentro del sistema
- Piezas de software con mayor número de líneas
- Piezas que generen productos críticos o de alta importancia
- Piezas que sean constantemente modificadas

Una vez teniendo identificadas todas aquellas piezas software que cumplan con la definición anterior es necesario que personal experto analice e identifique la funcionalidad principal de estos pequeños programas a fin de obtener toda la información necesaria sobre cómo operan estos programas para que posteriormente esta piezas pueda ser rediseñada, organizada y reescrita de la manera más óptima omitiendo la mayor cantidad de errores posibles.

El S336 al ser un sistema desarrollado y mantenido por muchas personas que casi siempre usan métodos de desarrollo diferentes y estilos propios de programación ha causado que lograr un entendimiento simple de la manera en la que funciona el sistema sea bastante complicado es por eso que se necesita gente experta que sea capaz de cumplir con esta actividad.

Traducción del código fuente

Tomando las últimas versiones de las piezas de software liberadas a producción para analizarlas y generar documentos que permitan extraer la lógica principal y prioritaria de estos programas a fin de que sean replicados en piezas nuevas de software reconstruidas bajo estándares establecidos por las reglas propias del lenguaje en el que se encuentran hechos, estándares de la institución y las reglas de negocio que deban cumplir, se pretende plasmar en este documento los puntos elementales del código fuente a un nivel entendible para cualquier persona, lograr así un entendimiento total de cada una de las piezas de software y trabajar de manera conjunta con el usuario final, analista de negocio y personal de desarrollo para complementar o en su caso corregir lo que ya se tiene y lograr una construcción de software más limpia.

En este punto es importante lograr un análisis detallado de las piezas para identificar exactamente su funcionalidad, se puede echar mano a diagramas o a la documentación que ya se tenga para tratar obtener una idea de lo que se debe encontrar al momento de realizar la revisión.

Ingeniería Inversa

Algo similar a lo que se propone en el punto anterior pero aplicándolo de manera más formal, se propone realizar un análisis de las piezas para extraer la funcionalidad más relevante de cada pieza de software, depurándola en su totalidad para obtener como resultado software sin herencia de errores, es decir, que al momento de estar extrayendo todos esos módulos de código fuente exista la posibilidad de ir mejorando las técnicas de programación y lógica que fueron aplicadas al momento de generar por primera vez esas piezas, como ejemplo podemos mencionar:

- Lecturas en base de datos
- Lógica de programación
- Uso de estándares de desarrollo establecidos por la institución y propios del lenguaje

entre otros puntos más a considerar en este punto no se planea dejar listas piezas de software al 100% pero si bloques de código que permitan posteriormente reconstruir la pieza de manera más óptima complementando esta extracción de funcionalidad que se tiene con la información obtenida en el punto anterior.

Mejora en la estructura de los programas

Conjuntando la información obtenida en los puntos anteriores se planea la definición de un modelo de desarrollo que permita lograr una clasificación de bloques de código que permitiría llevar a cabo la modificación de cualquier piezas de una manera sencilla que simplifique la depuración o adición de funcionalidad y que además garantice una afectación mínima a lo que permanezca.

Se propone la siguiente clasificación de bloques para cada pieza de software:

- Bloque para llamado a las librerías
 - Propias del sistema
 - De sistemas diferentes
 - De negocio
 - De sistema

- Bloque para generación de archivos
 - Apertura
 - Cierre
 - Validación
 - Grabado
 - Lectura
 - Procesamiento

- Bloque para procedimientos de bases de datos

Apertura

Cierre

Lecturas

Actualización

Borrado

Alta

- Bloque para manejo de eventos (En este apartado se incluye instrucciones generales propias del lenguaje)

Para stop al proceso

Para consulta del status proceso (al momento del *BATCH*)

Para recuperación de bitácora o *log* en caso de falla

Para dar alguna instrucción al proceso que implique procedimientos adicionales en el proceso en tiempo de ejecución.

- Bloque para el manejo procedimientos de negocio

Incluirá toda la lógica de negocio principal del proceso, la cual adicionalmente estará plasmada en documento que describa claramente lo que hace el proceso.

Cabe mencionar que es posible que no en todas las piezas puedan ser aplicar estas ya que van de acuerdo al tipo de funcionalidad que estas tiene.

