

Instituto Tecnológico de Costa Rica

Departamento de Ingeniería en Computación

**“METODOLOGIA PARA EL MEJORAMIENTO DE LA CALIDAD DE
LOS SISTEMAS DE INFORMACION DE LA SUPERINTENDENCIA DE
PENSIONES (SUPEN)”**

**Informe de Proyecto de Graduación para optar por el grado de
maestría en Computación con énfasis en Sistemas de Información**

Ronald Araya

Cartago, Junio 2001

Resumen

Este trabajo tiene su sustento en la importancia que ha tomado para las organizaciones desarrollar sistemas de información que contemplen parámetros que permitan verificar la calidad del producto desarrollado, con el fin disminuir tiempos y costos de desarrollo y facilitar el proceso de mantenimiento de los sistemas.

Entrando en esta misma dinámica, por las funciones que realiza y las repercusiones de índole pública e incluso judicial que pudiera tener, para la SUPEN es de vital importancia desarrollar sistemas de información de calidad que permita realizar una adecuada supervisión de las organizaciones que manejan fondos de pensiones.

Para la realización de este trabajo, se realizó revisión bibliográfica y búsqueda de información en la red Internet. Además se revisaron documentos generados en la SUPEN, especialmente lo realizado con su estructura organizacional, funciones y la metodología de desarrollo de sistemas de información actualmente utilizada.

Posteriormente, se realizó un análisis de la metodología de desarrollo de sistemas actual con las normas internacionales de la ISO y de la IEEE en la parte relacionada con sistemas de información, para identificar debilidades o carencias de la metodología de desarrollo de la SUPEN. Asimismo, se aplicaron cuestionarios y entrevistas a los funcionarios de Informática y a los usuarios de los sistemas de información transaccionales, para identificar problemas en los sistemas actuales y en el proceso de desarrollo.

Los cuestionarios y entrevistas aplicadas han permitido identificar faltas de controles en la aplicación de la metodología, inexistencia de estudios de factibilidad, las pruebas únicamente permiten probar la funcionalidad del sistema, falta de

controles en la aplicación de la metodología, inexistencia de documentación externa y la inexistencia de estándares en la fase de desarrollo.

La metodología de desarrollo de sistemas actual de la SUPEN tiene una serie de debilidades, entre otras, son la generalidad de la metodología, la necesidad de establecer actividades precedentes y consecuentes claramente definidas, la necesidad de establecer responsables de cada actividad, la falta e integración entre los requerimientos iniciales y los productos finales y la falta de control en la aplicación de las etapas de la metodología.

Producto del análisis de la información recogida en los cuestionarios, entrevistas y análisis de la metodología de desarrollo actual de sistemas, se ha procedido a plantear una metodología de desarrollo de sistemas que contemple aspectos de calidad que vengán a reforzar las debilidades encontradas en la metodología actual.

El aporte principal de la metodología propuesta refuerza una serie de aspectos: la metodología establece actividades claramente definidas, se establecen responsables de actividades y productos que deben ser generados en cada etapa (esto con el fin de controlar el desarrollo de la actividad en sí misma), asimismo establece una actividad de control de calidad al final de cada etapa. Por otra parte, propone una fase de estudio de factibilidad, brinda criterios para la aplicación de pruebas no solo de funcionalidad, sino también de rendimiento, de interfaz y de integración. Además todas las fases de desarrollo están integradas, especialmente la fase definición de requerimientos- diseño y pruebas.

Además de la metodología propuesta, se realizan recomendaciones con el fin de mejorar otros aspectos que vengán a proveer mayor calidad a todo el proceso de desarrollo, tales como estimular la participación del usuario, su conocimiento de la metodología y su grado de responsabilidad con los proyectos, mejorar el proceso

post implantación de los sistemas, con el fin de obtener retroalimentación que permita ir generando métricas de desarrollo y que sirvan de base para que en los nuevos desarrollos se mejoren las debilidades encontradas, generar criterios de aceptación de los sistemas y finalmente, establecer en conjunto con la auditoría interna de la organización, estrategias de revisión de los sistemas en uso, con el fin de mejorar la calidad de los mismos.

Palabras Claves: Metodología; Sistemas de Información; Calidad

Summary

This study is based on the increased importance for organizations to develop information systems whose limits allow for quality control of the product being developed, with the idea of reducing time and development costs and making it easier to service and maintain these systems.

Bearing this in mind, due to the role it plays and the both public and legal repercussions which could arise, it is of vital importance for the SUPEN to develop top quality information systems which in turn will allow adequate supervision of the organizations which manage pension funds.

Bibliographical research as well as information obtained through Internet were used in order to complete this study. Moreover, documents generated within SUPEN were studied, especially those pertaining to its organizational structure, functions and the existing methodology for developing information systems.

Afterwards, an analysis was made of the current methods for systems development using the international standards of ISO and the IEEE in the area of information systems in order to identify weaknesses or deficiencies in the methods used by the SUPEN. Likewise, employees in the data processing department and those using the transactional information systems, were interviewed and asked to fill out questionnaires in order to spot problems in the systems currently in use and in the development process.

The interviews and questionnaires have revealed a lack of controls in the application of the methodology, the nonexistence of a feasibility study, that it is only possible to asses the operational aspect of the system, the absence of external documentation and the total lack of standards in the development stage.

The present method for developing systems in the SUPEN has a series of deficiencies amongst which are the generality of the methods employed, the need to establish clearly defined activities preceding and following the process, the need to appoint someone to be responsible for each activity, the lack of congruence between the initial requirements and the final product, and a lack of control in the carrying out of the various stages of the methodology.

As a result of the analysis of the information collected through the questionnaires, interviews and the analysis of the present method of developing systems, a methodology has been proposed which includes qualitative aspects which will re-enforce the weaknesses found in the method currently in use.

The main contribution of the proposed methodology strengthens a series of aspects: it establishes clearly defined activities, responsibilities are clearly established and results defined for each stage of the process (this with the sole purpose of controlling the activity itself), and in the same way it establishes a quality control activity at the end of each stage. Apart from this it proposes a feasibility study, provides criteria for testing not only functioning, but also performance, interphase and integration , especially in the defining stage of design requirements and testing.

Beyond the proposed methodology recommendations are made with the aim of improving other areas which may raise the standard of the overall process such as encouraging the participation of the end user, his or her knowledge of the methodology and their level of responsibility with projects in general, improving the post-implementation process of the systems with the idea of getting feedback which will allow us to generate ways of measuring development and which will serve as a basis upon which deficiencies can be reduced in new developments, finding criteria for selecting systems and, finally, in collaboration with the organization`s internal auditing, to implement strategies for the assessment of the systems currently in use

in order to improve their quality.

Key words: Methodology; Information Systems; Quality

Aprobación de Borrador de Tesis

**“METODOLOGIA PARA EL MEJORAMIENTO DE LA CALIDAD DE LOS
SISTEMAS DE INFORMACION DE LA SUPERINTENDENCIA DE PENSIONES
(SUPEN)”**

**Ing. Luis Montoya P, MSc.
Profesor Asesor (Visto Bueno)**

**Dr. Francisco Torres.
Director Maestría (Refrendo)**

A Dios Nuestro Señor , a mis padres y a mi esposa

Al señor Luis Montoya, por su dirección durante la realización del trabajo.
Asimismo, al Ing. Juan Carlos Siles, MSc. por su apoyo en la realización del trabajo y al personal de la SUPEN por su colaboración en la respuesta de los cuestionarios aplicados.

Finalmente, deseo agradecer a la Srta. Gabriela Gutiérrez y a la Sra. Gabriela Salazar por su apoyo en la recolección de material bibliográfico.

TABLA DE CONTENIDO

<u>CAPÍTULO I. INTRODUCCIÓN</u>	1
<u>1.1 INTRODUCCIÓN</u>	2
<u>1.2 OBJETIVOS</u>	3
<u>1.2.1 Objetivo General</u>	3
<u>1.2.2 Objetivos Específicos</u>	3
<u>1.3 ANTECEDENTES Y JUSTIFICACION</u>	4
<u>1.4 METODOLOGÍA DE TRABAJO</u>	9
<u>1.5 LIMITACIONES Y ALCANCES DEL PROYECTO</u>	10
<u>Capítulo II. Generalidades de la SUPEN</u>	11
<u>Perfil de la organización</u>	12
<u>Aspectos Informáticos</u>	20
<u>Capítulo III. Marco Teórico</u>	30
<u>Conceptos de calidad</u>	32
<u>Calidad Interna Vrs. Calidad Externa</u>	33
<u>Control de calidad</u>	37
<u>Garantía de calidad</u>	37
<u>Problemas que debe resolver el aseguramiento de la calidad</u>	40
<u>¿Cómo conseguir la calidad?</u>	40
<u>Metas del aseguramiento de la calidad</u>	41
<u>Beneficios del Aseguramiento de la Calidad</u>	42
<u>Funciones del aseguramiento de la calidad</u>	43
<u>El papel del grupo de aseguramiento de la calidad</u>	44
<u>El Plan de aseguramiento de la calidad</u>	45
<u>Actividades del aseguramiento de la calidad.</u>	48
<u>Relación del aseguramiento de calidad con las fases del ciclo de vida</u>	54
<u>Estándares de Calidad para el desarrollo de Software</u>	56

<u>Estándares de la ISO</u>	58
<u>ISO 9000</u>	59
<u>Información contenida en el estándar ISO 9000</u>	59
<u>Beneficios del ISO 9000</u>	60
<u>Problemas del ISO 9000</u>	61
<u>Estándares de la IEEE</u>	62
<u>Capítulo IV. Análisis de la metodología en SUPEN</u>	64
<u>Introducción.</u>	65
<u>4.1 Análisis de las fortalezas y debilidades de la metodología de desarrollo de sistemas de la SUPEN desde el punto de vista de control de calidad.</u>	66
<u>4.2 Comparación de metodología de desarrollo SUPEN con estándares de IEEE e ISO.</u>	69
4.2.1 <u>Comparación metodología desarrollo SUPEN con estándar IEEE.</u> ...	69
4.2.2 <u>Comparación de metodología desarrollo SUPEN con estándar ISO.</u>	75
4.2.2.1 <u>Acciones de calidad relacionadas con el desarrollo de sistemas.</u>	75
4.2.2.2 <u>Acciones de calidad relacionadas con el ciclo de vida de sistemas.</u>	77
<u>4.3 Análisis de sistemas transaccionales de la SUPEN.</u>	79
4.3.1 <u>Análisis de los hechos descritos por los analistas de la Intendencia Informática</u>	79
4.3.2 <u>Análisis de los hechos de los usuarios de los sistemas transaccionales de la SUPEN.</u>	83
<u>4.4 Conclusión.</u>	87
<u>Capítulo V. Propuesta de una metodología de desarrollo</u>	89
<u>Capítulo VI. Conclusiones y Recomendaciones</u>	125
<u>Conclusiones</u>	126
<u>Recomendaciones</u>	128
<u>Capítulo VII. Bibliografía.</u>	130

TABLA DE FIGURAS Y TABLAS

<u>Dibujo 1. Flujo de Información OPCs-SUPEN</u>	16
<u>Tabla 1. Requerimientos de Información de las Operadoras de Pensiones Complementarias</u>	18
<u>Tabla 2. Lista de productos generados en la SUPEN</u>	19
<u>Gráfico 1. Organigrama Informática SUPEN</u>	27
<u>Tabla 3. Responsables de cada actividad</u>	91
<u>Tabla 4. Actividades de Inicio del Proyecto</u>	94
<u>Tabla 5. Actividades de Definición de Requerimientos</u>	98
<u>Tabla 6. Actividades de Diseño.</u>	103
<u>Tabla 7. Actividades de Codificación.</u>	106
<u>Tabla 8. Actividades de Pruebas.</u>	111
<u>Tabla 9. Actividades de Implementación.</u>	115
<u>Tabla 10. Actividades de Post-Implementación.</u>	119
<u>Tabla 11. Actividades de Mantenimiento.</u>	122

CAPÍTULO I

INTRODUCCIÓN

1.1 INTRODUCCIÓN

El presente proyecto se orientó a la definición de una metodología de desarrollo de sistemas que considerara actividades de aseguramiento de la calidad para la Superintendencia de Pensiones (SUPEN), institución que se dedica a la fiscalización de las operadoras de pensiones, con el objeto de producir software de mayor calidad. Lo anterior a través de la aplicación de métodos, técnicas, estándares y normas, basados en estándares internacionalmente aceptados, como es la IEEE y ISO.

La metodología para el mejoramiento de la calidad de sistemas que resulte de este trabajo de investigación, beneficiará a la institución en la satisfacción del servicio al cliente interno y externo, reducirá el tiempo y costo de mantenimiento de los sistemas de información desarrollados y en la disminución de los recursos asignados a los sistemas.

El presente documento está constituido por las siguientes secciones:

- i. Objetivos generales y específicos,
- ii. Antecedentes y justificación,
- iii. Metodología de trabajo,
- iv. Limitaciones y alcances del proyecto,
- v. Generalidades de la SUPEN,
- vi. Marco teórico,
- vii. Análisis de la metodología de desarrollo de la SUPEN,
- viii. Análisis de las metodologías de la IEEE y de la ISO,
- ix. Propuesta de una nueva metodología de desarrollo para la SUPEN,
- x. Conclusiones y recomendaciones,
- xi. Bibliografía y Anexos del trabajo,

1.2 OBJETIVOS

1.2.1 Objetivo General

- Definir una metodología de desarrollo de sistemas de información para la SUPEN que permita obtener mejores productos de software.

1.2.2 Objetivos Específicos

- Analizar los estándares y metodologías de la ISO y de la IEEE respectivamente, que permitan identificar actividades de aseguramiento de calidad que permitan corregir los problemas identificados en las diferentes fases del desarrollo de sistemas de la SUPEN.
- Identificar los métodos de inspección, revisión y control de calidad que deben ser aplicados en las diferentes etapas del desarrollo de software en la SUPEN.
- Ajustar la metodología de sistemas de información de la SUPEN, integrando actividades de aseguramiento de calidad en los productos de software desarrollados.

1.3 ANTECEDENTES Y JUSTIFICACION

La calidad del software se ha vuelto muy importante; dado su gran uso, los defectos en el mismo pueden causar serios daños, principalmente si se trata de sistemas tan versátiles como por ejemplo: el software para el control de vuelos, tráfico aéreo y vehicular, procesos de fábricas, plantas de operación, sistemas financieros, entre otros. Por esta razón, los ingenieros de software deben tener presente que su trabajo puede afectar el desarrollo y el bienestar de muchas personas y organizaciones. Aunado a lo anterior, debe existir interés por asegurar la satisfacción de las necesidades de los clientes, construyendo entonces, productos de calidad que le procuren a cada institución mantenerse competitiva.

En muchas empresas, se reconoce que la alta calidad de un producto representa ahorro de costos de mantenimiento y de rediseño, por lo que aparte de una necesidad, la calidad del software es una exigencia de nuestras empresas de hoy, ya que se podrá contar con un software de mayor facilidad de uso, adaptación al crecimiento del negocio y de mantenimiento.

Es importante tener presente que, aunque es cierto que la fiabilidad del hardware y del software ha ido en aumento, este crecimiento no se ha dado de una forma proporcional entre ambos, lo anterior debido especialmente al aumento en la complejidad y la importancia que el usuario ha puesto en el último. El fenómeno ha originado lo que se conoce como “Crisis del software”. **[PRESS98, cap 1]**.

A pesar de los buenos deseos de las empresas, de los desarrolladores de software y de los usuarios para que los proyectos salgan a tiempo y con el costo proyectado, se presentan algunos de los siguientes problemas: los proyectos de software sobrepasan los presupuestos, se retrasan y hasta se abandonan,

inadecuada definición de los requerimientos iniciales, el seguimiento y control ligero a empresas externas, como vendedores, proveedores y subcontratistas, la alta rotación de personal, mal planeamiento, inadecuada supervisión del proceso y cambios incontrolados del producto.

Adicionalmente y según el criterio de Roger Pressman [**PRESS 98, cap 27**], el mantenimiento del software existente representa más del 60 por ciento de las inversiones efectuadas por una organización de desarrollo, y ese porcentaje sigue ascendiendo a medida que se produce más software.

Debido a los problemas expuestos en los párrafos anteriores, con los productos de software desarrollados por las empresas, se considera necesario establecer métodos o técnicas que permitan controlar la calidad de los sistemas que se desarrollan y su posterior mantenimiento.

Los problemas de control de calidad de software se ven durante su vida útil. Los sistemas con problemas de calidad se caracterizan por tener estructuras mal diseñadas, codificación pobre y lógica inadecuada, así como una documentación deficiente. Todas estas características se traducen en problemas de mantenimiento, ya que:

1. Se hace difícil seguir su evolución a través de las versiones, debido a que no están documentadas.
2. Es difícil seguir el proceso por el que se construyó el software.
3. Se dificulta comprender un programa “ajeno” si solo existe el código documentado.

4. El personaje “ajeno” por lo general, no se encuentra en la organización, ya que la rotación de personal informático es elevada.
5. No existe documentación apropiada, que sea comprensible y consistente con el código.

En general, los problemas que afligen al desarrollo del software se pueden caracterizar bajo muchas perspectivas diferentes, pero los responsables de los desarrollos del mismo se centran sobre los aspectos de “fondo”: **[PRESS 93, cap 1]**

6. La planificación y estimación de costos son frecuentemente muy imprecisas.
7. La productividad de la comunidad del software no corresponde con la demanda de sus servicios.
8. La calidad del software no llega a ser a veces ni siquiera aceptable. Se han experimentado graves desajustes en los costos y se ha distorsionado la planificación en meses o años. Se ha hecho muy poco para mejorar la productividad de los trabajadores del software. Los errores en los nuevos programas producen en los clientes insatisfacción y falta de confianza. Todos estos problemas manifiestan otras dificultades más:

- No hay tiempo de recoger datos sobre el proceso de desarrollo del software. Sin datos históricos como guía, la estimación no es buena y los resultados previstos son muy pobres. Sin una indicación sólida de la productividad, no podemos evaluar con precisión la eficacia de las nuevas

herramientas, técnicas o estándares.

- La insatisfacción del cliente con el sistema “terminado” se produce con mucha frecuencia. Los proyectos de desarrollo del software se inician, a menudo, con sólo una vaga indicación de los requisitos del cliente. Normalmente, la comunicación entre el cliente y el que desarrolla el software, es muy escasa.
- La calidad del software es normalmente cuestionada.
 - El software existente puede ser muy difícil de mantener. La tarea de mantenimiento del mismo se lleva la mayor parte de todo el dinero invertido en él. El mantenimiento no se ha considerado como un criterio importante en la aceptación del software.

Otro factor que agrava los problemas, es el hecho de que los ingenieros de software han tenido muy poco entrenamiento formal en las nuevas técnicas de desarrollo. Hay desigualdad en la manera de trabajar, algunas personas desarrollan un método ordenado y eficiente de desarrollo del software mediante prueba y error, pero muchos otros desarrollan malos hábitos que dan como resultado una pobre calidad y un difícil mantenimiento.

La actitud es un factor crítico, ya que muchos desarrolladores indican estar interesados en los productos de calidad; sin embargo, la realidad es que hay muy poca motivación para mejorarla y en el caso de productos específicos, muchos ingenieros argumentan que “es un buen producto, que por supuesto tiene errores, pero... todo software tiene pulgas. Además, es mejor que el de la competencia...”.

[CRO 79]

Por último, adicionado a esta problemática, se puede hablar de la resistencia al cambio, la tendencia de una cultura conservadora, que se manifiesta principalmente por la satisfacción con el nivel actual de calidad, el temor de perderlo por intentar lograr uno mejor y el poco entendimiento de otras culturas.

Muchos de los problemas asociados con la crisis del software pueden ser solucionados con métodos recomendados por la ingeniería del software, tales como estándares, guías, métricas, entre otros.

1.4 METODOLOGÍA DE TRABAJO

A continuación, se describe la forma como se desarrolló este trabajo.

Para la definición de la justificación y marco teórico se realizó una investigación bibliográfica, recopilando material de libros, revistas, boletines y sitios sobre la red Internet.