Modularización de los programas

Adicionalmente la clasificación de bloques que se estaría aplicando en el punto anterior es vital considerar las reglas de negocio que juegan un papel muy importante dentro de este proceso, por lo cual la parte de procedimientos de negocio deberá estar considerando una clasificación por bloques de acuerdo a la operación que este atendiendo cada pieza, a la fecha existen programas que atienden a más de un módulo de negocio lo que las vuelve más críticas. Poder lograr esto es muy sencillo, ya que este sistema ha logrado un nivel de madurez lo suficientemente robusto para obtener una definición clara y concreta de cómo debe estar operando cada módulo del sistema. Estas reglas no solo están plasmadas a nivel condigo sino que también existen manuales de operación que describen claramente la manera en el negocio de debe estar operando, de esta forma podemos decir que una exacta clasificación de módulos seria:

- Afiliación
- Retiros
- Traspasos
- Unificación de cuentas
- Separación de cuentas
- Aportaciones voluntarias
- Dispersión

Internamente en el sistema la clasificación de módulos se realiza utilizando dos datos conocidos como servicio y operación en donde:

Servicio = indica el modulo / Operación = indica el submódulo

Algunos ejemplos de los servicios/operación que opera el sistema son:

SERVICIO	OPERACIÓN
01 Afiliación	10, 15, 17
02 Traspasos	09, 10, 11
03 Aportaciones	27, 29, 30
04 Retiros	12, 13, 16
05 Transferencias	01, 02, 03
06 Separación	12, 13, 14
08 Unificación	09, 10, 11

Tabla 1. Clasificación por servicios internos AFORE BANAMEX recuperada de Ramírez (2009).

Documentación del programa

Al momento de ir generando las piezas de software se planea tener documentos que plasmen en ellos la funcionalidad que tiene cada pieza con el propósito de obtener una especie de manuales de cada uno de los programas los cuales indiquen y explique de manera clara y entendible que función tiene cada bloque de código, estos manuales además de información técnica consideraran el uso de diagramas y descripción a negocio a fin de que cualquier persona que revise ese documento conozca el funcionamiento de cada componente del sistema sin la necesidad de conocer el lenguaje o plataforma en la que están contruidos.

También es necesario generar en cada modificación de software, generar información que explique de manera detallada el cambio que se aplica a fin ir generando documentos históricos que permitan salvaguardar información detallada que explique las modificaciones a la funcionalidad de cada una de las piezas sufre con el pasar de los años, esto nos garantizará que el conocimiento que solo la gente con más experiencia en el negocio tiene no se pierda.

Es necesario que toda esta información se publique y no solo quede en manos de algunos como ocurre actualmente, se deberán crear repositorios que permitan a cualquier persona involucrada en el proceso de mantenimiento de software tener acceso a estos documentos.

Revisión a documentación del sistema

También es necesario una vez que se tiene toda la información referente a la funcionalidad realizar la revisión documentos que son generados comúnmente por el mismo personal que da soporte al sistema casi de sus inicios a fin de llevar a cabo un rediseño de la manera en que opera todo el sistema así como generar formatos más cómodos para ir creando una historia sobre los cambios que ira teniendo el sistema en requerimientos posteriores.

Algunos de los documentos que más relevantes son:

- Especificaciones técnicas

Aunque incompletas y desactualizadas existen documentos en los que se especifica la funcionalidad que tiene cada una de las piezas, pocas veces suelen actualizarse y no todas

las piezas de software cuentan con esta documentación, pero lo poco que existe podría ser revisado a fin de obtener un poco de información de la manera en cómo opera cada programa, basándonos en estos documentos se generarían propuestas para las nuevas piezas tratando de cuidar se migre totalmente su funcionalidad en piezas nuevas.

- Modelos de datos

Esto suele ser un complemento del documento anterior, pero especificando a detalle las características de las bases de datos que son utilizadas por cada programa y sus propiedades.

- Documentos de diseño técnico

Son los documentos fundamentales en cada proyecto de software en los que se plasman los requerimientos solicitados por el negocio de manera general, mediante diagramas y modelos conceptuales se generan propuestas sobre la funcionalidad nueva que deberá tener cada componente del sistema y brindar un diseño general de como estarían operando estos componentes.

- Código fuente

Programas respaldados en los repositorios oficiales del banco, a la fecha el sistema está compuesto por más de 2000 piezas y 13 bases de datos.

- Documentación del usuario (negocio)

Como manuales de operación, y circulares emitidas por la CONSAR que establecen los criterios bajo los que deberán estar operando todas las AFORES sin excepción.

- Documentación técnica

Manuales técnicos de pantallas, manuales de programación en los lenguajes más utilizados como son COBOL, ALGOL, DMSII y WFL.

- Diagramas funcionales

Estos diagramas se generan en cada proyecto y deben especificar la manera nueva de operar del módulo o módulos que se modifican en cada nuevo proyecto, debe mostrar el antes y el después y deben estar bien sustentados con base al documento de diseño técnico. A diferencia de los diagramas que se generan en la etapa de diseño normalmente se generan con detalles más técnicos y propios de la plataforma del sistema que cualquier persona del área de sistemas puede entender.

- Interfaces o pantallas actuales

El S336 se compone de 500 pantallas las cuales igual que el código fuente se encuentran respaldados en los repositorios del banco. Cabe mencionar que a pesar de ser dependencias diferentes (Banco BANAMEX y AFORE BANAMEX) comparten varios repositorios.

- Información del sistema (datos almacenados en el sistema)

Información histórica que aloja el sistema para ser analizada, esto podrá permitir la prevención de fallas y corrección de anomalías existentes en el sistema.

- Documentos de prueba (guiones de prueba y casos de prueba)

Son documentos en los que se almacenan las evidencias de pruebas de las piezas modificadas en cada proyecto, estos se generan en cada proyecto y son requeridos para realizar liberaciones al ambiente productivo sirviendo como antecedente de la correcta modificación y ejecución de los procesos.

Revisión de datos del sistema

Analizar la información que aloja el sistema para hacerla más comprensible. Este punto considera una reorganización de la información existente en las bases de datos del sistema, aplicar una clasificación por tipo de información y tipo de servicio que es atendido.

Para este punto se propone la aplicación de una reorganización de la información existente en el sistema, mejor conocido como reingeniería de datos.

Es muy común encontrarse con información alojada en estructuras de manera aleatoria, es decir que no mantiene un orden o clasificación por lo que al consultar en cualquiera de las estructuras del sistema uno puede darse cuenta de que existe información combinada que no tiene una consistencia o relación directa con el modulo que atiende, como ejemplo de lo que ocurre al día de hoy se tiene el siguiente diagrama.

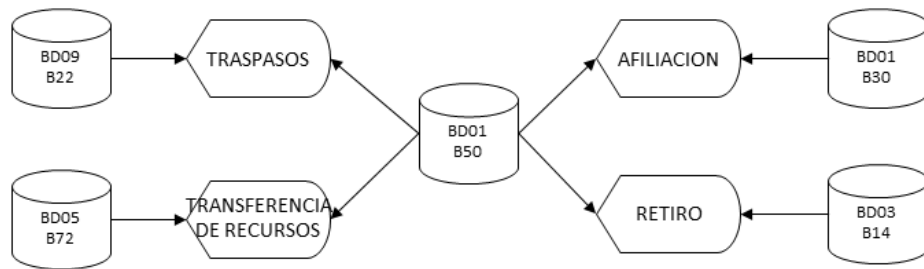


Figura 3. Esquema general de acceso a base de datos por modulo recuperada de Ramírez (2009).

En este diagrama se puede observar como distintos módulos del sistema comparten una tabla para alojar y obtener información que a nivel modulo no tiene ninguna relación entre sí como puede ser:

- Afiliación (Que considera datos personales del cliente)
 - Dirección
 - Teléfono
 - Email
 - Beneficiarios
 - Estatus del cliente

- Traspasos (Que considera lo relacionado a la información de las AFORE´s que están dentro de este proceso)
 - Datos de AFORE Cedente (Dirección, teléfono, entre otros)
 - Datos de AFORE Receptora (Dirección, teléfono, entre otros)
 - Saldo en AFORE Cedente

Perfil de inversión en AFORE Cedente

Perfil de inversión en AFORE Receptora

- Retiros

Saldo disponible del cliente

Información de retiros anteriores

Histórico de Movimientos

- Transferencia de recursos

SIEFORE

Subcuenta

Importe a transferir

Perfil de inversión

La idea principal de este punto es llevar un orden no sólo en la manera en la que operan los procesos si no en la información que estos procesan, aunque aparentemente este punto se estaría atendiendo al momento de corregir las piezas de software, no es así, ya que deberá corregirse toda la información histórica que ya se encuentra residente en las bases de datos actuales.