Con el fin de evaluar las debilidades de los sistemas y la aplicación de la actual metodología de desarrollo de sistemas, se aplicaron entrevistas y encuestas a los funcionarios informáticos y usuarios de las Intendencias. Los resultados de las encuestas Esta investigación sirvió de base para realizar un análisis de la calidad desde varias perspectivas, como por ejemplo el nivel de participación del usuario en el desarrollo del proyecto, conocimiento de la metodología de desarrollo, confiabilidad, rendimiento, usabilidad, nivel de satisfacción con el servicio.

Con toda esta información se procedió a realizar un análisis de todas las fases de la metodología de desarrollo, tomando como base de comparación los estándares de IEEE y de ISO. Además se retomo la información recabada de los analistas y de los usuarios de sistemas. Esto permitió mejorar los aspectos en los cuales se encontraron debilidades.

Con ese insumo se elaboro una propuesta de desarrollo y mantenimiento de sistemas de la SUPEN, incorporando actividades de aseguramiento de calidad.

1.5 LIMITACIONES Y ALCANCES DEL PROYECTO

Para el mejoramiento de la metodología de desarrollo se ha decidido adaptar dos (2) estándares (IEEE e ISO) que conformarán la técnica, notación, métodos y resolución de los problemas que se logren identificar del proceso de desarrollo del software de la SUPEN.

Se negoció con la Jefatura de la Intendencia Informática de la organización, el implementar la metodología producto del trabajo.

Capítulo II

Generalidades de la SUPEN

En esta sección se describe el perfil de la SUPEN, los objetivos de la organización, aspectos legales de la información manejada, información solicitada a los entes supervisados, información generada en la SUPEN y los aspectos informáticos de la SUPEN, que cubren tanto la mención de los objetivos y funciones de la Intendencia, los sistemas de información, la infraestructura de red y comunicaciones, la cultura informática, el personal y la metodología de desarrollo de sistemas.

Perfil de la organización

El régimen complementario voluntario de pensiones fue creado y regulado por la Ley 7523 de julio de 1995. Este régimen tiene como finalidad brindar a los beneficiarios protección complementaria ante los riesgos de la vejez y la muerte. El fondo constituido se invierte para el provecho primordial de los afiliados con el objetivo de contar con un ingreso adicional cuando llegue la etapa del retiro. El monto total de ese ahorro está compuesto por el capital aportado y los rendimientos derivados de la inversión de los recursos; por lo tanto es fundamental la buena administración de estos fondos.

El sistema nacional de pensiones está compuesto por los siguientes regímenes:

- Cuatro regímenes contributivos, donde se destaca el de Invalidez, Vejez y Muerte (IVM) administrado por la Caja Costarricense del Seguro Social,
- Dos regímenes no contributivos y,
- Planes complementarios de pensiones, que se dividen en:

⇒ Fondos creados mediante leyes especiales o convenciones colectivas, que amparan a los empleados de algunas instituciones públicas

y bancos (RECOPE, ICE, BNCR, BCR y BCAC) y,

⇒ Régimen complementario voluntario privado administrado por operadoras de planes de pensiones complementarias.

La Superintendencia de Pensiones tiene a su cargo la supervisión de los fondos complementarios de pensiones administrados por las Operadoras de Pensiones creadas al amparo de la Ley 7523, los sistemas complementarios creados por leyes especiales y el Régimen de Capitalización del Magisterio Nacional que forma parte de los regímenes contributivos del país.

Las labores fundamentales de esta entidad son fiscalizar el régimen y contar con indicadores de alerta que le permitan prevenir eventuales crisis en el sistema, con el fin de reducir el riesgo en el manejo de los fondos y consolidar el sistema, en función de su objetivo básico de protección del afiliado y los beneficiarios.

Con el fin de cumplir con esa labor la Superintendencia de Pensiones ha establecido las pautas de control necesarias que permitan alcanzar los niveles de eficacia, eficiencia y seguridad acordes con la misión de preservar los intereses de los afiliados y beneficiarios al régimen. Esas normas son los parámetros comunes para regular los aspectos centrales del sistema y dentro de ellos podemos citar: límites máximos de inversión por emisor e instrumento financiero, constitución y uso adecuado de las reservas de capital creadas, pautas para que la información suministrada a los afiliados y público en general sea correcta y oportuna y no induzca a error, políticas adecuadas de capital para los regulados, custodia de los títulos y potestad de dictar actos de suspensión o intervención de los regulados cuando sus acciones ponen en peligro el patrimonio de los clientes.

La Superintendencia, siempre con la idea de velar y apoyar en toda su extensión el desarrollo actual y futuro de la actividad, ha puesto gran énfasis en contar con

una tecnología informática de avanzada, de tal manera que permitan mejorar la calidad y la oportunidad en la supervisión, utilizando medios informáticos que le permitan automatizar los procesos de control de las operadoras de pensiones --- tales como por ejemplo dar la oportunidad a los supervisores de conectarse a la base de datos de SUPEN desde sus casas, o desde cualquier operadora utilizando el browser---, así como generar indicadores de alerta que permitan correcciones oportunas antes de que se presenten crisis en el sector.

Objetivos de la organización

- Velar por la seguridad de los recursos que administren los sistemas privados de pensiones.
- Regular y fiscalizar la inversión de los recursos de los fondos y velar por su seguridad.
- Establecer los requisitos y condiciones que deberán cumplir quienes administren o deseen administrar planes privados de pensiones.
- Autorizar los planes de pensiones, dictar normas, evaluar la solidez financiera de los fondos.

Aspectos Legales sobre la confidencialidad de la información

De acuerdo con la Ley 7523, Artículo 11, los funcionarios de las operadoras tienen total responsabilidad sobre la confidencialidad de la información que utilizan y no pueden utilizarla para provecho propio o para fines distintos de supervisión de los fondos. Además, ninguna información registrada en las cuentas individuales podrá ser suministrada a terceros, excepto los casos previstos en la Ley.

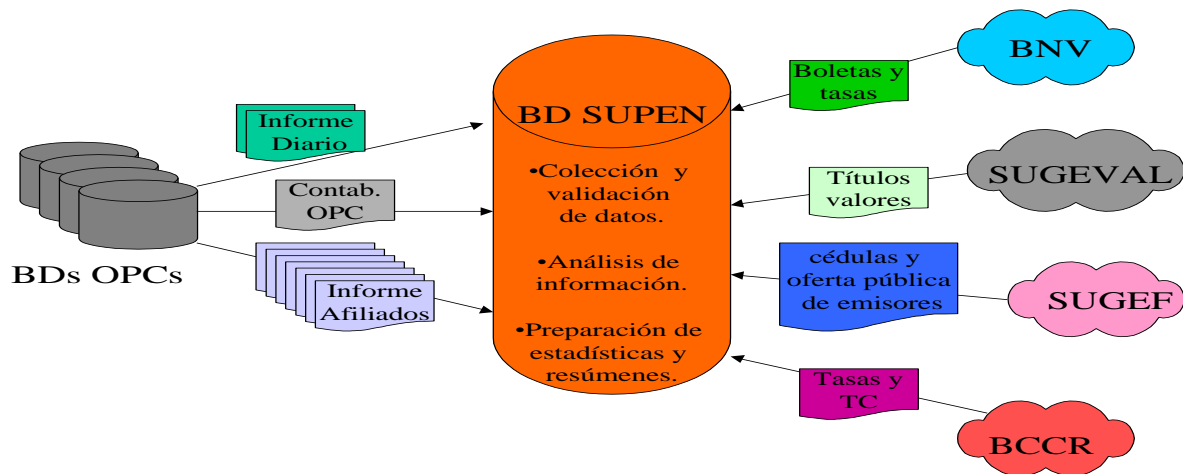
Para la Superintendencia de Pensiones, el Artículo 35, inciso e) declara que toda información requerida por el ente regulador será de interés público; sin embargo por norma constitucional existe un principio de confidencialidad de la información que proviene de documentos privados, la cual no podrá ser suministrada por el ente regulador.

La SUPEN cuenta con un Manual de Información que contiene las pautas teóricas e informáticas para el recibo electrónico de la información, donde se indican las características de como la información debe ser enviada . Este Manual está compuesto por cuatro capítulos básicos:

- Saldos contables del fondo
- Datos de afiliados
- Movimiento de Inversiones
- Información de la Operadora

En el dibujo adjunto, se presenta el flujo de información existente entre la SUPEN y otras entidades.

Flujo de información OPCs - SUPEN



Dibujo 1. Flujo de Información OPCs-SUPEN

En este dibujo, se puede observar que las operadoras de pensiones envían básicamente información de los movimientos diarios de inversión (base de datos SUPEN), información contable de las operadoras (validación de datos) y de los afiliados (preparación de estadísticas y resúmenes).

Pero además SUPEN requiere información de otras instituciones, como es el caso de las boletas y tasas enviadas por la Bolsa Nacional de Valores (base de datos SUPEN), información de títulos valores enviados por la Superintendencia General de Valores(SUGEVAL) (validación de datos), cédulas y oferta pública de emisores enviados por la Superintendencia General de Entidades Financieras (SUGEF) (análisis de información) e información de tasas y tipo de cambio enviado por el Banco Central de Costa Rica (BCCR) (preparación de estadísticas y resúmenes).

La tabla 1 muestra los requerimientos de la información de la SUPEN de parte de las operadoras de pensiones y la periodicidad con la que dicha información es enviada.

Requerimiento	Descripción	Periodicidad
Saldos Contables del Fondo	Contiene los saldos de las cuentas del fondo e información adicional del valor cuota y su composición. Por cada cuenta diferente de cero se informa un registro.	Diario
Movimientos de Inversiones	Detalla las inversiones que ha realizado la Operadora en un día específico y el formato con el que se solicita corresponde al de la Boleta de la transacción de la BNV. Cada movimiento de la cartera implica un registro que debe ser reportado.	Diario
Información de Afiliados	Se solicita la información de los afiliados bajo varios formatos, a saber: Datos personales del afiliado Contrato del Afiliado Datos de la empresa cotizante Datos del contrato del cotizante Beneficiarios del contrato del afiliado Datos sobre traspasos entre operadoras de un afiliado Movimientos de la cuenta individual de cada uno de los contratos del cotizante. Estos movimientos se piden desglosados por aporte personales, del cotizante, etc.	Mensual. Sin embargo, los movimientos se piden diarios.
Estados financieros de la Operadora	Contiene los saldos de las cuentas de la operadora como empresa.	Mensual
Datos Generales de la Operadora	Son los datos descriptivos de la empresa (nombre, cédula jurídica, fecha de constitución, socios, etc.)	Trimestral o cuando hay cambios
Datos de promotores	Datos personales de los promotores de planes, y su estado en cuanto a los exámenes y evaluaciones periódicas.	Trimestral o cuando hay cambios
Estados financieros auditados de las operadoras y de los fondos	Información financiera con el dictamen de un auditor externo	Anual

Requerimiento	Descripción	Periodicidad
Información fondos públicos complementarios de pensiones	Balance de Situación, Estado de Resultados, Composición de las inversiones por instrumento, emisor, plazo, afiliados, volumen del fondo, rentabilidad, jubilados	Mensual
Estudios Actuariales de los fondos públicos	Resultado del estudio actuarial (el cual debe estar actualizado) contratado por los administradores del Fondo	Más o menos cada dos años
Información del Régimen de Capitalización del Magisterio Nacional	Estados Financieros, rentabilidad, composición del portafolio de inversiones, estructura por emisor, instrumento y plazo de la cartera, número de afiliados, rentabilidad	Mensual
Información de los fondos complementarios de los bancos privados	Se está en etapa de diagnóstico de la situación de cada uno de esos fondos	Mensual (cuando se disponga)
Información sobre la custodia de los títulos valores	Se hace una inspección in situ en las custodias para verificar que el 95% de los títulos de cada fondo estén depositados	No tiene una frecuencia fija

Tabla 1. Requerimientos de Información de las Operadoras de Pensiones Complementarias

En la tabla 2 se muestra la lista de productos que se generan a nivel interno en la SUPEN con la información recibida.

Productos	Descripción	Periodicidad
Vector de Precios	Contiene la lista detallada de las inversiones transadas en Bolsa y el precio estándar calculado en la SUPEN que se utiliza para establecer la pérdida o ganancia de capital de la cartera.	Diario
Valoración del portafolio de inversiones de las Operadoras.	Es un listado con el detalle de la inversiones, su valor de mercado y contable.(Este cálculo se hace pero no se usa aún para la valoración)	Diario

Productos	Descripción	Periodicidad
Cuadro con desglose de la composición por instrumento y sector (público y privado) del portafolio inversiones	Contiene la estructura del portafolio, detallando el tipo de títulos y el sector.	Mensual
Cuadro con desglose de la estructura de vencimientos de la cartera del fondo de pensiones	Muestra por fondo las inversiones de corto, mediano y largo plazo	Mensual
Estados Financieros de las Operadoras de Pensiones	El balance de situación y el estado de resultados mensual de las administradoras de los fondos de pensiones	Mensual
Datos de los fondos públicos complementarios de pensiones	Balance de Situación y de Resultados. Además, número de afiliados y de jubilados	Mensual
Boletín Trimestral	Contiene un tema relevante para el campo de las pensiones y un anexo estadístico sobre el comportamiento durante el periodo de las operadoras y los fondos	Trimestral
Normativa	Reglamentos y directrices que regulan los distintos tópicos para una adecuada operación de las administradoras. Participación en elaboración de reglamentos que son de competencia de otros órganos reguladores y del banco central.	Continua
Publicación hoja web	Información general de la SUPEN, publicaciones y datos financieros de los fondos de pensiones administrados por las operadoras. Así como, publicaciones técnicas relativas al tema de pensiones.	Mensual y cada vez que hay cambios en la Normativa

Tabla 2. Lista de productos generados en la SUPEN

Aspectos Informáticos

El objetivo general de la Intendencia Informática es:

Proveer a la Superintendencia de sistemas de información y de las herramientas informáticas que apoyen su función de supervisión y fiscalización de las operadoras de fondos de pensión, así como asesorar a las áreas de la SUPEN y a las operadoras en aspectos informáticos.

Dentro de sus funciones básicas están:

- Asesorar a la SUPEN en las funciones de establecimiento de políticas, determinación de estrategias, aprobación de planes, determinación de prioridades de proyectos, adquisición de herramientas informáticas, entre otros.
- Desarrollar, mantener actualizado y ejecutar el plan de desarrollo de sistemas de la SUPEN.
- Desarrollar la automatización de oficinas de la SUPEN.
- Administrar los recursos computacionales de la SUPEN.
- Atender y resolver consultas a los usuarios de las diferentes herramientas computacionales con que se cuenta.

La Superintendencia de Pensiones tiene como objetivo básico ser una entidad automatizada, de tal manera que la mayor parte de los procesos se realicen en forma electrónica. Con esa finalidad tanto la Superintendencia como las operadoras han desarrollado sistemas y enlaces que permiten el flujo de información entre ellas.

Con el fin de conocer las bases técnicas de tal sistema, a continuación se ofrece un listado de los principales equipos y sistemas utilizados:

A lo interno de la SUPEN:

- Los servidores trabajan con NT.
- Las estaciones trabajan con Windows 95 y 98.
- El software de correo interno es Lotus Notes
- Las bases de datos son Oracle 8.
- Las herramientas de desarrollo son Developer 2000
- El software de oficina es Office 97

A lo externo de la SUPEN:

- Conexión con las operadoras a través de la red de la Bolsa Nacional de Valores (BNV).
- El software de correo con las operadoras es Lotus Notes.

Opciones de Comunicación:

- Conectarse a través del Banco Central. Aquí se utiliza la conexión actual que existe entre la Superintendencia de Pensiones y esa Institución.
- Conectarse a través de la red de la BNV, por medio de una línea de conexión a 64 Kbps.
- Conexión directa entre entes reguladores.

Sistema de información

El desarrollo de sistemas de información se fundamenta en las siguientes políticas establecidas por el Superintendente de Pensiones:

- Oficina pequeña. En la cual se mantenga el personal mínimo requerido con una capacidad profesional e intelectual capaz de atender múltiples tareas de análisis y control de los entes fiscalizados.
- Oficina cero papeles. Utilización de todos los recursos informáticos disponibles que permitan a la SUPEN eliminar al máximo el uso del papel como medio para remitir información entre los empleados de las institución y entre la SUPEN y las Operadoras.
- Comunicación en línea. En la SUPEN se busca establecer enlaces dedicados entre la institución y las entidades fiscalizadas, realizar todas las comunicaciones oficiales en forma electrónica y extraer los datos de los sistemas transacciones desde la fuente, de tal forma que no existan transformación o alteraciones de estos datos.
- Adquisición de hardware y software. Toda adquisición de hardware y software debe originarse en la Intendencia Informática. Las especificaciones técnicas y el proceso de evaluación de las ofertas buscarán incorporar la mejor tecnología, en calidad, respaldo y actualización.
- Actualización de equipos y software. La Superintendencia maneja la política de actualizar el 30% de su equipo de cómputo al año cuando éste sea obsoleto. El nivel de obsolescencia que se maneja es de 3 años para

el equipo de cómputo.

- No se desarrollarán aplicaciones para servicios administrativos. Las funciones administrativas de las entidades de desconcentración máxima SUPEN, SUGEF y SUGEVAL de acuerdo con la directriz del Consejo Nacional de Supervisión han sido asignadas al Banco Central. Por lo que no se justifica el uso de recursos de la SUPEN para el desarrollo de aplicaciones que provee el BCCR.

En la SUPEN se han desarrollado los sistemas de información para responder a los requerimientos que se han planteado de acuerdo a las políticas que se han dictado en la Institución. Básicamente existen tres áreas en las cuales se ha trabajado:

- Automatización de trámites internos.

En esta área se trata de automatizar el archivo y control de documentos, faxes, etc. Para lo cual se han desarrollado sistemas de información para automatización de oficinas que se listan a continuación:

- ⇒ Sistema de Normativa
- ⇒ Sistema de Discusión
- ⇒ Sistema de Promotores
- ⇒ Sistema de Control de Vehículos
- ⇒ Sistema de Inventario Mobiliario y Equipo
- ⇒ Sistema de Control de trámites V.2
- ⇒ Sistema de Operaciones \geq \$10000

- Automatización de trámites externos

Estos son sistemas que se han elaborado con el fin de automatizar la información que se maneja entre los entes supervisados y la SUPEN, los cuales se listan a continuación:

- ⇒ Sistema de Información de Operadoras
- ⇒ Sistema de Control Correspondencia Externa

- Sistemas de Apoyo a la Supervisión

La tercer área donde se ha determinado la necesidad de automatizar procesos, son los denominados sistemas de apoyo a la Supervisión, los cuales a diferencia de los anteriores sistemas se han desarrollado en ORACLE. Dichos sistemas son los siguientes:

- ⇒ Sistema de Control de Inversiones
 - ◆ Módulo de Informe Diario
 - ◆ Módulo de Precios
 - ◆ Módulo de Valoración
- ⇒ Sistemas de Afiliados

De estas aplicaciones todas están en funcionamiento y requieren de una actualización y mantenimiento constantes que demanda tiempo importante de los funcionarios de la Intendencia Informática.

Infraestructura de red y comunicaciones

Una de las herramientas mas importantes para cumplir con los objetivos de la SUPEN es la utilización de una infraestructura de red interna y externa que le permita atender con fluidez el intercambio de información que se desea realizar. A continuación se detalla la información relevante de esta infraestructura:

- Estaciones de trabajo, cada funcionario cuenta en su escritorio con una computadora, con un procesador Pentium II o superior y multimedios.
- Conexiones a red interna a 100 Mbps, en una red estructurada y certificada por ALCATEL.
- Servidores, se tiene la información distribuida de acuerdo a la actividad que se requiere en la SUPEN, para lo que se cuenta con

⇒ un servidor de Lotus Notes en donde se encuentra el correo electrónico y las bases de datos documentales

⇒ un servidor de Bases de datos transaccionales ORACLE, el cual se utiliza para guardar la información que remiten las entidades (en el futuro será extraída) que constituyen los datos de funcionamiento de la OPC

⇒ un servidor de IMÁGENES donde se mantienen los documentos escaneados que deben ingresar al sistema de trámites de la institución

⇒ un servidor de FAX donde se reciben y envían los faxes

⇒ un servidor de comunicaciones, el cual se utiliza para intercambiar datos con las Operadoras, así como, comunicaciones oficiales entre las dos instituciones

⇒ un servidor que sirve como muro de fuego, para monitorear los accesos a Internet y evitar el ingresos de elementos indeseables a la red de la SUPEN.

⇒ Equipo portátil para los inspectores de campo.

Cultura informática

Desde el inicio de la organización se planteó la necesidad de que la Superintendencia de Pensiones se constituyera en una oficina pequeña y con un gran apoyo en la tecnología para cumplir con sus objetivos en la supervisión del mercado de pensiones complementarias.

La mayoría de los funcionarios de la institución provienen de el Banco Central y la SUGEF, instituciones donde existe un nivel tecnológico bueno, pero que no han incursionado en conceptos como la oficina sin papeles, la extracción de datos vía remota, etc. Por esta razón, a todo el personal se ha dado una inducción en el uso de las herramientas informáticas que permiten desarrollar el concepto de oficina cero papeles y en la nueva organización que este tipo de tecnología provee.

Por otra parte, el Superintendente y la Intendente General velan prácticamente todo el tiempo porque se estén utilizando los recursos de la forma mas eficiente posible y que toda comunicación sea por medios electrónicos.

Personal de Informática

La estructura de la intendencia informática es como se presenta en el siguiente gráfico:

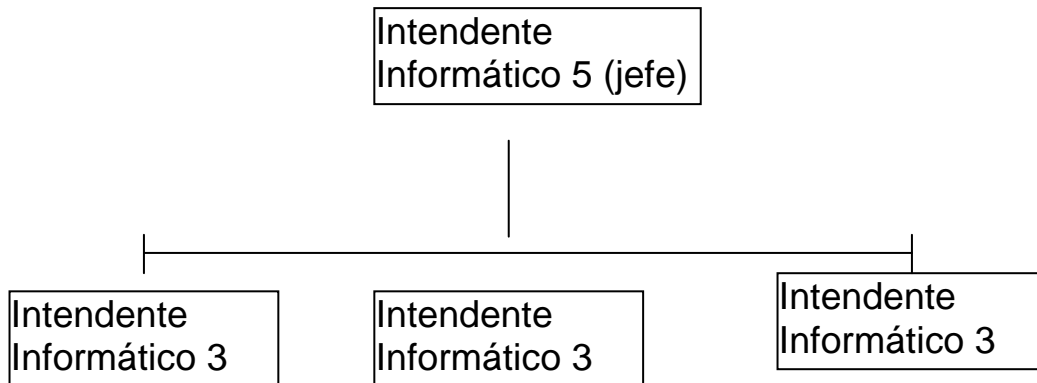


Gráfico 1. Organigrama Informática SUPEN

La Intendencia Informática está constituida por un jefe y tres analistas de sistemas que coordinan su trabajo de acuerdo a lo indicado por la jefatura. Los tres funcionarios que dependen de la Intendencia Informática tienen la misma categoría y realizan funciones que son variadas dependiendo del área al que son asignadas. Es así como la persona que se encarga de la administración de la Red de Datos (Oracle) también realiza actividades de desarrollo de sistemas por ejemplo.

Metodología de desarrollo de sistemas.

La metodología que existe data de Julio de 1996 y fue elaborada por un funcionario externo a la SUPEN. Cabe destacar que desde su creación hasta la fecha, no se le ha dado ningún tipo de mantenimiento. Dicha metodología está compuesta por cinco (5) etapas que a grandes rasgos son: **[SUPEN 96]**

- Etapa 1: Realización del estudio preliminar, en la cual, se determina la factibilidad y necesidad para la institución de realizar el sistema y se elabora el cronograma de actividades del proyecto.

- Etapa 2: Análisis de requerimientos del sistema. Se lleva a cabo el análisis de requerimientos de conformidad con el cronograma de actividades del proyecto, identificando adicionalmente usuarios de apoyo, programando entrevistas y concertando citas para entrevistas con estos usuarios.
- Etapa 3: Diseño del prototipo. En esta etapa, se planifica, se controla y se evalúa el diseño del prototipo conforme al cronograma del proyecto
- Etapa 4: Construcción e iteración del prototipo, donde se refina el prototipo, desde su planificación hasta su evaluación.
- Etapa 5: Implantación y pruebas del sistema. En esta última etapa, se planifica, ejecuta, controla, evalúa y aprueba la implantación y pruebas del sistema.

La metodología que se propone para el mejoramiento de la calidad de sistemas de la Intendencia Informática de la SUPEN, podría brindar los siguientes beneficios:

- El tiempo invertido en mantenimiento sería mucho menor; quedando más oportunidad para desarrollar aplicaciones que le den mayor valor agregado a la institución.
- Establecer una serie de pasos que permitan controlar el buen funcionamiento de los sistema (tanto a nivel de funcionalidad del sistema propiamente dicha,

así como facilitar el proceso de desarrollo del sistema), lo cual implicaría un mejor servicio a clientes y usuarios.

Adicionalmente se considera el contenido del proyecto como un aporte importante para guiar procesos de control de calidad similares, en otras instituciones del país que quizás no tienen claro cómo poder lograr mejoras en sus procesos de construcción de software.

Capítulo III

Marco Teórico

El marco teórico, tiene como objetivo, familiarizar al lector con el tema a desarrollar y el trabajo de investigación. Se incluyen conceptos generales de calidad, factores que la determinan y que pueden ser medidos directa e indirectamente, estableciendo una clasificación para los mismos, objetivos de la garantía de calidad, problemas que resuelve la misma, así como los beneficios inmediatos.

Adicionalmente, se explica la forma de conseguir la calidad, las metas y funciones del aseguramiento de la calidad. Con el fin de que se entienda cómo está estructurado el proceso de aseguramiento de la calidad del software, se detallarán las actividades y tareas que comprende, así como las herramientas y métodos técnicos. Dentro de la descripción del proceso, se indican todas aquellas amenazas para la calidad de los sistemas.

Como parte complementaria, se explicarán características importantes de un grupo de Aseguramiento de Calidad de Software (ACS), incluyendo su papel y funciones más importantes. Muy relacionado, también se encuentran los detalles de un plan de ACS.

En relación con los estándares de calidad, se explicará a nivel general qué es un estándar, cuáles son sus objetivos generales y razones para utilizarlos. Se hará referencia a algunos estándares del IEEE e ISO, que estén relacionados directamente con la calidad del desarrollo de software y se dará énfasis a aquellos que se van a adaptar en este proyecto.

Conceptos de calidad

En el ciclo de desarrollo de software, existen tres actividades que están sumamente involucradas: **[NASA-GB 96]** administración, ingeniería y aseguramiento. La administración del software es un conjunto de actividades en las que se incluye planificación, control y dirección del proyecto. La ingeniería de software es el conjunto de actividades que analiza requerimientos, desarrolla diseños, escribe código y estructura bases de datos. El aseguramiento del software garantiza que los esfuerzos de administración e ingeniería, dan como resultado un producto que satisface todos los requerimientos.

De acuerdo a **[PRESS 98, cap 8]**, todas las piezas fabricadas según un proceso de ingeniería exhiben algún nivel de variación. El control de la variación, es el centro de la calidad. Un fabricante quiere reducir la variación entre los productos que se fabrican. ¿De qué manera podría una organización de desarrollo de software necesitar controlar la variación? De un proyecto a otro, se quiere reducir la diferencia entre los recursos necesarios planificados para terminar un proyecto y los recursos reales utilizados, entre los que se incluyen personal, equipo y tiempo. En general, es indispensable asegurar que el programa de pruebas abarca un porcentaje conocido del software de una entrega a otra. No sólo se quiere reducir el número de defectos que se extraen para ese campo, sino también es importante asegurar que los errores ocultos también se reducen de una versión a otra. Es probable que los clientes se molesten si una tercera entrega de un producto contiene diez veces más defectos que la anterior. Lo realmente importante es reducir las diferencias en velocidad y precisión de las respuestas de soporte a los problemas de los clientes.

Calidad Interna Vrs. Calidad Externa

La calidad es la característica o atributo de algo. Como un atributo de un artículo, la calidad se refiere a las características medibles: cosas que se pueden comparar con estándares conocidos como longitud, color, propiedades eléctricas, maleabilidad y otras. Sin embargo, el software en su totalidad, como entidad intelectual, es más difícil de caracterizar que los objetos físicos, lo cual no implica que no existan características que permitan medir la calidad de un programa.

Existen dos tipos de factores que determinan la calidad del software: **[PRESS 93, cáp 17]**

- a. Los que se pueden medir **directamente**. Ejemplo: Errores por línea de código en una unidad de tiempo.
- b. Los que sólo se pueden medir **indirectamente**. Ejemplo: Facilidad de uso o de mantenimiento.

Estos factores se centran en tres aspectos:

- Operativos
- Del producto
- De transición

Operativos:

1. **Corrección:** es el grado en que el producto satisface las especificaciones y consigue los objetivos.

2. **Confiabilidad:** probabilidad de operar libre de fallos en un ambiente y

un período especificados.

3. **Eficiencia:** cantidad de recursos y código requerido para realizar funciones.

4. **Integridad:** grado en que se puede controlar el acceso al software o a los datos con personal no autorizado.

5. **Facilidad de uso:** esfuerzo para aprender a utilizarlo.

Del Producto:

1. **Facilidad de mantenimiento:** esfuerzo para localizar y corregir errores.

2. **Flexibilidad:** esfuerzo para modificar un programa.

3. **Facilidad de prueba:** esfuerzo para probar un programa asegurando su funcionalidad.

De transición:

1. **Portabilidad:** esfuerzo para transferir un programa de un ambiente a otro.

2. **Reusabilidad:** grado en que el programa o parte del programa se puede volver a utilizar en otra aplicación.

3. **Interoperabilidad:** esfuerzo para integrar o acoplar un sistema a otro.

Las características de medición varían de acuerdo al punto de vista de quien interprete la calidad, por lo que podemos hablar de calidad interna y calidad externa. La calidad interna está relacionada con el enfoque dado por desarrolladores y analistas de sistemas, y la calidad externa está ligada a la forma de cuestionarse la calidad, por parte de los usuarios finales de los sistemas y aplicaciones.

Para el caso de los usuarios, a ellos les interesa evaluar el grado en que el software afecta su trabajo. La calidad la miden en términos de facilidad de uso, confiabilidad del producto, correctitud, integridad y eficiencia.

Por otro lado, para una persona dedicada al desarrollo de software, la calidad se mide en términos de cohesión, números de puntos de función, líneas de código y muchas otras. Cuando se examina un componente, según las características que pueden ser medidas, se pueden encontrar dos tipos de calidad: calidad del diseño y calidad de concordancia. **[PRESS 98, cap 8]**

La calidad de concordancia se centra en la implementación. Si la implementación sigue el diseño y el sistema resultante cumple los objetivos de los requisitos y de rendimiento, la calidad de concordancia es alta.

Adicionalmente, la calidad del software es la ausencia de defectos, la conveniencia de uso, el alcanzar las especificaciones, la adherencia a los estándares, además de confiabilidad, facilidad de mantenimiento, portabilidad, facilidad de prueba, reusabilidad, correctitud, eficiencia, interoperabilidad y flexibilidad. **[PRESS 93, cap 17]**

A pesar de todo el énfasis que se hace en la gran importancia de la calidad,

algunos desarrolladores de software, continúan creyendo que la calidad es algo en lo que se empieza a preocupar una vez que se ha generado el código. La calidad se construye desde el comienzo del desarrollo, a través de los siguientes componentes:

- La gente que crea el producto.
- La tecnología que se usa para crearlo y
- Una buena disciplina administrativa para controlarlo, mediante un grupo de aseguramiento de la calidad, que busca la planificación y ejecución del programa a través de los tres componentes.

El ingeniero de software que escribe un programa debe ser capaz de saber encontrar y corregir sus defectos, en conjunto con otras personas. Sin embargo, es importante que no se piense que el aseguramiento de la calidad es sinónimo de pruebas. Cuando se tiene este punto de vista, sucede lo siguiente: **[PRESS 93, cap 17]**

- La gente prueba sus propios programas, lo cual dificulta el encontrar las “pulgas” (pequeños defectos o inconsistencias en el funcionamiento de una aplicación).
- Se tiende a pensar que las “pulgas” son una forma de vida independiente, y que por lo tanto es normal que existan en el software.
- Las pruebas se dan solamente una vez, al final de la fase de desarrollo.
- Las pruebas tratan solo con el producto, no con el proceso.

Es sumamente importante que los ingenieros de software tomen la responsabilidad de preocuparse por la calidad de los sistemas que producen. Escribir programas libres de errores, es todo un reto y requiere de métodos efectivos,

herramientas, práctica y datos. De ahí, la importancia de prácticas adecuadas y cuidadosas de ingeniería de software.

Control de calidad

Consiste en medir la calidad de un producto. El control de cambios puede equipararse al control de calidad. El control de calidad consiste de una serie de inspecciones, revisiones y pruebas utilizados a lo largo del ciclo de desarrollo para asegurar que cada producto cumple con los requisitos que le han sido asignados. Este control se lleva a cabo mediante la puesta en marcha de un plan de Verificación y Validación que permita afinar el proceso cuando los productos de trabajo creados fallan al cumplir sus especificaciones. Este enfoque ve al control de calidad como parte del proceso de fabricación.

Las actividades de control de calidad pueden ser manuales, completamente automáticas o una combinación de éstas últimas con la interacción humana. Un concepto clave del control de calidad es que se hayan definido todos los productos y las especificaciones medibles, en las que se pueden comparar los resultados de cada proceso. La retroalimentación es esencial para reducir los defectos producidos.

Garantía de calidad

La garantía de calidad es una actividad esencial en cualquier empresa que elabora productos que van a ser usados por otros. Consiste en medir la calidad de los procesos que son usados para crear un producto. Es importante, ya que en la industria del software, un producto de calidad puede bajar los costos de soporte técnico y elevar los niveles de satisfacción del cliente. Una pequeña inversión en aseguramiento de la calidad del software, puede actualmente ofrecer mayores

dividendos, incluyendo menores costos por mantenimiento y desarrollo cuando se requiere mejorar el producto.

Antes del siglo 20, la garantía de calidad era responsabilidad única de la persona que construía el producto. La primera función de control y de garantía de calidad formal, fue introducida por los laboratorios Bell en 1916 y se extendió rápidamente por toda la industria. **[PRESS 98, cap 8]**

Durante los primeros años de la informática (años 50 y 60) la calidad era responsabilidad únicamente del programador. Durante los años 60 se introdujeron estándares de calidad para el software en los contratos militares para desarrollo de software y se han extendido rápidamente a los desarrollos de software en el mundo comercial. Hoy en día, todos tienen responsabilidad: desarrolladores, jefes de proyecto, clientes y miembros del grupo de aseguramiento de la calidad.

El aseguramiento de la calidad es un diseño de acciones planificado y sistemático, aplicadas a través de todo el proceso de desarrollo de sistemas. Se requiere para asegurar que los procesos y productos de software, están contruidos conforme los requerimientos, estándares y procedimientos. Los “procesos” incluyen todas las actividades involucradas en el diseño, desarrollo, mejoras y mantenimiento del software; los productos incluyen el software, datos asociados, documentación y todos aquellos documentos de reportes y mantenimiento. **[PRESS 98, cap 8]** Hoy en día, la implicación es que muchos de los que constituyen una organización tienen responsabilidad de garantía de calidad del software (ingenieros de sistemas, jefes de proyectos, clientes, vendedores y aquellas personas que trabajan dentro de un grupo de aseguramiento de la calidad).

El objetivo de la garantía de calidad es proporcionar la gestión para informar de los datos necesarios sobre la calidad del producto, por lo que se va adquiriendo una

visión más profunda y segura de que la calidad del mismo está cumpliendo sus objetivos. Por supuesto, si los datos proporcionados mediante la garantía de calidad identifican problemas, es responsabilidad de la gestión afrontar los problemas y aplicar los recursos necesarios para resolverlos.

El grupo de aseguramiento de la calidad sirve como representación del cliente en casa. Es decir, la gente que lleva a cabo el aseguramiento, debe mirar el software desde el punto de vista del cliente, con el fin de determinar si la aplicación logra satisfacer de forma adecuada los factores de calidad y si se ha realizado el desarrollo de la misma de acuerdo con los estándares preestablecidos.

Adicionalmente se debe hacer hincapié en tres puntos importantes: **[PRESS 98, cap 8]**

1. Los requerimientos del software son la base de las medidas de calidad. La falta de concordancia con los requerimientos es una falta de calidad.
2. Los estándares especificados definen un conjunto de criterios de desarrollo que guían la forma en que se aplica la ingeniería del software. Si no se siguen estos criterios, casi siempre habrá falta de calidad.
3. Existe un conjunto de requerimientos implícitos que a menudo no se mencionan (por ejemplo, el deseo de un buen mantenimiento). Si el software se ajusta a sus requerimientos explícitos pero falla en alcanzar los requisitos implícitos, su calidad quedará en entredicho.

El aseguramiento de la calidad del software involucra: **[PRESS 93, cap 17]**

- Metodologías y herramientas de análisis, diseño, codificación y pruebas.
- Revisiones técnicas formales que se aplican durante cada fase del ciclo de vida.
- Estrategias para la realización de pruebas.
- Procedimiento que asegure ajustes a los estándares cuando sea necesario.

Como punto clave, se debe procurar que todos los productos y especificaciones estén definidos para que se puedan comparar con los resultados de cada proceso.

Problemas que debe resolver el aseguramiento de la calidad

El aseguramiento de la calidad, busca:

- Minimizar el número de defectos en el software entregado.
- Crear mecanismos para controlar su desarrollo y su mantenimiento.
- Asegurar que el producto va a ser utilizado.
- Mejorar la calidad de las versiones y de los productos futuros.

¿Cómo conseguir la calidad?

A pesar de los esfuerzos de muchas instituciones por conformar una organización formal para el aseguramiento de la calidad, su creación no es suficiente. Debe procurarse cambiar la cultura informática de la empresa. Para lograr esto, es necesario: **[YOURDON 89, cap 8]**

- Crear un propósito constante de mejorar los sistemas y el servicio.
- Adoptar una filosofía de calidad.
- Constantemente mejorar el proceso de desarrollo.
- Institucionalizar la capacitación, de tal manera que sea una actividad accesible para todos los que la requieran.
 - Manejar el temor para trabajar efectivamente.
 - Eliminar las barreras entre las diferentes áreas, procurando el trabajo en equipo.
 - Eliminar la tendencia a quedarse en palabras y no concretizar aquellas acciones que se hayan establecido en la organización, que permitan construir calidad.
 - Remover la barrera de orgullo de destrezas, delegando a todos la tarea de realizar el cambio.

Metas del aseguramiento de la calidad

El aseguramiento de la calidad y su grupo, procuran: **[YOURDON 89, cap 8]**

- Mejorar la calidad de las aplicaciones, a través de un monitoreo tanto del software como del proceso de desarrollo que lo produce.
 - Asegurar que los estándares y los procedimientos establecidos se estén cumpliendo totalmente.
 - Asegurar que cualquier anomalía en el producto, el proceso o los estándares sea atendido por la administración oportunamente.
 - Revisar que estén completos los planes de desarrollo y de calidad, promoviendo la anticipación del riesgo y la preparación y activación adecuada de los planes de contingencia.
 - Procurar mejoras en la comunicación e interacción de los equipos de

desarrollo.

- Identificar y documentar los puntos críticos del proceso de desarrollo.
- Participar como moderadores en las inspecciones de diseño y de código y en todos los puntos críticos del proceso de desarrollo, para asegurar que los malentendidos y errores sean detectados lo más tempranamente posible, antes de que provoquen retrabajo, entregas tardías y cansancio organizacional.
- Revisar que los planes de pruebas se adhieran a los estándares.
- Auditar periódicamente la configuración del software, para proteger la integridad del producto, reduciendo el riesgo de esfuerzos en vano y mejorar a su vez la productividad.
- Revisar que una muestra significativa de los resultados de las pruebas se adhieran a los planes.
- Participar en las revisiones para verificar si los estándares y procedimientos se siguen, documentando adecuadamente todos los desacuerdos.