Al seguir esta serie de pasos de manera constante y utilizándolos de manera similar a una metodología de desarrollo se espera que en un periodo de tiempo no muy largo la mayoría de la

problemáticas mencionadas en este documento se hayan disipado, y el nuevo código generado facilite al área de sistemas a llevar a cabo su trabajo de manera más sencilla.

Lo idea principal de esta propuesta la generación de un documento formal que sea la guía para la gente que se va incorporando a los procesos de mantenimientos de sistemas legados, esto permitirá minimizar la herencia de fallos, errores y malos hábitos del personal involucrado en estos procesos, sin duda esta información puede ser de gran ayuda para los líderes de proyectos de software como guía a seguir ya que a pesar de solo ser una propuesta se espera que este trabajo logre ser la base para futuros desarrollos, que se enriquezca con la experiencia de otros y se logre tener como resultado una herramienta que aporten mecanismos que sean posibles de aplicar no solo al S336 si no que sean usados en cualquier sistema legado de los tantos que aún sobreviven.

Conclusiones

Con la aplicación de los puntos anteriores se logró notar una disminución considerable de fallas durante el proceso de mantenimiento de software al S336, se observaron proyectos más organizados y una disminución en el número de fallas reportadas al realizar la instalación de nuevas adecuaciones al sistema.

Por otra parte es importante mencionar que para llevar a cabo la correcta ejecución de estas propuestas fue necesaria la cooperación y trabajo en conjunto de todo el equipo involucrado en los procesos de mantenimiento al sistema S336 a fin de adaptarse a las nuevas prácticas propuestas lo cual fue bastante complicado debido a lo acostumbrado que estaba el personal a trabajar de manera tan desorganizada.

Este trabajo estuvo enfocado a atacar distintos fallos dentro del proceso de mantenimiento del sistema S336, sin embargo algunos mostraron mejoras más notorias dado que fueron los que tenían una mayor relevancia para el negocio, entre estos destacan:

- Datos e información histórica la cual se analizó, depuró y se generó nuevamente considerando únicamente información realmente necesaria, y a su vez eliminando información basura, esto permitió una ejecución más ágil de los procesos y una reducción considerable en el tiempo de procesamiento de la información.

- Reducción en malas prácticas de programación del personal desarrollo de software, esto debido a que anteriormente no se respetaban los estándares establecidos por el propio lenguaje ni por la institución en la que son operados estos sistemas.
- Estimaciones más reales lo que permite calcular tiempos de ejecución de los proyectos de software de manera más exacta, esto permite que las modificaciones al sistema se realicen con mayor calidad.
- Reducción de fallas en el diseño de algunas piezas de software lo que propician aún más fallas en otros componentes del sistemas como por ejemplo error en pantallas, archivos generados con inconsistencias o algunas de las veces incompletos, error en programas en tiempo de ejecución lo cual resulta bastante costoso para el negocio, entre otros.
- Documentación actualizada, clara y ordenada la cual permite tener antecedentes detallados sobre el manejo y función de los procesos.

De manera general se logró generar pasos a seguir más sencillos de realizar para todos aquellos que se ven involucrados en estos procedimientos administrativos a la hora de dar mantenimiento a fin de hacer más sencillas sus labores, sin embargo siguen existiendo actividades las cuales no pudieron ser adaptados a lo propuesto debido a que se consideran políticas obligatorias a seguir dentro de la organización.

Si bien el presente trabajo abordó algunas opciones sobre cómo mejorar las malas prácticas de programación al dar mantenimiento a sistemas legados es muy notoria la poca existencia de información sobre este tipo de tecnologías debido a que el interés para atender

estos sistemas cada vez es menor. En futuras publicaciones puede ser este un interesante tema a tratar no solo para los profesionales de las tecnologías de la información sino también para los auspiciantes que tienen por delante la necesidad de usar y conocer este tipo de herramientas.