Beneficios del Aseguramiento de la Calidad

Entre los principales aspectos positivos de realizar aseguramiento de la calidad se tienen:

[PRESS 93, cap 17]

- El software tendrá menos defectos latentes, resultando en menor esfuerzo y menor tiempo en prueba y mantenimiento.
- Da mayor confiabilidad y satisfacción al usuario.
- Se pueden reducir costos de mantenimiento.
- Disminuye el costo del ciclo de vida total del software.

Funciones del aseguramiento de la calidad

Al establecer las funciones del aseguramiento de la calidad, debe incluirse lo siguiente:

[YOURDON 89, cap 8]

- Prácticas de aseguramiento de la calidad, como herramientas de desarrollo, técnicas, métodos y estándares que puedan luego ser utilizados como estándares de revisión.
- Evaluación de la planificación de proyectos, para que puedan ser implementados adecuadamente.
- Evaluación de requerimientos, para verificar que los que se establecen inicialmente, cumplan los estándares, ya que es difícil desarrollar productos de alta calidad con requerimientos de baja calidad.
- Evaluación del proceso de diseño, para determinar si se sigue la metodología planeada y se implementan los requerimientos.
- Evaluación de las prácticas de codificación.
- Evaluación del proceso de pruebas e integración del software, preferiblemente por personas independientes al área de desarrollo.
- Evaluación de la administración y del proceso de control, asegurando que el proceso administrativo está trabajando y por lo tanto se asegure que la organización como un todo, está enfocada en producir resultados de calidad.
- Adaptación de los procedimientos de calidad a las necesidades únicas de cada proyecto.

El papel del grupo de aseguramiento de la calidad

En general los responsables de la calidad, son los grupos de desarrollo. El papel del grupo de aseguramiento, es monitorear la forma como esos grupos realizan sus tareas para cumplir sus responsabilidades. Debe trabajar desde las fases iniciales, con el propósito de establecer planes, estándares y procesos que puedan satisfacer los requerimientos, tanto en el ámbito organizacional como del proyecto, agregando así, más valor al mismo. **[PRESS 93, cap 17]**

Debe quedar claro, que este grupo no puede por sí solo, realizar algo por la calidad y la existencia del mismo, no asegura el uso de estándares y procedimientos.

En cuanto a la efectividad de su trabajo y recomendaciones, si no existe un compromiso demostrable hacia el grupo, por parte de la administración, será muy probable que las observaciones no sean tomadas en cuenta.

Para que sea efectiva la labor del grupo de aseguramiento, se deben presentar las siguientes condiciones: **[PRESS 93, cap 17]**

- Reportar a través de una cadena administrativa independiente.
- Estar asesorados por profesionales competentes.
- Procurar que el personal de desarrollo y mantenimiento, vea el trabajo como un apoyo a sus tareas, que permitan el mejoramiento de la calidad de los productos.

Una vez que han sido establecidos los planes, estándares y procesos, debe calendarizarse una revisión que permita verificar que se está dando realmente el aseguramiento.

El Plan de aseguramiento de la calidad

Como lo plantea Pressman en **[PRESS 98, cap 8]**, el plan de aseguramiento de la calidad, proporciona un mapa para institucionalizar la garantía de calidad del software. El plan, desarrollado por un grupo de aseguramiento y el equipo del proyecto, sirve como plantilla de actividades para cada proyecto de software.

Cada proyecto ya sea de desarrollo o mantenimiento, debe basarse en un plan de aseguramiento de la calidad, que especifique los objetivos y las tareas a desarrollar, así como los estándares contra los cuales se va a medir el trabajo, los procedimientos, las herramientas, las metodologías y la estructura organizacional. Adicionalmente, debe incorporar procedimientos para identificar, recolectar, llenar y dar mantenimiento a los registros acerca de lo encontrado en materia de calidad.

En el Plan de aseguramiento de la calidad, debe especificarse los cronogramas de todas las actividades y la revisión de las tareas de ingeniería (desarrollo, operación, mantenimiento entre otras) para determinar su ajuste a los estándares, planes, procesos y requerimientos de usuarios.

El planeamiento de todas las actividades que se realizarán con el fin de buscar y asegurar la calidad en el software, tiene como objetivo principal el lograr que se adopten estrictos estándares de desarrollo para asegurar calidad en las fases de su desarrollo.

Si no se ejerce un control sobre estas situaciones, suceden muchos problemas. Se ha comprobado que las dificultades se agravan cuando no se utiliza la definición correcta de calidad porque se han tomado en cuenta todas las características técnicas, pero no se han tomado las características sugeridas (en cuanto interfaz,

funcionalidad, facilidad de uso u otro) por el usuario en el plan de calidad. De ahí, que una adecuada metodología debe contemplar lo siguiente:

- Incluir siempre el punto de vista del usuario en el plan de aseguramiento de la calidad.
- Dar al usuario un papel más activo en el proceso de desarrollo.
 - Incluir ajustes en la fase de requerimientos, de tal manera que los usuarios pueden estar activamente involucrados, en la formulación del diseño y en el proceso de toma de decisiones.

Cuidando estos aspectos se pueden lograr muy buenos resultados como:

- Corrección de defectos importantes.
- Economía de dinero y tiempo en muchos procesos.
- Reducción de costos de mantenimiento.
- Mejor especificación de requerimientos por participación del usuario desde el comienzo.
- Mayor satisfacción del usuario.
- Nuevas maneras de construir software.
- Elevar la motivación en el personal de desarrollo.

Probablemente durante las revisiones, se encuentren incumplimientos. Esto debe tener un procedimiento eficaz que permita darle solución a las desviaciones al nivel adecuado. Estas soluciones a su vez, también deben ser monitoreadas para garantizar que realmente los problemas fueron superados. El primer paso para establecer un plan de aseguramiento de la calidad en una organización, es cerciorarse de que la administración superior está de acuerdo con las metas, con el

fin de evitar que surjan conflictos entre la gente de aseguramiento de la calidad y la administración.

Al iniciar el plan de aseguramiento, se debe realizar lo siguiente: [HUMP 95, cap 8]

- Definir papeles claves y buscar el compromiso público de la administración.
- Documentar metas y responsabilidades.
- Identificar líderes.

Una vez que se ha hecho lo anterior, debe identificarse los problemas claves relacionados con la calidad, los cuales deben ser atendidos de inmediato, con el fin de pasar a la etapa de escribir el plan.

Al escribir el plan de aseguramiento, se busca definir las actividades de auditoría y control, identificando estándares y procedimientos, los cuales deben ser revisados y aprobados. Seguidamente, debe establecerse las funciones del grupo de aseguramiento, que ayuden a cumplir el plan establecido.

Posteriormente, debe haber una preocupación por conducir una capacitación adecuada sobre los métodos de aseguramiento y lograr promover el plan de aseguramiento, mediante reuniones y revisiones para que el personal del proyecto conozca los objetivos del plan. Con esto podría entonces iniciarse la implementación del mismo. Para ello, debe procurarse que:

- Cada actividad clave sea asignada a una persona específica.
- Se desarrolle un cronograma adecuado.
- Se implemente un sistema de resolución de problemas.

Cuando se implemente el plan, es recomendable que se ejecute una evaluación del programa, auditando todas las funciones para determinar su efectividad para alcanzar su meta. En caso de encontrar problemas, se deben identificar acciones correctivas e implementarlas.

Actividades del aseguramiento de la calidad.

Existen siete actividades principales para el aseguramiento de la calidad: [PRESS 93, cap 17]

1. Aplicación de métodos técnicos.
2. Realización de revisiones técnicas formales.
3. Prueba del software.
4. Ajuste a los estándares.
5. Control de cambios.
6. Mediciones.
7. Registro y realización de informes.

1. Aplicación de métodos técnicos: el aseguramiento de la calidad, comienza realmente con un conjunto de herramientas y métodos técnicos que ayudan al analista a conseguir una especificación y un diseño de alta calidad.

El planificador aplica técnicas y herramientas para calcular estimaciones de esfuerzo y de tiempos, basándose en la comprensión de los requerimientos funcionales del software, en las características de rendimiento, en las restricciones del sistema y en los aspectos de fiabilidad.

[PRESS 98, cap 11] Ciertos investigadores independientes, han desarrollado un

enfoque orientado al equipo para la recopilación de requerimientos, que se aplica durante las primeras etapas de análisis y especificación. Este enfoque, denominado *Técnica para Facilitar la Especificación de la Aplicación* (TFEA), comprende la creación de un equipo mixto de clientes y personas encargadas del desarrollo que trabajan juntos para identificar el problema, proponer elementos de solución, evaluar diferentes enfoques y especificar un conjunto preliminar de requerimientos de la solución.

Se han propuesto muchos enfoques diferentes para el método TFEA, pero todos aplican alguna variación de las siguientes directrices básicas:

- a. Se lleva a cabo una reunión en un lugar neutral, a la que asisten tanto técnicos como clientes.
- b. Se establecen reglas para la preparación y participación.
- c. Se sugiere una agenda que sea lo suficientemente formal para que cubra todos los puntos importantes, pero lo suficientemente informal para estimular el flujo libre de ideas.
- d. Se acuerda la elección de un “facilitador” (puede ser el cliente, el técnico o alguien ajeno) para controlar la reunión.
- e. Se utiliza un “mecanismo de definición” (hojas de trabajo, diagramas, pizarras o tableros).
- f. El objetivo es identificar el problema, proponer elementos de una solución, evaluar los diferentes enfoques y especificar un conjunto preliminar de requerimientos de la solución en una atmósfera adecuada para alcanzar el objetivo.

2. Revisiones técnicas formales: una vez que se ha creado una especificación (o prototipo) y un diseño, debe ser garantizada su calidad mediante una revisión técnica formal. La revisión técnica formal (RTF) es una especie de reunión del

personal técnico con el único propósito de descubrir problemas de calidad.

Los objetivos de la RTF son:

- a. Descubrir errores en la función, la lógica o la implementación de cualquier representación del software.
- b. Verificar que el software bajo revisión alcanza sus requerimientos.
- c. Garantizar que el software ha sido representado de acuerdo con ciertos estándares predefinidos.
- d. Conseguir un software desarrollado de forma uniforme y hacer que los proyectos sean más manejables.

La RTF es realmente una clase de revisión que incluye recorridos, inspecciones, torneos de revisiones y otras tareas de revisión técnica del software. Cada RTF se lleva a cabo mediante una reunión y sólo tendrá éxito si es bien planificada, controlada y atendida.

Cualquier reunión de revisión debe acogerse a las siguientes restricciones:

- Deben convocarse a la reunión normalmente entre tres y cinco personas.
- Cada participante se debe preparar por adelantado, pero sin que requiera más de dos horas de trabajo a cada uno.
- La duración de la reunión de revisión debe ser menor de dos horas.

La reunión de revisión es llevada a cabo por el jefe de revisión, los revisores y el productor (desarrollador del producto). Uno de los revisores toma el papel de registrador, es decir, registra de forma escrita todos los sucesos importantes que se produzcan durante la revisión. El productor procede con el “recorrido de

inspección” del producto (explica el material), mientras que los revisores exponen sus dificultades, basándose en su preparación previa. Cuando se descubren problemas o errores válidos, el escriba los va anotando.

Al final de la revisión, todos los participantes en la RTF deben decidir si (1) aceptan el producto sin posteriores modificaciones; (2) rechazan el producto debido a los serios errores encontrados, que una vez corregidos debe hacerse otra revisión o (3) aceptan el producto provisionalmente, es decir, se han encontrado errores menores que deben ser corregidos, pero sin necesidad de otra revisión posterior. Una vez tomada la decisión, todos los involucrados firman para indicar que han participado en la revisión y que están de acuerdo con las conclusiones del equipo de revisión.

3. Prueba del software: combina una estrategia de múltiples pasos con una serie de métodos de diseño de casos de prueba que ayudan a asegurar una efectiva detección de errores. La prueba del software es un elemento crítico para la garantía de calidad del mismo y representa una revisión final de las especificaciones, del diseño y de la codificación.

El principal objetivo del diseño de casos de prueba, de acuerdo con la opinión de Pressman, **[PRESS 98, cap 16]**, es derivar un conjunto de éstas que tengan la mayor probabilidad de descubrir los defectos del software. Para llevar a cabo este objetivo, se usan dos categorías diferentes de técnicas: prueba de la caja blanca y prueba de la caja negra.

A menudo, los experimentados realizadores de software dicen que “... la prueba nunca termina, simplemente se transfiere del encargado del desarrollo al cliente. Cada vez que el cliente usa el programa, lleva a cabo una prueba...”. Aplicando el diseño de casos de prueba, el ingeniero de software puede conseguir una prueba

más completa y descubrir y corregir así el mayor número de errores antes de que comiencen las “pruebas del cliente”.

4 . **Ajuste a los estándares:** el grado de aplicación de procedimientos y estándares en el proceso de la ingeniería del software varía de una empresa a otra. En muchos casos, los estándares vienen dados por los clientes o mandatos por regulación. En otras situaciones, los estándares se imponen por sí solos. Si existen estándares formales, se debe establecer una actividad de ACS para garantizar que se siguen. La garantía de seguimiento de estándares puede ser llevada a cabo por los encargados del desarrollo del software como parte de una revisión técnica formal o en situaciones en que se requiera una verificación del seguimiento independiente, por el grupo de ACS mediante su propia auditoría.

5. **Control de cambios:** los cambios son una de las principales amenazas para la calidad del software. Cada cambio realizado sobre el software puede introducir errores o crear efectos laterales que propaguen errores. El proceso de control de cambios contribuye directamente a la calidad del software, al formalizar las peticiones de cambio, evaluar su naturaleza y controlar su impacto. Este proceso comienza con una petición de cambio, lleva a una decisión de proseguir o no y culmina con una actualización controlada del elemento de configuración de software que se ha de modificar.

6. **Mediciones:** un objetivo importante de ACS es seguir la pista a la calidad del software y evaluar el impacto de los cambios de metodología y de procedimiento que intentan mejorarla. Para conseguir esto, se deben recolectar métricas del software. La medición es muy común en el mundo de la ingeniería. [PRESS 93, cap 2]. Hay varias razones para medir el software: (1) para indicar la calidad del producto; (2) para evaluar la productividad de la gente que desarrolla el producto; (3) para evaluar los beneficios (en términos de productividad y de calidad) derivados del uso de

nuevos métodos y herramientas de ingeniería del software; (4) para establecer una línea base para la estimación; (5) para ayudar a justificar el uso de nuevas herramientas o de formación adicional.

Las mediciones pueden englobarse en dos categorías: *medidas directas* y *medidas indirectas*.

Entre las medidas directas del proceso de ingeniería del software, se encuentran el esfuerzo y el coste aplicado. Entre las medidas directas del producto se encuentran las líneas de código (LDC) producidas, la velocidad de ejecución, el tamaño de la memoria y los defectos observados en un determinado periodo de tiempo. Entre las medidas indirectas del producto, se encuentran la funcionalidad, calidad, complejidad, eficiencia, fiabilidad, facilidad de mantenimiento y muchas otras. El coste y esfuerzo requeridos para construir software, el número de líneas de código y otras medidas directas son relativamente fáciles de obtener, siempre que se hayan establecido convenios específicos de medidas. Sin embargo, la calidad y la funcionalidad del software, o su eficiencia y facilidad de mantenimiento son más difíciles de evaluar y solo se pueden medir indirectamente.

Las métricas del software pueden clasificarse en:

- a. Métricas de productividad, que se centran en el rendimiento del proceso de la ingeniería del software
- b. Métricas de calidad, que proporcionan una indicación de cómo se ajusta el software a los requerimientos implícitos y explícitos del cliente (conveniencia del software para su utilización)
- c. Métricas técnicas, que se centran en las características del software (por ejemplo su complejidad lógica o el grado de modularidad) más que en el proceso a través del cual ha sido desarrollado el software.

d. Métricas orientadas al tamaño, que se utilizan para obtener medidas directas del resultado y de la calidad de la ingeniería del software. Ejemplo: KLDC (miles de líneas de código).

e. Métricas orientadas a la función, que proporcionan medidas indirectas y las

f. Métricas orientadas a la persona, que ofrecen información sobre la forma en que la gente desarrolla software y sobre el punto de vista humano de la efectividad de las herramientas y métodos.

7. Registro y realización de informes: el registro de información y la generación de informes dan procedimientos para la recolección y divulgación de información de ACS. Los resultados de las revisiones, auditorías, control de cambios, prueba y otras actividades de ACS deben convertirse en una parte del registro histórico de un proyecto.

Relación del aseguramiento de calidad con las fases del ciclo de vida

De acuerdo con **[NASA-GB 97]**, durante el ciclo de vida de desarrollo, existen en cada fase del mismo, diversas actividades de aseguramiento de la calidad. En la conclusión de cada fase, el aseguramiento es un elemento clave para que la administración determine si se puede iniciar con la siguiente fase. Se sugieren las siguientes actividades:

1. En la fase de **conceptualización e inicio**, el aseguramiento se relaciona con escribir y revisar el plan administrativo, para asegurar que los procesos, procedimientos y estándares identificados en el plan, son apropiados, claros, específicos y auditables

2. En la fase de **requerimientos de software**, se asegura que los requerimientos están completos y fáciles de probar. Deberá existir “rastreadabilidad” entre los requerimientos del software y los requerimientos del usuario. Cada requerimiento del usuario debe estar relacionado con un requerimiento en el software. La especificación de requerimientos de software, recolecta la definición del probable comportamiento de éste así como sus características de desempeño y funcionalidad.

Los requerimientos deben ser analizados para identificar inconsistencias entre ellos, para determinar si se pueden realizar con las herramientas y recursos actuales y para detectar problemas que luego podrían ocurrir durante la implementación.

De la misma manera, se debe determinar las interfaces entre los requerimientos y otros componentes del ambiente operativo y su impacto en los primeros. Es muy importante priorizarlos y procurar, que más que esperar a desarrollar un producto con todos ellos implementados, se busque liberar una secuencia de productos en los que se van incorporando poco a poco los requerimientos definidos.

Es esencial que se mantenga a los usuarios, informados sobre el estado de los requerimientos, ya sea mediante reuniones o comunicaciones formales de tal manera que se tenga una línea base actualizada. Esta comunicación debería darse de manera similar, durante el resto del ciclo de vida del sistema. Los requerimientos deben ser la base para definir los casos y métodos de prueba del software

3. En la etapa de **diseño**, se asegura la adherencia a los estándares de

diseño aprobados, se controla que los documentos se construyan de acuerdo con la forma y contenido dictaminado en los estándares y se verifica que se le da solución a todos los componentes o requerimientos del software.

4. En la fase de **implementación**, se audita los resultados de las actividades de diseño y codificación, así como el estado de todos los ítems desarrollados. Se debe verificar que cada unidad de software, satisface los requerimientos y que todo se documenta debidamente. Cuando existen inconsistencias se deben tomar acciones correctivas.

5. En la fase de **integración y pruebas**, se busca controlar que todos los ítems implementados, sean probados. A la vez se verifica que las pruebas se ejecutan de acuerdo con los planes y en caso de alguna inconformidad, esto se reporta y se resuelve.

6. En la fase de **entrega y aceptación del producto**, se asegura el desempeño adecuado de la configuración final del software, de tal manera que se demuestre que todos los ítems del mismo están listos y funcionando adecuadamente.

Estándares de Calidad para el desarrollo de Software

Un estándar es una regla o base de comparación que se utiliza para evaluar el tamaño, el contenido o el valor típicamente establecido como una práctica común. Ayuda a evaluar la calidad de un objeto o de una actividad. **[Hump 95]**

El establecimiento de estándares y procedimientos para el desarrollo de software, es crítico, por cuanto éstos proveen el marco de referencia a través del cual éste evoluciona. Los estándares y procedimientos, según lo determina Bedini **[BEDINI**

89], establecen los métodos para su desarrollo, indicando el “modo de hacer las cosas”, para asegurar que se desarrollará un software con alta calidad. El papel del aseguramiento de la calidad es procurar su existencia y que sean adecuados. La documentación detallada de estándares y procedimientos es necesaria para que las actividades de aseguramiento no trabajen con conceptos erróneos.

Existen diferentes tipos de estándares, entre los que se incluyen: **[Hump 95]**

- Estándares de documentación, los cuales especifican la forma y el contenido para la documentación de planificación, control y productos, además de proveer consistencia a lo largo de todo el proyecto.
- Estándares de diseño, que especifican la forma y contenido del diseño. Ellos proveen reglas y métodos para trasladar los requerimientos del software al diseño y para representar este último en la documentación respectiva.
- Estándares de codificación (programación), especifican el lenguaje en el cual el código es escrito y define las restricciones de uso de ciertas características del mismo. Adicionalmente, definen las estructuras, convenciones de estilos, reglas para las estructuras de datos e interfaces y la documentación interna.

Las razones más importantes para tomar la decisión de utilizar estándares son:

[Hump 95]

- Son indispensables cuando muchas personas, productos y herramientas deben coexistir, ya que permiten la comunicación entre los desarrolladores y promueven el uso consistente de mejores metodologías y herramientas.
- Facilitan el mantenimiento del software.
- Facilitan la capacitación del personal.
- Proveen una base para evaluar los diferentes productos de software.

- Permiten definir el proceso de software de una organización.

En este trabajo, se hará referencia básicamente a los estándares de la ISO y de la IEEE.

Estándares de la ISO

De acuerdo con la información de **[CADENA 96]**, la Organización Internacional para la Estandarización, mejor conocida como ISO, conformada por representantes de los cuerpos normalizadores de aproximadamente cien países, fue establecida oficialmente el 23 de febrero de 1947 con el objeto de promover la estandarización internacional, de tal manera que se facilitara el intercambio internacional de bienes y servicios así como el desarrollo científico, tecnológico, intelectual y económico. En los Estados Unidos, la representación se llama The American National Standards Institute (ANSI).

ISO comprende alrededor de 180 comités técnicos. Cada uno es responsable de una o más áreas de especialización que comprende desde asbestos hasta zinc. Como ejemplos se pueden mencionar que hay normas ISO que abarcan desde las abreviaturas de los sistemas de medición hasta la especificación de protocolos de transferencia, pasando por especificación de tornillos, lentes, contenedores marítimos, medios magnéticos, hojas de papel, cables, elementos estructurales, pruebas de seguridad, simbología, medio ambiente, etc. **[JAGUADO 89]**

Los resultados del trabajo técnico de ISO son publicados como estándares internacionales. En este sentido, la ISO 9000 es producto de dicho proceso y es uno de los estándares de calidad más populares actualmente. Esta serie de estándares fue desarrollada por el Comité Técnico ISO 176 (ISO/TC176) en el año 1979 para armonizar el incremento de la actividad internacional en materia de administración de

la calidad y aseguramiento de estándares de calidad.

ISO 9000

Las series de ISO 9000 son un grupo de estándares que aunque son individuales, están relacionados y definidos a nivel internacional para la administración y aseguramiento de calidad. Ellos son genéricos, no específicos para cualquier producto, pueden usarse igualmente para manufactura y servicios industriales. Estos estándares fueron desarrollados para documentar efectivamente los elementos de sistemas de calidad que son instrumentados para mantener un sistema eficiente de calidad en la empresa. [ISO 93]

ISO 9000 proporciona al usuario una guía para la selección y uso de ISO 9001, 9002, 9003 y 9004. ISO 9001, 9002 y 9003 son modelos de sistemas de calidad para el aseguramiento de calidad externa.

Información contenida en el estándar ISO 9000

La serie de estándares ISO 9000 es genérico. Cada estándar abarca diferentes aspectos del aseguramiento de la calidad, dependiendo de las necesidades del usuario. El contenido de los estándares es revisado cada cinco años. El ISO 9001, ISO 9002 y 9003 describen tres distintos modelos de calidad. ISO 9001, 9002 y 9003 incluyen la necesidad de: **[ISO 91]**

- Un sistema de calidad efectivo, asegurando que las medidas son válidas.
- Que el equipo para pruebas y medición es ajustado regularmente.
- El uso de técnicas estadísticas apropiadas.
- Tener un sistema de rastreo e identificación del producto.

- Mantener un adecuado sistema de registro.
- Dar un adecuado manejo al producto.
- Tener un sistema de empaque y entrega.
- Poseer un sistema adecuado de inspección y pruebas.
- Asegurar el adecuado entrenamiento y experiencia del personal.

Beneficios del ISO 9000

Los estándares ISO 9000 han ganado mucha aceptación de manera rápida, desde su primera publicación en 1987. Esto se explica a raíz de que hay algunos beneficios al aplicar estas series en la empresa. Por ejemplo:

- Conduce a darle calidad al producto o servicio y evitar costos de inspecciones finales, costos de garantías y reprocesos.
- Reduce el número de auditorías de los clientes a los procesos de operación.
- Cada vez más los clientes aceptan proveedores con sistemas de calidad registrados que han sido evaluados por una tercera persona con base en esos estándares.
- Incrementa las ventas y ganancia.
- Incrementa la satisfacción de los clientes.
- Reduce costos y el tiempo del ciclo de vida.
- Mejora la consistencia y confiabilidad de los productos.
- Mejora la planificación.

Problemas del ISO 9000

La serie ISO 9000, ha tenido problemas serios en su implementación, por lo que ha sido objeto de muchas críticas como por ejemplo:

- Aunque las normas ISO 9001 a ISO 9003 tienen alcances específicos que se aplican a diferentes tipos de contratos, hubo desafortunadamente un error de percepción general en el sentido de que el nivel de calidad exigido por ISO 9001 era más alto que ISO 9003. De hecho varias empresas han seguido la táctica de certificarse gradualmente primero con ISO 9002 y luego ISO 9001, dando a la implementación del sistema de calidad un carácter gradual.
- Aunque el objetivo de las normas es establecer una confianza en los sistemas de calidad, se le ha venido dando un enfoque de que el cumplimiento de las normas ISO 9000 da una confianza en los productos. Esto es totalmente incorrecto, ya que ningún producto puede llevar el símbolo de ISO 9000 como sello de calidad.
- Hay una percepción extensa de que la serie ISO 9000 es muy mecánica, burocrática y que no utiliza las técnicas contemporáneas de administración de calidad.
- Pocos son los auditores que exigen evidencia de cumplimiento de atributos de calidad de los productos o servicios.

Estándares de la IEEE

Fundado en 1884, El Instituto de Ingenieros Eléctricos y Electrónicos es la más grande sociedad sin fines de lucro a nivel mundial, de profesionales técnicos dedicados al progreso de la teoría y aplicación de ingenieros eléctricos, electrónicos y de computación. Este instituto ayuda al desarrollo mundial, mediante la promoción de procesos de ingeniería como creación, desarrollo, integración y aplicación de conocimientos acerca de las tecnologías y ciencias que busquen el beneficio de la humanidad y las profesiones. El IEEE produce más de 310.000 ingenieros eléctricos, científicos y profesionales en aproximadamente 150 países. A través de estas personas, el IEEE se ha convertido en la autoridad y líder en muchas área técnicas que comprenden desde ingeniería informática, hasta tecnología biomédica y telecomunicaciones.

Mediante las publicaciones técnicas que emite, así como conferencias y actividades relativas a la definición de estándares, el instituto produce más del 30% de la literatura relativa a la ingeniería eléctrica, computadoras y tecnologías de control, manteniendo hasta la fecha más de 800 estándares activos en distintas ramas de aplicación.

Dentro del área de la informática, la IEEE tiene un conjunto de estándares de producto, de proceso y un glosario de términos.

Los estándares de proceso, abarcan:

- Aspectos del proceso de desarrollo.
- Inspecciones y auditorías.
- Administración de la configuración.
- Mantenimiento, entre otros.

Los de producto, se relacionan con:

- Características del software.
- Análisis de requerimientos.
- Diseño.
- Documentación del usuario.

Capítulo IV

Análisis de la metodología de desarrollo en SUPEN

Introducción.

Esta sección del documento tiene como objetivo realizar un análisis de la metodología de desarrollo actual de sistemas de la SUPEN (ver anexo 1) a la luz de tres hechos importantes:

1. Analizando las fortalezas y debilidades de la metodología de desarrollo actual de la SUPEN, enfocándose especialmente en el control de calidad. Al final de esta sección, se brinda una opinión del autor acerca de la metodología teniendo como parámetro principal el control de calidad.

2. Comparando la metodología de desarrollo de la SUPEN con los estándares de la IEEE y de la ISO. De esta comparación se espera obtener un análisis detallado de las fortalezas y debilidades de la metodología de desarrollo empleada por la SUPEN con base en algunos estándares de aceptación internacional.

3. Análisis de los sistemas transaccionales de la SUPEN. Para este análisis se han utilizado el cuestionario no. 1 denominado “Desarrollo de Sistemas de Información de la SUPEN” (ver anexo 3), que se aplicó a los funcionarios de la Intendencia Informática, quienes han sido los que han desarrollado los sistemas transaccionales. También se ha utilizado el cuestionario no. 2 denominado “Utilización de Sistemas de Información de la SUPEN” (ver anexo 4), que se aplicó a los usuarios del sistema, con el propósito de determinar cual ha sido la impresión de los usuarios de la SUPEN de los sistemas de información transaccionales, en relación con el desarrollo y el mantenimiento de los sistemas.

Esta sección concluye realizando un análisis de los sistemas transaccionales de la SUPEN

4.1 Análisis de las fortalezas y debilidades de la metodología de desarrollo de sistemas de la SUPEN desde el punto de vista de control de calidad.

La metodología de desarrollo actual de sistemas de la SUPEN se basa en cinco actividades principales (para mayor detalle ver anexo 1):

- Un estudio preliminar
- Análisis de requerimientos de sistemas
- Diseño por prototipo
- Construcción e interacción por prototipo
- Implantación del sistema.

De la lectura de dicho documento, se pueden desprender algunas conclusiones importantes acerca de la metodología de desarrollo actual de sistemas de la SUPEN:

Fortalezas

- Señala el propósito y los factores críticos de éxito en cada etapa.
- Define explícitamente una serie de actividades a realizar en cada fase de desarrollo.
- Define un comité de control de calidad.
- Se indica las personas responsables del buen funcionamiento de cada

etapa.

- Define el producto final de cada etapa.
- Brinda una serie de recomendaciones para agilizar el desarrollo de los proyectos en cada etapa.
- Señala en modo general las tareas de control de calidad al final de cada etapa.
- Se indican los formatos que deben tener los productos finales de cada etapa.
- Al ser una metodología de desarrollo por prototipos y al no incluir controles excesivos, desea agilizar el desarrollo de los proyectos.

Debilidades:

- Pese a definir las actividades, la metodología no profundiza en la forma en la que se deben realizar dichas actividades.
- No se señalan explícitamente las funciones y responsabilidades del comité de control de calidad.
- No se señalan con claridad las funciones y responsabilidad de los usuarios y los analistas durante el desarrollo de cada etapa.
- El control de calidad se realiza únicamente al final de cada etapa, es decir,

solo se revisan productos finales, con lo cual no se controla la misma durante el proceso de desarrollo.

- Pese a indicar los formatos que deben tener los productos finales de cada etapa, no se especifica claramente el contenido de cada documento final .

- La metodología no señala las actividades precedentes ni siguientes, ni las responsabilidades específicas de los involucrados durante el desarrollo de la etapa. Esto hace que se pierda el control de la misma.

Aún no se ahondará en las ventajas o problemas que se presenta en la metodología de desarrollo. Sin embargo, se puede notar que desde el punto de vista de la calidad de software, existen serias debilidades en la metodología de desarrollo de sistemas actualmente utilizada en la SUPEN.

- La metodología solo indica las actividades a realizar, pero no la forma en como deben hacerse. Esto se presta para una seria ambigüedad por parte de los participantes del proyecto.

- No existe una responsabilidad clara y definida de los participantes de los proyectos. Esto puede provocar un grave problema de autoridad y responsabilidad, ya que los participantes de los proyectos indicarán desconocimiento de lo que tienen que hacer. Esto puede provocar retrasos en la entrega de productos por un lado y, por otro lado, provocará irremediablemente omisiones en los documentos o productos finales.

- Al existir control de calidad solo al final de cada fase, se descuida el proceso del proyecto durante la fase misma.

- Los formatos de entrega de documentos son ambiguos, por tal motivo, los encargados de redactar los documentos de cada fase podrían ser ambigüedades al momento de desarrollar el mismo y corren el peligro de no desarrollar el documento en un formato fijo que sea estándar.

- Al no existir claridad en la precedencia y consecuencia de las actividades, queda a responsabilidad de los miembros del equipo de desarrollo controlar que las actividades del proyecto se desarrollen en la secuencia correcta y que todas las actividades de la etapa se hayan realizado.

4.2 Comparación de metodología de desarrollo SUPEN con estándares de IEEE e ISO.

Dado que en la sección anterior se señalaron las etapas de desarrollo de la metodología de desarrollo de la SUPEN, no se señalarán las etapas nuevamente. Por otro lado, en vista que lo esencial de este trabajo se enfoca al control de calidad en el desarrollo de los sistemas, se señalarán en esta sección únicamente las actividades de calidad que se desarrollan en los estándares de la IEEE y de la ISO y que no se están realizando en la metodología de desarrollo de la SUPEN.

4.2.1 Comparación de metodología desarrollo SUPEN con estándar IEEE.

- Al inicio del proyecto, se deben clarificar o realizar una serie de tareas importantes del sistema, tales como:
 - Razón por la cual surge el proyecto.
 - Estudio de factibilidad
 - Factores críticos de éxito del proyecto
 - Conocimiento del presupuesto y demás recursos del proyecto

- Establecer una etapa de definición de requerimientos
 - Identificar ideas o necesidades.
 - Refinar las ideas iniciales
 - Definir requerimientos de software. En esta parte se debe tener en cuenta:
 - Ambiente del proyecto
 - Restricciones del sistema
 - Análisis de riesgos.

- Definir requerimientos de interfase

- Priorizar e integrar requerimientos de software

- Definir requerimientos que cumplan las características de ser no ambiguo, completo, verificable, consistente, modificable y rastreable.

- Para cada requerimiento se debe tener en cuenta que éste sea:
 - Funcional
 - Considerar rendimiento (velocidad, disponibilidad, tiempo de respuesta)
 - Limitaciones impuestas en la implementación (por lenguaje de programación, políticas de integridad, ambiente operativo y límite de recursos)
 - Atributos (Portabilidad, correctitud, mantenibilidad, seguridad)
 - Interfaces externas (gente, hardware, con otro software)

- Establecer una etapa de diseño de software.
 - Definir la arquitectura del sistema, es decir, transformar los requerimientos de software en conceptos de diseño

 - Diseñar la base de datos

 - Diseñar interfaces

 - Desarrollo de algoritmos (procesos de funciones específicas)

 - Ejecutar diseño detallado.

- Establecer una etapa de codificación
 - Crear un ambiente de desarrollo
 - Crear objetos del sistema en unidades procesables
 - Crear programas fuente
 - Pruebas internas a los programas
 - Si el desarrollo del sistema se trabaja con prototipos, se deben realizar una serie de pasos previos:
 - Establecer propósito y alcances del prototipo

 - Descripción general del prototipo
 - Perspectiva y funciones del prototipo
 - Características de uso
 - Restricciones generales

- Especificación de requerimientos
 - Requerimientos funcionales. Para cada requerimiento funcional establecer entradas, procesos, salidas y limitaciones, esto es, como debe comportarse el software cuando ocurren errores o excepciones .
 - Establecer interfaces externas, para conocer necesidades de software y hardware adicional o necesidades de comunicación.
 - Establecer requerimientos de ejecución.
 - Diseño de atributos

- Establecer una etapa de prueba
 - Crear un ambiente de prueba
 - Crear datos de prueba
 - Crear casos de prueba
 - Ejecución de pruebas
 - Revisión de resultados de pruebas
 - Definir plan de integración
 - Ejecutar plan de integración

- Establecer una etapa de Postdesarrollo
 - Plan de instalación del sistema
 - Distribución del software
 - Instalación del sistema
 - Generar aceptación en el cliente

- Establecer un estándar de documentación que debe contemplar al menos

la descripción de los procedimientos, operación del sistema, instalación del software, comandos, mensajes de error, llamados al programa, guía de referencia rápida y utilitarios del sistema.

- Establecer una fase de mantenimiento. Se debe definir un estándar que permita cambiar o adaptar el software en el tiempo, durante su vida útil. El mantenimiento debe establecer una serie de acciones:

- Identificación y clasificación del problema o modificación.
- Análisis del cambio.
- Diseño del cambio.
- Implementación.
- Pruebas.
- Aceptación de las pruebas por parte del usuario.
- Implementación del cambio.
- Documentación del cambio.

- Establecer planes de verificación y validación de software, que entre otras cosas debe contemplar una serie de puntos importantes:

- Propósito
- Alcance
- Estudio de la organización
 - Asignación de responsabilidades
 - Descripción del método, técnicas y herramientas a utilizar

- Aplicación de verificación y validación en las distintas fases de desarrollo
 - Conceptual
 - Definición de requerimientos
 - Diseño
 - Codificación
 - Prueba
 - Implementación
 - Operación y mantenimiento

- Establecimiento de un plan de aseguramiento de calidad del software que contemple al menos lo siguiente:
 - Establecer un propósito de calidad

 - Establecer requerimientos mínimos de documentación, que corresponda al estándar de calidad elegido por la organización en al menos las siguientes fases:
 - Especificación de requerimientos de software
 - Descripción del diseño de software
 - Plan de validación y verificación del software
 - Documentación del usuario
 - Plan de administración de la configuración de software

- Establecimiento de estándares, prácticas, convenciones y métricas que correspondan al plan de calidad trazado en la organización.

- Realizar revisiones o auditorías del software, que contemple al menos lo siguiente:

- El propósito de la revisión o auditoría

- Establecer requerimientos mínimos de auditoría en las siguientes fases:

- Revisión de requerimientos de software

- Revisión del diseño de software.

- Realizar auditoría funcional, física (revisión interna) y lógica del software.

- Establecer planes de entrenamiento en el software a los usuarios.

- Especificar los métodos y procedimientos empleados para identificar, asegurar, monitorear y controlar áreas de riesgo durante el desarrollo del software.

4.2.2 Comparación de metodología desarrollo SUPEN con estándar ISO.

4.2.2.1 Acciones de calidad relacionadas con el desarrollo de sistemas.

- Políticas de calidad. Se indica que es responsabilidad gerencial señalar los objetivos de la organización en relación con la calidad y su compromiso con ésta.

- Responsabilidad y autoridad. Se define la responsabilidad, autoridad e

interrelación de todo el personal que administre, efectúe y verifique trabajo que incida en la calidad, especialmente en lo que se refiera a iniciar acciones, identificación y registro de problemas de calidad del producto y control del producto hasta que se haya corregido la deficiencia o condición no satisfactoria del producto.

- Asignación de personal a las actividades de verificación, tales como inspección, ensayo y control de diseño.
- Se debe establecer y mantener un sistema de calidad documentado, el cual se supone es integrado a través de todo el ciclo de vida.
- Auditorías internas del sistema de calidad, para verificar si las actividades cumplen con convenios planificados y determinar la eficacia del sistema de calidad.
- Implementar acciones correctivas que permitan investigar la causa de sistemas que no provoquen conformidad y de esta manera implementar acciones preventivas.
- Las actividades de calidad se deben organizar de acuerdo con un ciclo de vida de desarrollo de sistemas implementado.
- Establecer el alcance y los requerimientos del proyecto y documentarlos
- Identificar riesgos y contingencias e implementar acciones para enfrentarlos.
- Establecer criterios de aceptación del proyecto.

- Establecer la forma de trabajar los problemas detectados después de haber aceptado el proyecto.