Es claro que este proceso no será de la noche a la mañana debido a que se necesita un periodo de tiempo para que el personal involucrado en estos procesos se adapte al cambio. Es de suma importancia lograr que todos estos personajes se apeguen en su mayoría a las propuestas realizadas en este trabajo así como a los estándares que ya se encuentran establecidos para lograr que los procedimientos de mantenimiento de software se lleven a cabo de manera correcta, con un indicador de fallas mínimo, y en el menor tiempo posible siempre con la calidad que se requiere.

Es muy probable que conforme pase el tiempo y estas propuestas se vayan aplicando deban hacerse modificaciones o en su caso complementarse con puntos que vayan surgiendo. Es importante obtener por parte de gente con experiencia en este proceso ideas y comentarios que aporten mejoras que puedan acrecentar aún más la importancia de este trabajo a fin de abarcar aún más los posibles puntos débiles que puedan existir dentro de estos procesos.

Referencias

Antonio, V. (2011). *Procedimiento de análisis para sistemas heredados en la plataforma mainframe* (Tesis de maestría). Instituto Politécnico Nacional, Ciudad de México.

Trigo, V.(2012, 25 de Diciembre). Historia y evolución de los lenguajes de programación.

Revista ACTA. Recuperado de <http://www.acta.es/>

Banco Nacional de México. (2017). *Portal CitiBanamex*. Ciudad de México: Banamex.

Recuperado de <https://www.banamex.com>.

Bertolín, J. A. (2008). *Seguridad de la información. Redes, informática y sistemas de información*. Editorial Paraninfo.

Calderón, O. (2003). El cambio tecnológico y las nuevas formas de trabajo, el caso del centro de Cómputo de Banamex, una visión subjetiva de los trabajadores. En Contreras, F (Presidencia). *Cuarto Congreso Nacional de Estudios del Trabajo*. Ponencia llevada a cabo en el IV Congreso Nacional de Estudios el Trabajo, Hermosillo, Sonora.

De Dienheim, C., Zaragoza, K., Vizcaíno, T., Roa. E., Gutiérrez, M., Bahena, R. (2003, 08 de Enero). ¿Qué son las AFORE?. *Revista Jurídica IUS*. Recuperado de <http://www.unla.mx/>

Gutiérrez, A., Motz, R., Revello, B., & Silva, L. (2001). Construcción de un sistema de apoyo a la toma de decisiones para el área gerencial del Hospital de Clínicas. En Luna, D. & Garfi, L (Presidencia), *Informática y Salud*. En Simposio de Informática en Salud-Jornadas Argentinas de Informática e Investigación Operativa, Buenos Aires, Argentina.

Hualde, Alfredo, & Gomis, Redi. (2007). PYME de software en la frontera norte de México: desarrollo empresarial y construcción institucional de un cluster. *Problemas del desarrollo*, 38(150), 193-212.

Jaimes, N. (2007, 01 de Marzo). Conviviendo con sistemas legados. *Revista Electronica Paradigma*. Recuperado de <http://paradigma.uniandes.edu.co/>

Laudon, F., & Laudon, J. (1996). *Sistemas de Información*. Ciudad de México: Editorial Diana.

Procuraduría Federal del Consumidor. (2013). *Lo que debes saber de las AFORES*. Recuperado de http://www.profeco.gob.mx/encuesta/brujula/bruj_2013/bol248_Lo_que_debes_saber_sobre_afores.asp

Sommerville, I. (2005). *Ingeniería del Software* (7 ed.). Madrid, España: Pearson Education.

Stephens, D. (2008). *What on earth is a mainframe* (1 ed.). Australia: Longpela Expertise.

Teresa, C. (2011). *Metodología sistémica para solucionar problemas económico-financieros. Caso AFORES*. (Tesis doctoral). Instituto Politécnico Nacional, Ciudad de México.

Unisys DMSII. (2016, 8 de octubre). *En Wikipedia, la enciclopedia libre*. Recuperado de https://en.wikipedia.org/w/index.php?title=Unisys_DMSII&oldid=743172877.

Work Flow Language. (2016, 6 de enero). *En Wikipedia, la enciclopedia libre*. Recuperado de https://en.wikipedia.org/w/index.php?title=Work_Flow_Language&oldid=698520157.