- Establecer métodos para acordar los requerimientos y aprobar los cambios.

4.2.2.2 Acciones de calidad relacionadas con el ciclo de vida de sistemas.

- Establecer una formulación de objetivos como referencia en la definición de proyectos.

- Organizar los recursos del proyecto, incluyendo la estructura del equipo, las responsabilidades y los recursos materiales por usar.

- Establecer programa del proyecto, identificando las tareas a realizar, los recursos y el tiempo requerido en cada una y las interrelaciones entre tareas.

- Establecer planes de calidad, estableciendo objetivos medibles.

- Establecer planes de administración de la configuración para identificar, rastrear y controlar las versiones de cada ítem del software.

- Para cada fase de desarrollo del sistema se deben establecer: las entradas, las salidas, los procesos de verificación y los problemas potenciales que se presentan en cada fase.

- Establecer algunas métricas que sean representativas del desempeño tanto

a nivel informático como a nivel del usuario. Los valores obtenidos de estas métricas deben registrarse en forma periódica, con el fin de determinar el nivel adecuado de medición y poder aplicar acciones correctivas en caso que se excedan o empeoran los niveles aceptables para cada métrica.

- Establecer un adecuado control del proyecto, que permita determinar los grados de avance, las responsabilidades organizacionales y la asignación de recursos al trabajo.
- Establecer criterios de aceptación en cada fase que sean cruciales para la seguridad y el desempeño adecuado.
- Establecer criterios de verificación en cada fase, para asegurar que las salidas de la fase de desarrollo cumplan con requisitos especificados en la entrada de la fase, efectuando revisiones de desarrollo en puntos apropiados de la fase.
- Establecer planificación de pruebas, el nivel de desarrollo de la misma, la integración de los sistemas, definición de datos y los criterios de aceptación de las pruebas.
- Establecer un plan de mantenimiento, que defina el alcance del mantenimiento, la resolución de los problemas identificados, el registro del mantenimiento en un formato predefinido y el apoyo a las actividades de mantenimiento en caso que este sea repentino.

4.3 Análisis de sistemas transaccionales de la SUPEN.

Para realizar este análisis, se procederá a presentar los hallazgos y hechos más relevantes descubiertos con la aplicación de los cuestionarios.

4.3.1 Análisis de los hechos descritos por los analistas de la Intendencia Informática

- Metodología y procedimientos.

Se utiliza la “Metodología para desarrollo de sistemas de la SUPEN”, la cual básicamente se basa en desarrollo de sistemas mediante la técnica de prototipos.

- Existen procedimientos por escrito que guíen las fases de desarrollo.

Se utilizan los que están descritos en la metodología de sistemas actualmente utilizada.

- No se realizan estudios de factibilidad en las fases previas de los proyectos. Esto implica que no se establece un dimensionamiento real de los sistemas a realizar. Los proyectos de desarrollo de sistemas de la SUPEN obedecen estrictamente a las labores de apoyo que Informática debe dar a las funciones que realiza la SUPEN, las cuales están determinadas por las leyes y que además, son de acatamiento obligatorio. Desde el punto de vista operacional, no se analiza como el sistema repercute en las funciones de las personas de la organización, técnicamente no se analiza si los recursos con los que cuenta la Intendencia Informática pueden realizar los proyectos o si tienen las capacidades técnicas o si la organización cuenta con los recursos tecnológicos necesarios para realizarlos y económicamente no se miden los costos económicos de los proyectos, a excepción que sea mediante un desarrollo

externo, para el cual hay que contar previamente con dinero en la partida de inversiones.

- Existe una biblioteca de procedimientos almacenados pero ya no se usa. En este momento, se le ha contratado a una empresa externa el desarrollo de una biblioteca de procedimientos que serán utilizados luego por los sistemas de la SUPEN.

- El procedimiento que se sigue para aplicar pruebas al software producido, dada la naturaleza de los sistemas de la SUPEN (carga y validación de datos recibidos de otras entidad), es preparar archivos de prueba con datos correctos y erróneos, someterlos a aprobación del (de los) usuario(s), procesarlos en el sistema y analizar los resultados. Por otra parte, en vista que los sistemas se desarrollan por medio de prototipos, es más fácil desarrollar sistemas y hacer pruebas durante todo el desarrollo del mismo.

- Durante el desarrollo de las pruebas, participan trabajando conjuntamente los analistas, los usuarios y también un funcionario de la Auditoría Interna. Los datos de prueba los prepara generalmente el usuario, aunque en ocasiones y dependiendo de la complejidad, lo hace conjuntamente con el analista. Este proceso consiste en la preparación de archivos con estructura idéntica a los que se solicitan a las entidades externas, incluyendo en ellos registros con información correcta y errónea, para luego proceder a la revisión conjunta.

- La metodología empleada para encontrar defectos y fallas en el software es “walkthrough” la cual consiste en conformar un grupo que evalúe el trabajo de una etapa. El analista encargado de la etapa deberá exponer su trabajo realizado y el grupo deberá encontrar posibles errores o mejoras en el sistema. Al finalizar esta sesión se deberá hacer un documento que indique las observaciones

correspondientes a la etapa en evaluación. En caso de existir observaciones en la etapa evaluada se deberán hacer los cambios necesarios y volver a hacer la evaluación, hasta que la etapa esté aprobada.

- Según indican los analistas de sistemas, las pruebas son validadas contra los requerimientos de los sistemas pues durante las pruebas se someten a revisión todos los procesos incluidos en el sistema.

- No existen métricas que permitan medir la adaptabilidad, interoperabilidad, confiabilidad, flexibilidad, funcionalidad y eficacia del sistema. Para considerar las pruebas exitosas únicamente se considera la exactitud de los resultados.

- Si bien la Metodología contempla dos etapas iniciales para el análisis y diseño del sistema y participación del usuario en la prueba del sistema, no existe una forma de validar los requerimientos del usuario, el diseño del sistema con los requerimientos del usuario y el sistema generado como un todo.

- Al final de cada fase, se genera un documento según lo indica la metodología de desarrollo de sistemas de la SUPEN. Dicho producto es sometido a evaluación y aprobación por parte del usuario.

- Una debilidad encontrada es que actualmente no existe algún tipo de control que verifique que se está siguiendo la metodología de acuerdo a los manuales o guías existentes, por tal motivo, queda a responsabilidad del analista emplear o no la metodología.

- La documentación que se maneja de los sistemas es interna, no externa. No se mantiene actualizada.

- No existen estándares para elaborar planes de proyectos, documentar sistemas, para realizar comentarios, codificar sistemas, realización de las pruebas y control de versiones.

- La planificación de un nuevo sistema se da generalmente como producto de reuniones donde participan informáticos y funcionarios de las áreas usuarias. La definición de requerimientos se realiza conjuntamente entre el usuario y el analista durante la fase de análisis y luego todo esto se incluye en el documento que se genera como producto final de esa fase. Cabe destacar que las estimaciones de tiempo las realiza el analista basado en su experiencia.

- Ya relacionado propiamente con el mantenimiento, la prioridad de los cambios solicitados en el sistema la define propiamente el Superintendente, los Jefes de Área y los usuarios. En el proceso para realizar el mantenimiento, el usuario especifica sus requerimientos, el analista los realiza, los somete a revisión por parte del usuario y se ponen en producción.

- Hasta ahora no se tiene un mantenimiento de versiones como tales, pero los cambios se registran en una bitácora y se documentan internamente en el código fuente.

- La frecuencia de cambios en los sistemas se da entre 30 y 90 días, el mantenimiento que se da es sobretodo correctivo, en algunos casos son cambios sustanciales y en otros son cambios superficiales. Como se mencionó anteriormente, los cambios son documentados internamente.

4.3.2 Análisis de los hechos de los usuarios de los sistemas transaccionales de la SUPEN.

- El tiempo de uso por parte de los usuarios es variable. En el caso de los usuarios responsables de los sistemas, casi todos tienen dos años de estar utilizando estos sistemas, fueron los que definieron los requerimientos. Otros usuarios, que no han sido los responsables del desarrollo de las aplicaciones, varían su tiempo de utilización del sistema entre 8 y 14 meses.
- Los usuarios responsables de los sistemas han tenido una participación importante en el desarrollo del sistema y que su opinión siempre ha sido tomada en cuenta. Ellos describen que los requerimientos salen en conjunto entre el área Informática y ellos.
- De todos los usuarios encuestados, solamente uno de ellos indicó que tenía algún conocimiento de la metodología de desarrollo de sistemas. Pero todos coinciden en indicar que nunca han recibido algún tipo de capacitación en la misma.
- Los usuarios indican que básicamente han participado en las etapas de definición de requerimientos y pruebas del sistema.
- Los usuarios no mencionan que haya habido problemas de comunicación con los analistas. Más bien, destacan siempre su espíritu de colaboración.
- Los usuarios indican que no ha habido malos entendidos en la comprensión de los requerimientos por parte de los analistas en términos generales. En todo caso, cuando algún requerimiento no se ha comprendido, se ha recurrido a explicar mejor el requerimiento, en ocasiones empleando hojas

electrónicas.

- Los usuarios han participado en la presentación de prototipos, especialmente en la fase de definición de requerimientos y coinciden en que esta técnica les ha ayudado a corregir problemas de comprensión de requerimientos presentados.

- Los usuarios indican que cuando se han presentados errores, en algunos casos estos se han corregido rápido, pero en otras ocasiones no. Sin embargo, ellos comprenden que los funcionarios de informática en ocasiones no han podido atenderlos por estar asignados a otros trabajos.

- Los usuarios reconocen que los cambios en los sistemas se dan frecuentemente debido al dinamismo con el que se mueve el sector financiero. En ocasiones los cambios se han podido realizar rápidamente.

- Los usuarios tienen conocimiento que no existe documentación de los sistemas.

- La mayor parte de los usuarios destaca que los sistemas son confiables, seguros, claros y fáciles de usar. Uno de los usuarios destacó que la información que los sistemas muestran esta basada en los datos enviados por las operadoras de pensiones y en el caso que no existan los adecuados controles o que la SUPEN flexibilice los criterios de validación de información, esto repercute en la confiabilidad de la información de los sistemas.

- En términos generales, los usuarios están satisfechos con los sistemas desarrollados. Algunos de los problemas que los usuarios destacan en los

sistemas de información actual, se listan a continuación:

- Los tiempos de respuesta de los sistemas son lentos.
- El manejo de errores por parte de los sistemas no es amigable.
- Los sistemas no permite generar consultas personalizadas.
- No existe capacidad de adaptar rápidamente los sistemas a los cambios presentados en el sector de pensiones.

También destacan otra serie de problemas, que no pueden achacarse directamente a los sistemas, pero que afectan la percepción de los usuarios en relación con los sistemas, problemas tales como:

- Consideran que el equipo ya no llena las expectativas de rendimiento.
- Falta de personal en el departamento de Informática, lo cual en ocasiones atenta contra la oportunidad de recibir un mejor servicio.
- Las operadoras de pensiones no han podido corregir una serie de errores en el envío de información. Esto hace que algunos módulos de los sistemas existentes aún no hayan podido probarse. Por tanto, aún no han podido verificar si el sistema responde adecuadamente a los requerimientos de los usuarios.

Después de presentar los puntos de vista de los analistas de sistemas y de los usuarios, se destacan algunos problemas importantes existentes en los sistemas de información de la SUPEN:

- En general, la percepción general que se puede sentir, es que la organización SUPEN como tal aún está en una fase de arranque de sus

sistemas, es decir, aún estos no han madurado y están todavía a prueba.

- Existe una metodología de desarrollo, pero no es conocida por los usuarios y es utilizada a discreción de los analistas, sin que exista una forma de controlar la labor que éstos realizan.
- Al no existir estudios de factibilidad no existe un debido análisis del entorno de los proyectos de software.
- El único parámetro de evaluación del sistema es su funcionalidad, es decir, que brinde resultados correctos en casos concretos definidos por los usuarios. Dado que no se manejan métricas, el resto de aspectos a evaluar de un sistema de información son dejados de lado.
- Al no existir estándares ni controles efectivos durante el desarrollo de los sistemas, queda claro que los sistemas pueden presentar muchas debilidades de tipo estructural.
- No hay una forma efectiva de validar los requerimientos del sistema ni de comprobar la unicidad del sistema como un todo.
- Existe mantenimiento sobre sistemas que no han sido completamente probados en períodos muy cortos de tiempo (de 30 a 90 días).
- No es posible tener sistemas que brinden resultados correctos sino son corregidos los datos fuente que alimentan los sistemas de la SUPEN, que en este caso son los datos enviados por las operadoras de pensiones.

- No existe documentación externa y la documentación interna es manejada a discreción del analista.
- Los usuarios han escuchado hablar de prototipos, pero no han recibido capacitación del área informática.
- Los usuarios están usando sistemas que no se han terminado de probar.
- La falta de estándares en las distintas fases de desarrollo de sistemas afectan desde ya (y sin que los sistemas estén aún finalizados) el desempeño de los sistemas.

4.4 Conclusión.

Después de haber realizado el análisis de la metodología de desarrollo de la SUPEN a la luz de las tres consideraciones señaladas en el punto 4.1, se puede realizar un análisis en términos generales de la metodología de desarrollo de sistemas actual de la SUPEN. Dicho análisis al mismo tiempo sirve de base para la metodología que se propone en el próximo capítulo.

- La metodología contempla una serie de actividades que apoyan a los analistas en su labor de desarrollo, pero queda a un nivel muy general.
- Pese a mostrar una actividad de control de calidad al final de cada fase, no se hace una descripción clara de las actividades y responsabilidades de los funcionarios durante cada etapa de desarrollo.
- No están claramente definidas las actividades precedentes y consecuentes.

Además no están definidas las responsabilidades de control durante el desarrollo de sistemas.

- Los usuarios no tienen conocimiento de la metodología de desarrollo de sistemas de la SUPEN.
- En la práctica, se hace mayor énfasis en las etapas de definición de requerimientos y pruebas del sistema.
- Pese a existir una fase de documentación, en la práctica ésta no se da.
- En los sistemas actuales, hay módulos que aún no se han probado. Así que nadie está seguro que funcionen correctamente.
- Los analistas de sistemas aplican la metodología de desarrollo a su responsabilidad. Es más, los proyectos no se desarrollan con una adecuada planificación, sino según se vaya ocupando.
- Según los analistas de sistemas, no hay una forma efectiva de validar los requerimientos ni la unicidad del sistema como un todo.
- El mantenimiento que se hace en los sistemas es de tipo correctivo, en forma muy seguida y no hay un adecuado control de versiones.
- Después de realizar el análisis de las metodologías de la IEEE y de la ISO, se puede llegar a la conclusión que la metodología de desarrollo de sistemas de la SUPEN presenta deficiencias en muchas de sus fases y especialmente, en la aplicación de control de la calidad.

Capítulo V

Propuesta de una metodología de desarrollo en la SUPEN

Introducción

En esta sección del documento, se procede a realizar la definición de la propuesta de una metodología de desarrollo de sistemas para la SUPEN, procurando que dicha metodología enfatice más en el control de calidad que se deben aplicar en las distintas etapas del desarrollo del sistema.

Para enfatizar las etapas de control de calidad, hay que considerar que la metodología actual de desarrollo de sistemas de la SUPEN presenta muchas debilidades en los aspectos de control. De esta manera, el trabajo de análisis que se realizó en el capítulo anterior ha ayudado a tener claro los problemas que se están presentando actualmente y que podrían venir a ser solucionados con la metodología que aquí se propone.

Para la elaboración de la metodología, primeramente se definen los pasos a seguir en cada fase de la metodología, los responsables de cada actividad y los productos que se deben generar. Posteriormente, se procederá a explicar con detalle en que consiste cada etapa que se propone, los productos resultantes y a sugerir lineamientos que se deben tener en cuenta.

Antes de iniciar la descripción del sistema, se establece la terminología que identifica los responsables de cada actividad en la tabla 3.

INICIALES	DESCRIPCION
Equipo de trabajo del proyecto	Esta constituido por: <ul style="list-style-type: none"> • Usuario Responsable • Analista Líder • Grupo de usuarios • Grupo de analistas
Intendente Informático	Responsable de la Intendencia Informática
Comité Informático	Esta constituido por: <ul style="list-style-type: none"> • Superintendente General • Intendente General • Intendente Financiero • Intendente de Control
U.R.	Usuario responsable del sistema
A.R.	Analista Responsable del sistema
Grupo de usuarios	Grupos de usuarios de apoyo
Grupo de analistas	Grupo de analistas de apoyo en desarrollo
D.B.A.	Administrador de la base de datos
Adm. Red	Administrador de red
Soporte Técnico	Personal de apoyo de soporte técnico

Tabla 3. Responsables de cada actividad

La metodología de desarrollo que se propone consta de ocho etapas:

1. Inicio del proyecto. Se realiza un estudio del ambiente y de los recursos del proyecto, para tener una idea de la viabilidad del proyecto.

2. Definición de requerimientos. Es la etapa en la cual los usuarios definen sus requerimientos de información.

3. Diseño. En esta etapa se propone a nivel conceptual una solución a los requerimientos de los usuarios.

4. Codificación. Es la etapa de programación, es decir, convertir los requerimientos en instrucciones para la computadora.

5. Pruebas. En esta etapa se prueba la correctitud de los programas del usuario.

6. Implementación. Poner en funcionamiento la solución programada.

7. Post-implementación. Sirve como una etapa de evaluación de las actividades realizadas durante el desarrollo del sistema

8. Mantenimiento. Es una etapa que se realiza posterior al desarrollo de sistema.

Todas estas etapas se hacen en forma secuencial, procurando poner un énfasis especial en la etapa de definición de requerimientos, a efectos de que el resto del proyecto no sufra algún retraso.

En ocasiones, existen proyectos que por su naturaleza no necesitan realizar ciertas etapas. Por ejemplo, un caso que podría ilustrar esta situación es cuando se compra algún paquete que realiza alguna función específica. En estos casos, podría ser que no sea necesario realizar las primera cuatro etapas de la metodología. En este caso, se puede iniciar las pruebas inmediatamente.

En este sentido, la metodología es flexible y no obliga a que para todos los sistemas, se deban realizar todas las etapas. Sin embargo, si sería conveniente que en esos casos, se realice una aclaración documentada en la cual se explican las razones por las cuales no se hicieron las etapas precedentes.

Esta metodología es posible de usarse en caso que los desarrollos en la SUPEN se realizará con empresas de desarrollo externa.

Inicio del proyecto

ACTIVIDAD	RESPONSABLES	INSUMO	PRODUCTOS
IP-01 Definición del proyecto	Equipo de trabajo del proyecto	INS-01 Documentación de la SUPEN: <ul style="list-style-type: none"> • Leyes • Normativa • Reglamentos • Circulares • Memorandum 	
IP-02 Conocimientos del presupuesto y demás recursos del proyecto	Equipo de trabajo del proyecto	INS-02 Presupuesto INS-03 Listado de analistas de sistemas de la organización y sus perfiles	
IP-03 Estudio de factibilidad	Equipo de trabajo del proyecto	INS-04. Perfil de la aplicación a realizar INS-05. Plataforma tecnológica	
IP-04 Factores críticos de éxito	Equipo de trabajo del proyecto	Toda la documentación anterior, así como investigaciones técnicas y económicas	
IP-05 Conclusión y recomendaciones	Equipo de trabajo del proyecto		PR-IP-01. Estudio preliminar del proyecto
IP-06 Control de calidad de la etapa	<ul style="list-style-type: none"> • Equipo de trabajo • Intendente Informático • Comité Informático 	PR-IP-01. Estudio preliminar del proyecto	

IP-07 del Informático	Dictamen Comité	Comité Informático		PR-IP-02. Carta con resolución del Comité Informático
-----------------------------	--------------------	--------------------	--	---

Tabla 4. Actividades de Inicio del Proyecto

La tabla 4 muestra las actividades de la fase Inicio del Proyecto, las cuales son descritas a continuación.

- Definición de proyecto.
 - Inicialmente los miembros del grupo que realiza la definición del proyecto deben analizar la documentación de la organización relacionada con el sistema. Es importante considerar los documentos existentes para que todos los miembros del equipo de trabajo conozcan la temática del proyecto, proceso manual que realizan los usuarios.
 - En esta fase se debe identificar cual es la razón por la que surgió el proyecto, que necesidad viene a reforzar y si es realmente necesario implementar un sistema para satisfacer dicha necesidad.
 - En esta etapa también se debe analizar el tamaño del proyecto para tener una idea de los recursos y del costo que dicho proyecto supondrá para la SUPEN.
 - Conocimiento del presupuesto y demás recursos del proyecto. Se debe tener conocimiento del presupuesto con el que se va a contar y conocimiento de los recursos del proyecto, tales como tiempo así como de los funcionarios y sus calidades e idoneidad para el desarrollo del presupuesto.
- Estudio de factibilidad. Se realiza un estudio de factibilidad técnica,

económica y operativa que contemple algunos de los siguientes aspectos:

- Técnicamente.
 - En este estudio, analizar desde el punto técnico la capacidad del hardware y del software para realizar el proyecto que se desea realizar.
 - Estar seguro que con la plataforma a usar se obtendrá confiabilidad, integridad, exactitud y seguridad en los datos.
 - Estar seguro que con la plataforma a usar se obtendrán tiempos de respuesta adecuados con la cantidad de los usuarios que se tienen en la organización.

- Económicamente.
 - Considerar los costos totales del proyecto, que significa por ejemplo desarrollo de sistemas interno, lo cual a su vez significa tener uno o más analistas encargados del proyecto o bien, contratar alguna empresa o persona que venga a desarrollar el sistema. Otros costos relacionados con el desarrollo y que se deben considerar en esta etapa, es el costo de hardware, software y capacitación para la aplicación que se está considerando.
 - Beneficios que se obtendrán en la reducción de costos, reducción de errores, o bien, desde el punto de vista cualitativo analizar los beneficios de la oportunidad con que se pueden procesar los datos y obtener la información.
 - Importante es también analizar los costos de la organización

en caso de no realizar el proyecto.

- Operativamente.
 - Tener el apoyo del Superintendente e Intendente General, de parte del director de División y de los usuarios que hayan sido nombrados como encargados del proyecto o bien, como usuarios de apoyo.
 - Considerar que es importante que los usuarios tengan conocimiento de la metodología de desarrollo de sistemas y a su vez, que los usuarios participen desde las etapas iniciales del proyecto.
 - Tener en cuenta los beneficios que el sistema tendrá operativamente para los usuarios; es decir, se procura con el sistema apoyar la labor que realizan los usuarios, por tanto desde esta etapa se debe definir en qué y cómo viene a beneficiar el sistema a los usuarios.
 - Se evalúa también la funcionalidad en operación del sistema. Se evalúan los procedimientos existentes para analizar si se necesita crear una nueva estructura organizativa y nuevos puestos para soportar la operación del sistema y de la organización.

- Factores críticos de éxito del proyecto. En esta actividad, se deben considerar todos los aspectos que podrían hacer que el proyecto fracase. Estos factores podrían ser tanto internos como externos a la organización. La idea es levantar una lista de dichos factores así como la manera en la cual se enfrentarían los problemas.

- Conclusiones y Recomendaciones. En esta actividad, el grupo de trabajo debe llegar a conclusiones relacionadas con la conveniencia y factibilidad del desarrollo del proyecto y emitir una recomendación en el sentido de desarrollar o no el proyecto.

- Control de calidad de la etapa. Esta actividad consiste en analizar los resultados de la etapa y más aún, los criterios utilizados por los miembros del equipo de trabajo para llegar a las conclusiones y recomendaciones dadas. En esta etapa, la asesoría del Intendente Informático o bien, de un consultor externo viene a enriquecer y a dar mayores criterios de decisión

- Dictamen del Comité Informático. El Comité Informático debe decidir con base en la recomendación dada por el grupo de trabajo y con base en el estudio detallado del documento, si procede o no el desarrollo del sistema.

Definición de requerimientos

ACTIVIDAD	RESPONSABLES	INSUMO	PRODUCTOS
DR-01 Identificar ideas o necesidades	Equipo de trabajo del proyecto	PR-IP-07 Dictamen del Comité Informático	PR-DR-01. Ideas iniciales del proyecto
DR-02 Refinar ideas iniciales	Equipo de trabajo del proyecto	PR-DR-01. Ideas iniciales del proyecto	PR-DR-02. Documento inicial de requerimientos del sistema
DR-03 Definir requerimientos de interface	Equipo de trabajo del proyecto		
DR-04 Priorizar e integrar requerimientos de software	Equipo de trabajo del proyecto	PR-DR-02. Documento inicial de requerimientos del sistema	PR-DR-03. Documento definitivo de requerimientos del sistema
DR-05 Control de calidad de la etapa	<ul style="list-style-type: none"> • Equipo de trabajo • Intendente Informático • Comité Informático 	PR-DR-03. Documento definitivo de requerimientos del sistema	PR-DR-04 Observaciones y correcciones al documento de definición de requerimientos
DR-06 Aprobación de usuario responsable de la etapa de definición de requerimientos	<ul style="list-style-type: none"> • U.R. • A.R. 	PR-DR-03. Documento definitivo de requerimientos del sistema corregido	PR-DI-05 Aprobación de requerimientos por parte del usuario

Tabla 5. Actividades de Definición de Requerimientos

La tabla 5 muestra las actividades de la fase Definición de Requerimientos, las cuales son descritas a continuación.

- Identificar ideas o necesidades. El equipo de trabajo debe reunirse y a partir de estas reuniones, definir las ideas o necesidades iniciales que se van a desarrollar en el sistema. Básicamente en esta etapa, se define un

listado de requerimientos sin entrar en mucho detalle con respecto a cada requerimiento. Es importante que los usuarios revisen una y otra vez este listado para asegurarse que esta definiendo todos los requerimientos,

- Refinar las ideas iniciales. Es una etapa de revisión de ideas iniciales. En esta etapa, se debe tomar requerimiento por requerimiento y detallarlo, contemplando que datos se necesitan de entrada, que tipo de procesos se deben realizar sobre los datos y que información produce el requerimiento.

En la definición de cada requerimiento, los analistas de sistemas deben procurar que éstos contemplen ciertas características:

- Funcional
- Considerar rendimiento (velocidad, disponibilidad, tiempo de respuesta)
- Limitaciones impuestas en la implementación (por lenguaje de programación, políticas de integridad, ambiente operativo y límite de recursos)
- Atributos (Portabilidad, correctividad, mantenibilidad, seguridad)
- Interfaces externas (gente, hardware, con otro software)
- Los requerimientos deben ser no ambiguos, completos, verificables, consistentes, modificables y rastreables.

De tal forma, la responsabilidad del analista es proporcionar los requerimientos que el usuario necesite, pero procurando que cumplan características que se expusieron en el párrafo anterior.

- Definir requerimientos de interface. Es importante contemplar los requerimientos de interface desde dos puntos de vista: primero, desde el punto de vista de la interface del usuario, es decir, de su interacción con los programas y las facilidades que le brinde el programa para trabajar. Segundo, desde el punto de vista de su relación con otros sistemas, de la información que trasiega entre ellos y de como el sistema a desarrollar afecta otros sistemas de la empresa.

Analizando los requerimientos de interface desde el punto de vista del usuario, es importante que se tomen en cuenta la facilidad para los usuarios de interactuar con los programas. En este sentido, es bueno que existan estándares de pantallas y de reportes que brinden guías para el desarrollo de los programas.

Desde el punto de vista de interface en relación con otros sistemas, el analista de sistemas debe analizar como se verán afectados otros sistemas con la información del sistema a desarrollar. En este sentido, se analiza que procesos o datos pueden afectar otros sistemas y como afecta a otros sistemas la reversión de procesos en el sistema a desarrollar, a efectos de mantener la integridad en los datos.

- Priorizar e integrar requerimientos de software. Una de las partes importantes en el desarrollo de los sistemas, es el proceso de priorización en los requerimientos de los sistemas. Esto significa que algunos requerimientos se deben haber satisfecho antes que otros, sea porque son más importantes o sea, porque los nuevos requerimientos necesitan que otros se hayan cumplido antes.

Finalmente, debe tomarse en cuenta, que en todas las fases del

desarrollo de requerimientos, es responsabilidad de todo el grupo de trabajo (usuarios del sistema y analistas). Por tanto las decisiones que se tomen, es responsabilidad de todo el grupo.

- Control de calidad de la etapa. Esta actividad consiste en analizar los requerimientos que se hayan definido de la etapa. Algunas de las cosas que se revisan son las siguientes:

- Requerimientos listados. Se revisa que todos los requerimientos identificados sean los que se necesitan; es decir, verificar que no falte alguno.

- Integración de los requerimientos. Todos los requerimientos deben estar integrados con el fin de cumplir un propósito. En esta parte es necesario que se analice que todos los requerimientos definidos conjuntamente permiten que la organización logre un propósito. En ocasiones cuando no existe esta actividad, los requerimientos de los usuarios pueden dispersarse.

- Redacción de los requerimientos. Es importante revisar la redacción de los requerimientos. En general, éstos deben ser redactados en forma de verbo infinitivo, con la menor cantidad de palabras o redundancias posibles y procurar que estén lo más claro posible, de tal manera que cualquier persona que no tenga relación con el proyecto, pueda comprender en que consiste el trabajo que se va a realizar.

- Relación con otros sistemas. Se debe tener en cuenta que el sistema a desarrollar va a funcionar en conjunto con otros sistemas.

Para el personal del Comité Informático debe quedar claro de que manera el sistema a desarrollar afecta otras aplicaciones que están funcionando, de manera que se puedan tomar previsiones en caso que sea necesario.

- Restricciones o limitaciones para el cumplimiento de requerimientos. Debe estar claro para el Comité Informático que cosas son las que el nuevo sistema no va a considerar, es decir, debe quedar claro el ámbito del trabajo del nuevo sistema. Pero además, es necesario que se sepa que restricciones o limitaciones se pueden tener para el desarrollo del nuevo sistema. Estas limitaciones pueden ser tanto de índole técnica, económica o de factores humanos.
- Prioridad de ejecución de los requerimientos. En ocasiones, especialmente cuando los sistemas a desarrollar son grandes, se debe tener en cuenta que se pueden desarrollar en fases. Esto significa que algunos requerimientos tienen prioridad sobre otros, por tanto se desarrollarán antes. Esto tiene que ser del conocimiento del Comité Informático.
- Aprobación del usuario responsable de la etapa de definición de requerimientos. Una vez que se han presentado las correcciones a la etapa de definición de requerimientos, el usuario responsable procede a aprobarla en caso que este de acuerdo.

Diseño

ACTIVIDAD	RESPONSABLES	INSUMO	PRODUCTOS
DI-01 Definir arquitectura del sistema	<ul style="list-style-type: none"> • A. R. • Grupo analistas 	PR-DR-05. Documento definitivo de requerimientos del sistema	PR-DI-01 Arquitectura del sistema
DI-02 Preparar diseño conceptual	<ul style="list-style-type: none"> • A. R. • Grupo analistas 	PR-DI-01 Arquitectura del sistema	PR-DI-02 Diseño Conceptual
DI-03 Revisión del diseño conceptual por parte del usuario	<ul style="list-style-type: none"> • A. R. • U.R. • Grupo de usuarios 	PR-DI-02 Diseño Conceptual	PR-DI-03 Aprobación usuario del diseño conceptual
DI-04 Preparación del diseño detallado	<ul style="list-style-type: none"> • A. R. • Grupo analistas 	PR-DI-02 Diseño Conceptual PR-DI-03 Aprobación usuario del diseño conceptual	PR-DI-04 Diseño Detallado PR-DR-05. Documento definitivo de diseño
DI-05 Control de calidad de la etapa	<ul style="list-style-type: none"> • A. R. • Grupo analistas • U.R. • Grupo de usuarios • Intendent e Informático • Comité Informático 	PR-DR-05. Documento definitivo de requerimientos del sistema PR-DI-02 Diseño Conceptual PR-DI-04 Diseño Detallado	PR-DI-06 Observaciones y correcciones al documento de diseño
DI-06 Aprobación de usuario responsable de la etapa de diseño	<ul style="list-style-type: none"> • U.R. • A.R. 	PR-DI-06. Documento de diseño corregido	PR-DI-07 Aprobación del diseño por parte del usuario

Tabla 6. Actividades de Diseño.

La tabla 6 muestra las actividades de la fase Diseño, las cuales son descritas a continuación.

- Definir la arquitectura del sistema, es decir, transformar los requerimientos de software en conceptos de diseño. Para realizar esta actividad, los analistas pueden apoyarse en el modelo entidad-relación, que permite conocer entidades, procesos y datos.

- Preparar diseño conceptual. La preparación del diseño conceptual conlleva el desarrollo de una serie de productos, que se reseñan a continuación:

- Preparación de diccionario de datos
- Diagrama entidad-relación
- Descripción de componentes, procesos y datos
- Descripción de interfaces con otros sistemas

- Preparar diseño detallado. La preparación del diseño detallado conlleva el desarrollo de una serie de productos, que se reseñan a continuación:

- Transformación del modelo lógico en un modelo físico. Se considera aspectos de integridad de datos y uso de formulas normales. Esto conlleva el desarrollo de script de bases de datos, descripción de tablas, descripción de procesos, preparación de índices, procedimientos almacenados, requerimientos de auditoría, aspectos de seguridad (triggers, definición de perfiles, matriz de procesos/usuarios y tablas/usuarios).

- Diseñar interfaz de usuario. Descripción de menús, pantallas, reportes, mensajes, manejo de excepciones y manejo de errores.

- Desarrollo de algoritmos (procesos de funciones específicas)

- Control de calidad de la etapa. Esta actividad consiste en analizar la solución presentada por el analista responsable y por el equipo de analistas de apoyo. Lo importante en esta fase es que se puedan analizar los requerimientos definidos (apoyándose en el documento respectivo) y contrastarlos con la solución presentada por los analistas de sistemas. Esto significa hacerse una pregunta: ¿de qué manera se da solución a este requerimiento?. Los analistas explican la forma en como dan solución a un problema y se retroalimentan con la opinión de los miembros del Comité Informático, Intendente Informático, Usuario Responsable y Grupo de usuarios de apoyo.

- Aprobación del usuario responsable de la etapa de diseño. Una vez que se han presentado las correcciones a la etapa de diseño, el usuario responsable procede a aprobarla en caso que este de acuerdo.

Codificación

ACTIVIDAD	RESPONSABLES	INSUMO	PRODUCTOS
CO-01 Crear ambiente de desarrollo	<ul style="list-style-type: none"> • A. R. • Grupo analistas 	PR-DR-05. Documento definitivo de requerimientos del sistema PR-DI-06. Documento de diseño corregido	PR-CO-01 Base de datos de prueba
CO-02 Crear objetos del sistema	<ul style="list-style-type: none"> • A. R. • Grupo analistas 	PR-CO-01 Base de datos de prueba	PR-CO-02 Unidades, procedimientos, bibliotecas de programas
CO-03 Crear programas fuente	<ul style="list-style-type: none"> • A.R. • Grupo analistas 	PR-DR-05. Documento definitivo de requerimientos del sistema PR-DI-06. Documento de diseño corregido	PR-CO-03 Programas fuentes del sistema
CO-04 Pruebas internas a los programas	<ul style="list-style-type: none"> • A. R. • Grupo analistas 	PR-CO-03 Programas fuentes del sistema	PR-CO-04 Listado de errores o problemas detectados en los programas PR-CO-05 Programas fuentes del sistema corregido
CO-05 Control de calidad de la etapa	<ul style="list-style-type: none"> • A. R. • Grupo analistas 	PR-CO-02 Unidades, procedimientos, bibliotecas de programas PR-CO-05 Programas fuentes del sistema corregido	PR-CO-06 Listado de problemas detectados

Tabla 7. Actividades de Codificación.

La tabla 7 muestra las actividades de la fase Codificación, las cuales son descritas a continuación.

- Crear ambiente de desarrollo. Para iniciar el desarrollo de un sistema, es necesario inicialmente crear una base de datos de pruebas, que permitirá crear un ambiente que simula el ambiente sobre el cual correrá finalmente la aplicación. Se aconseja crear los scripts con el modelo físico de base de datos y la carga inicial de tablas de códigos (por ejemplo, tabla de provincias, cantones o distritos, códigos y tipos de producto).
- Crear objetos del sistema. Esta actividad pretende crear objetos de uso general o específico en el sistema, que son independientes de los programas y de los datos. Por lo general, son programas que reciben parámetros de entrada y devuelven datos a los programas que los llaman.

Estos objetos suelen tener un nombre que los identifica, seguido de una lista de parámetros de entrada y salida. Además permite definir sus propias variables de trabajo interno. Es recomendable poner comentarios que indiquen la fecha de creación, una descripción general del propósito de dicho programa, significado de los parámetros de entrada y de salida, comentarios en las líneas que se requiera, explicando lógica que se considere compleja.

- Crear programas fuente. Estos programas son los que le aparecerán al usuario en su menú. Es aconsejable tomar algunas previsiones para su desarrollo:
 - Utilizar estándar creado por la organización para efectos de diseño de pantallas y reportes.

- Utilizar el concepto de programación estructurada.
- Utilizar comentarios donde se considere necesario. La posibilidad de crear comentarios es algo que debe minimizarse. En general, los nombres de variables debe ser lo suficientemente claro como para que se comprenda en que consiste el programa.
 - Utilizar una adecuada indentación en las líneas de código.
 - Todo tipo de rutinas de uso general deben estar en procedimientos almacenados. Por ejemplo, manejo de errores, manejo de excepciones, llamado a reportes y otros. Esto evita repetir codificación.
 - Cuando se realizan modificaciones al código, es recomendable aclarar en que consiste el cambio, la persona y la fecha en que lo realizó.
 - En la medida de lo posible, la lógica de negocios debe implementarse en procedimientos almacenados y no en el código fuente del programa. La razón de esta recomendación es que ante un cambio en la lógica del negocio, no será necesario modificar todos los programas, por tanto, debe buscar mantener separados la lógica del negocio y los datos.
- Pruebas internas a los programas. Las pruebas a los programas debe ser responsabilidad de cada programador. El tiene la responsabilidad de probar todo el código que se tiene en los programas. Con respecto a las pruebas internas, se pueden dar las siguientes recomendaciones:
 - Asegurar que la base de datos de prueba esta creada y que funciona adecuadamente.
 - Redactar todos los casos de prueba que considere necesario,

incluyendo excepciones, manejo de errores y llamado a procedimientos almacenados.

- Probar absolutamente todo el código creado. En ocasiones se crea código que nunca es probado. Es responsabilidad de los analistas asegurarse que todo código creado tiene una razón de ser y por tanto, debe ser probado.

- Crear situaciones límite. Es decir, plantear casos que permitan saber como se comportarán los programas en caso que se presenten fallos no contemplados.

- Probar interfaces con otros sistemas. Los analistas-programadores deben probar todas las interfaces con otros sistemas. De tal manera, que la base de datos de pruebas debe contemplar tablas de otros sistemas en caso que sea necesario. Es importante que se compruebe como un proceso en nuestro sistema puede afectar datos en otros sistemas.

- Probar integralmente todos los programas. La idea de probar cada programa individualmente no significa que se deba realizar una prueba integral al sistema como un todo. Los analistas-programadores deben asegurarse de tener control sobre lo que realiza cada programa y de como se afectan las tablas de datos entre los programas.

- Control de calidad de la etapa. En este caso, el Analista Líder es responsable de lo que realiza su equipo de analistas. En esta fase, es importante realizar una integración en el trabajo que realizan sus subordinados, así como aclarar conceptos, revisar uso de estándares de programación, no duplicar código ni esfuerzo, uso de bibliotecas y subprogramas por parte de todos los analistas en sus programas y corregir problemas de comunicación detectados durante el desarrollo. Además,

deben analizarse los errores presentados en las pruebas realizados por cada analista, ver como afecta el trabajo que realiza los otros analistas y asegurarse que todos los módulos y programas realizados funcionen integralmente como un todo.

Pruebas

ACTIVIDAD	RESPONSABLES	INSUMO	PRODUCTOS
PR-01 Crear casos de prueba	<ul style="list-style-type: none"> • U.R. • Grupos de usuarios 	PR-DR-05. Documento definitivo de requerimientos del sistema	PR-CO-01 Casos de prueba
PR-02 Crear datos de prueba	<ul style="list-style-type: none"> • U.R. • Grupos de usuarios • A.R. Grupo analistas 	PR-CO-01 Casos de prueba	PR-CO-02 Datos de prueba
PR-03 Ejecución de pruebas	<ul style="list-style-type: none"> • U.R. • Grupos de usuarios • A.R. • Grupo analistas 	PR-CO-01 Casos de prueba PR-CO-02 Datos de prueba	PR-CO-03 Resultados de pruebas
PR-04 Revisión de resultados de pruebas	<ul style="list-style-type: none"> • A. R. • Grupo analistas • U.R. • Grupos de usuarios 	PR-CO-03 Resultados de pruebas	PR-CO-04 Listado de errores o problemas detectados en los programas PR-CO-05 Programas fuentes del sistema corregido
PR-05 Control de calidad de la etapa	<ul style="list-style-type: none"> • A. R. • Grupo analistas • U.R. • Grupos de usuarios 	PR-CO-02 Unidades, procedimientos, bibliotecas de programas PR-CO-05 Programas fuentes del sistema corregido	PR-CO-06 Listado de problemas detectados
PR-06 Aprobación de la etapa de pruebas	<ul style="list-style-type: none"> • U.R. 	PR-CO-06. Documento de diseño corregido	PR-CO-07 Aprobación de las pruebas por parte del usuario

Tabla 8. Actividades de Pruebas.

La tabla 8 muestra las actividades de la fase Pruebas, las cuales son descritas a continuación.

- Crear casos de prueba. Los usuarios deben crear todos los casos de prueba que se consideren importante. En esta fase, es bueno que se tome el documento de definición de requerimientos, se analice la forma apropiada de probar cada requerimiento (es decir, qué pruebas funcionales deben hacerse sobre los programas para asegurarse que el requerimiento ha sido cumplido).
- La ayuda y guía de personal técnico en este punto es muy importante. Los casos de prueba generalmente son de índole funcional (es decir, se desea probar que el programa regrese los resultados esperados); sin embargo, no solo la funcionalidad debe ser probada en esta fase, sino que también hay que realizar pruebas de rendimiento (donde sea importante), pruebas de interfaz y pruebas de integración.
- Las pruebas de rendimiento permiten determinar los tiempos de respuesta de los sistemas. En ocasiones, es parte de los requerimientos que los programas no tarden más de un tiempo determinado, pero en otras ocasiones, los usuarios no dan un tiempo determinado de respuesta. Sin embargo, estas pruebas deben realizarse para verificar que los tiempos de respuesta de los sistemas son adecuados para los usuarios.
- Las pruebas de interfaz son el manejo de la interfaz de los programas. Estas pruebas permiten ver la facilidad de uso, la existencia de listas de valores en los programas, pruebas de valores límites en los campos.

- Las pruebas de integración se realizan en forma incremental. Es decir, no se prueba módulo por módulo en forma separada, sino que se prueba un módulo y cuando se prueba otro, se prueba conjuntamente los dos módulos, con el fin de verificar que los resultados entre programas sean correctos y consistentes.

- Los usuarios deben estar involucrados en estas pruebas y especialmente, los usuarios que serán encargados de utilizar los sistemas.

- Crear datos de prueba. Es responsabilidad de los usuarios crear datos de prueba (por ejemplo, archivos), siguiendo algún formato definido por los analistas de sistemas. En caso que por limitaciones de operación del programa no sea posible para los usuarios crear los datos de prueba, éstos serán creados por los analistas de sistemas.

- Ejecución de pruebas. La ejecución de pruebas se realiza conjuntamente entre los usuarios y los analistas. Los analistas aseguran que el ambiente funciona adecuadamente y certifican que los usuarios aplican todos los casos, siguiendo la operación adecuada del sistema. Los analistas además toman nota de los resultados obtenidos en las pruebas para efectos de retroalimentación y corrección.

- Revisión de resultados de pruebas. Después de realizar las pruebas, viene la etapa de revisión de los resultados de la etapa. Se analizan que errores se presentaron, las causas, la completitud de las pruebas, la forma adecuada de ejecución de pruebas, problemas presentados durante el desarrollo de las pruebas y cualquier otro factor que sea relevante.

Asimismo se analizan los resultados, con el fin de verificar si los

resultados son aceptables desde el punto de vista de funcionalidad, rendimiento, interfaz e integración.

- Control de calidad de la etapa. El equipo de desarrollo (el analista responsable y el grupo de analistas de apoyo) revisan conjuntamente que problemas se presentaron, las causas de los mismos y la forma de corregirlos.

- Los lineamientos generales de la corrección deben darse en equipo y no individualmente por parte de cada analista. Esto se debe a que la codificación se trato de realizar en forma conjunta, especialmente lo que es el desarrollo de la lógica del negocio. Por tal motivo, los problemas presentados se analizan conjuntamente, se asegura que los cambios se realicen en todos los módulos que deban realizarse. Posteriormente y con las guías dadas por el Analista Líder, todos el equipo de analistas realizan las correcciones individualmente.

- Aprobación del usuario responsable de la etapa de pruebas. Una vez que se han presentado las correcciones a la etapa de pruebas, el usuario responsable procede a aprobarla en caso que este de acuerdo.

Implementación

ACTIVIDAD	RESPONSABLES	INSUMO	PRODUCTOS
IM-01 Plan de instalación sistema	A.R.		PR-IM-01 Plan de instalación del sistema
IM-02 Preparar base de datos de producción	D.B.A.	PR-DI-04 Diseño Detallado	PR-IM-02 Base de datos de producción
IM-03 Definir esquema de seguridad en B.D.	<ul style="list-style-type: none"> • U.R. • A.R. • D.B.A. 	PR-DI-04 Diseño Detallado	PR-IM-03 Esquema de seguridad de base de datos
IM-04 Definir esquema de seguridad en la red	<ul style="list-style-type: none"> • A. R. • Adm. Red 	PR-DI-04 Diseño Detallado	PR-IM-04 Esquema de seguridad en la red
IM-05 Instalación de hardware y software	<ul style="list-style-type: none"> • Soporte Técnico • Grupo analistas 		PR-IM-05 Sistema en producción
IM-07 Capacitación a usuarios	<ul style="list-style-type: none"> • U.R. • Grupo usuarios • A.R. • Grupo analistas 	PR-IM-03 Esquema de seguridad de base de datos	
IM-08 Operación inicial del sistema	<ul style="list-style-type: none"> • U.R. • Grupo usuarios • A.R. • Grupo analistas 	PR-IM-06 Sistema actual	PR-IM-07 Errores detectados en el sistema
IM-09 Corrección de ambiente de producción	<ul style="list-style-type: none"> • A.R. • Soporte Técnico • D.B.A. • Adm. Red 	PR-IM-07 Errores detectados en el sistema	PR-IM-08 Sistema en producción corregido
IM-10 Generar aceptación por parte del usuario	<ul style="list-style-type: none"> • U.R. • A.R. 	PR-IM-08 Sistema en producción corregido	PR-IM-09 Aprobación de la operación inicial del sistema

Tabla 9. Actividades de Implementación.

La tabla 9 muestra las actividades de la fase Implementación, las cuales son descritas a continuación.

- Plan de instalación del sistema. La primera actividad de la implementación consiste en definir un plan de acción para poner en funcionamiento el sistema. Estas son las actividades importantes que se establecen en el plan de implementación:

- Preparar base de datos de producción.
- Definir esquema de seguridad en la base de datos.
- Definir esquema de seguridad en la red.
- Instalación de hardware y software.
- Capacitación a usuarios.
- Operación inicial del sistema.
- Corrección de ambiente de producción.

Estas actividades son detalladas a continuación.

- Preparar base de datos de producción. Consiste en crear el esquema de base de datos definido en el diseño físico del sistema. Esta tarea es realizada por el Administrador de la Base de Datos (DBA), el cual corre el script de base de datos, estableciendo las relaciones entre las tablas del esquema creado y tablas de otros sistemas, dando o restringiendo accesos sobre tablas y otros objetos, tales como procedimientos almacenados.

- Definir esquema de seguridad en la base de datos. Esta tarea consiste en definir la seguridad de acceso sobre los objetos de las bases de datos y sobre los programas. Esta tarea es responsabilidad del Usuario

Responsable con la ayuda del analista de sistemas, el cual proporciona al usuario la lista de programas del sistema y el usuario responsable define quién tiene acceso a que programa. Con esta información, el analista le prepara al DBA una tabla con permisos sobre objetos (tablas, procedimientos) y programas de usuario final.

- Definir esquema de seguridad en la red. Esta tarea es ejecutada por el Administrador de la Red. Esto implica crear niveles de seguridad de acceso a nivel del sistema operativo, creando usuarios a nivel de red, estableciendo sus accesos a recursos de la red.

- Instalación de hardware y software. Esta tarea es generalmente realizada por el personal de soporte técnico (instala hardware) y el analista de sistemas (instala software de aplicación).

- Capacitación a usuarios. Esta tarea es coordinada por el Usuario Responsable y el Analista Responsable. El Usuario Responsable define la lista de personal que necesita aprender el uso del sistema. Es conveniente que la capacitación a los usuarios se de por perfiles de usuario, esto significa que los usuarios que realizan la misma tarea reciben la capacitación al mismo tiempo y que su capacitación no sea al mismo tiempo para otro grupo de usuarios que realizan un grupo de tareas diferentes.

- Operación inicial del sistema. Una vez que el sistema ha sido instalado y que los usuarios han recibido la capacitación, inicia la operación del sistema. En ocasiones esto implica que los usuarios sigan utilizando el sistema automatizado anterior o en caso que no existiera sistema, realizan manualmente la operación de los datos y posteriormente los resultados son

comparados con los resultados del nuevo sistema implementado.

En caso que exista alguna de las situaciones descritas en el párrafo anterior, es necesario que se establezca un tiempo prudencial de llevar en paralelo el sistema anterior (manual o automatizado) y el sistema nuevo implementado. Durante ese tiempo se debe llevar una comparación periódica de los resultados de los sistemas para proceder a corregir los errores de cálculo presentados.

- Corrección de ambiente de producción. De la operación inicial del sistema, se dan las correcciones en el ambiente de producción, que puede ser por problemas presentados en los programas fuente, problemas de permisos sobre objetos de la base de datos o permisos sobre la red.
- Generar aceptación por parte del usuario. Una vez que las pruebas de operación iniciales han sido superadas a satisfacción del usuario, el usuario da por aceptado el sistema de información. Con esta aceptación, se da por terminado el desarrollo del sistema.

Post-Implementación

ACTIVIDAD	RESPONSABLES	INSUMO	PRODUCTOS
PI-01 Evaluar problemas y situaciones presentadas	<ul style="list-style-type: none"> • U.R. • Grupo analistas • Intendente Informático 	Toda documentación y hechos generados del desarrollo del sistema	PR-PI-01 Evaluación del desarrollo del sistema
PI-02 Registrar resultados y costos reales del sistema	<ul style="list-style-type: none"> • U.R. • Grupo analistas • Intendente Informático 	Toda documentación y hechos generados del desarrollo del sistema	PR-PI-02 Registro de resultados reales del desarrollo del sistema
PI-03 Establecer métricas	<ul style="list-style-type: none"> • U.R. • Grupo analistas • Intendente Informático 	PR-PI-01 Evaluación del desarrollo del sistema PR-PI-02 Registro de resultados reales del desarrollo del sistema	PR-PI-03 Métricas de desarrollo

Tabla 10. Actividades de Post-Implementación.

La tabla 10 muestra las actividades de la fase Post-Implementación, las cuales son descritas a continuación.

- Evaluar problemas y situaciones presentadas. Se analizan todas las situaciones presentadas durante el desarrollo del sistema y que provocaron problemas durante el desarrollo de la misma. La idea es que los grupos puedan corregir y evaluar cualquier situación con el fin que situaciones problemáticas no se vuelvan a presentar en los proyectos futuros.

- Registrar resultados y costos reales del sistema. Se registran resultados reales del sistema, tiempo de duración real de las actividades,

recursos reales asignados en las actividades y costo real del proyecto.

- Establecer métricas. Con toda la información recolectada, se puede empezar a establecer métricas. Estas métricas pueden ser establecidas por la Intendencia Informática de la organización, con el fin de establecer parámetros de medición real y adecuados a la organización.

Es de entender que el establecimiento de métricas no es una tarea que se haga de la noche a la mañana, sino que más bien es un proceso que se desarrolla con el tiempo.

Finalmente, se quiere destacar la importancia de esta etapa, la cual en muchas ocasiones no se realiza en las organizaciones. La única forma de corregir los problemas es evaluando y estableciendo métricas reales que permitan en un futuro a la organización desarrollar sistemas en menor tiempo, con una mejor asignación de recursos.

Documentación

La documentación es desarrollada durante todas las actividades del sistema. En el anexo 5 se describe en términos generales, el contenido de cada documento de la metodología propuesta.

Mantenimiento

ACTIVIDAD	RESPONSABLES	INSUMO	PRODUCTOS
MA-01 Identificación y clasificación del problema o modificación	<ul style="list-style-type: none"> • U.R. • A.R. • Grupo analistas 	Solicitud de cambios por parte del usuario	PR-MA-01 Documento de cambio solicitado por usuario
MA-02 Análisis del cambio	U.R. A.R. Grupo analistas	PR-MA-01 Documento de cambio solicitado por usuario PR-DR-05. Documento definitivo de requerimientos del sistema	PR-MA-02 Variaciones al documento definitivo de requerimientos del sistema
MA-03 Diseño del cambio	U.R. A.R. Grupo analistas	PR-MA-02 Variaciones al documento definitivo de requerimientos del sistema PR-DI-06. Documento de diseño corregido	PR-MA-03 Variaciones al documento de diseño
MA-04 Codificación del cambio	U.R. A.R. Grupo analistas	PR-CO-02 Unidades, procedimientos, bibliotecas de programas PR-CO-05 Programas fuentes del sistema corregido	PR-MA-04 Variaciones a programas fuente, unidades, procedimientos y bibliotecas de programas

MA-05 Pruebas del cambio	U.R. A.R. Grupo analistas	PR-MA-04 Variaciones a programas fuente, unidades, procedimientos y bibliotecas de programas	PR-MA-05 Documento con resultados de pruebas a los cambios realizados
MA-07 Aceptación del cambio por parte del usuario	U.R. A.R. Grupo analistas	PR-MA-05 Documento con resultados de pruebas a los cambios realizados	PR-MA-06 Aprobación del cambio por parte del usuario
MA-08 Implementación del cambio	U.R. A.R. Grupo analistas	PR-MA-06 Aprobación del cambio por parte del usuario	PR-MA-07 Programas cambiados en producción
MA-08 Documentación del cambio	U.R. A.R. Grupo analistas		PR-MA-08 Modificaciones a programas cambiados en producción

Tabla 11. Actividades de Mantenimiento.

La tabla 11 muestra las actividades de la fase Mantenimiento, las cuales son descritas a continuación.

- Identificación y clasificación del problema o modificación. Establecer la naturaleza del cambio y la forma en como afecta el sistema. Por otra parte, se debe determinar que tipo de cambio es (correctivo, preventivo) con el fin de determinar si el cambio se debe a problemas presentados en los resultados, si hubo un cambio en los requerimientos del usuario y cuando fue la última vez que se dio un cambio en el sistema.
- Análisis del cambio. Se realiza un análisis similar a la que se presento en la fase de Definición de Requerimientos. Se analiza el impacto del cambio en los requerimientos existentes y en como afecta esto al sistema

como un todo.

- **Diseño del cambio.** Se realiza un diseño para darle solución a los nuevos requerimientos del usuario. Hay que tomar en cuenta que esto puede significar un cambio en las estructuras de datos existentes, en el modelo entidad-relación de la base de datos, en las interfaces de los sistemas, en el diccionario de datos y en las especificaciones lógicas.

- **Codificación del cambio.** Se realizan la codificación de los cambios en los programas, manteniendo los mismos cuidados que se reseñaron en la etapa de codificación expuesta arriba.

- Especialmente importante es poner comentarios que indiquen la naturaleza, persona, fecha y explicación del cambio, especialmente si esto viene a cambiar la forma en como se realizaban las cosas anteriormente.

- **Pruebas del cambio.** Se realizan pruebas del analista a los cambios realizados y cuando este seguro que funciona bien, pasa a realizar las pruebas de usuario. En estos casos se siguen las mismas recomendaciones que en el apartado de pruebas de la metodología, lo cual implica desarrollar también un guión de pruebas, que incluya casos que vengán a probar los cambios y también otras situaciones que se puedan generar de estos cambios.

- **Aceptación del cambio por parte del usuario.** Una vez que el usuario ha revisado los cambios, debe darlos por aceptado.

- **Implementación del cambio.** Se pone en funcionamiento el nuevo

programa, siguiendo las mismas recomendaciones dadas en la fase de implementación de la metodología.

- Documentación del cambio. Existen sistemas que por su naturaleza cambian mucho, en ocasiones son asuntos de forma pero en otras son asuntos de fondo. Lo que se recomienda en este caso es llevar estadísticas de la frecuencia y naturaleza de los cambios en los sistemas. Esto permitiría en un futuro rediseñar una aplicación conociendo a fondo como ha venido cambiando ésta durante el tiempo.

Capítulo VI

Conclusiones y Recomendaciones

Conclusiones

- La metodología de desarrollo de sistemas actual de la SUPEN presenta una serie de problemas que afectan el desarrollo de los sistemas, tales como:
 - Muy general, sin una clara definición de responsabilidades y sin una clara relación entre productos y actividades.
 - Es desconocida por parte del usuario.
 - Existen algún problema de coordinación en la aplicación de la metodología, tal es el caso de la documentación externa de los sistemas, la cual es inexistente.
 - La aplicación de la metodología se enfoca a probar la funcionalidad de los sistemas como único parámetro de medición.
 - Desde el punto de vista de los usuarios, los sistemas presentan algunos problemas de calidad, que aunque no son achacables directamente a la metodología, si puede verse como problemas en la definición de requerimientos de velocidad (tiempos de respuesta de los sistemas son lentos), requerimientos de interfaz (el manejo de errores por parte de los sistemas no es amigable) y requerimientos de adaptabilidad. (incapacidad de adaptar rápidamente los sistemas a los cambios presentados en el sector de pensiones).
- Después de haber analizado las metodologías utilizadas de la IEEE y de la ISO, fue posible observar que la metodología de desarrollo de sistemas presenta problemas de control de calidad tanto en las fases de desarrollo como en la manera de enfocar el proyecto como un todo.

- La metodología que se propone en conjunto con las conclusiones que se presentan, permite el cumplimiento de los objetivos trazados al inicio del trabajo.

- El aporte principal de la metodología propuesta consiste en reforzar los aspectos en los cuales se han encontrado problemas. En este sentido, la metodología establece actividades claramente definidas, se establecen responsables de actividades y productos que deben ser generados en cada etapa (esto con el fin de controlar el desarrollo de la actividad en sí misma), asimismo establece una actividad de control de calidad al final de cada etapa. Por otra parte, todas las fases de desarrollo están integradas, especialmente la fase definición de requerimientos- diseño y pruebas.

Recomendaciones

Algunas recomendaciones que plantea este estudio, son las siguientes:

- La implementación de la metodología propuesta, la cual trata de enfocarse a reforzar los problemas de control de responsabilidad(es decir, la aplicación de roles para el personal) y productos, identificados en cada fase de desarrollo de los sistemas (análisis, diseño, codificación, pruebas, implementación, documentación). Al mismo tiempo, se incorpora en forma puntual en la metodología las fases de post-implantación (como una forma de retroalimentación del proyecto realizado) y mantenimiento (como una forma de controlar y documentar el mantenimiento que se vaya realizando a los sistemas en operación).
- Es importante involucrar a todos los usuarios que tengan un grado de responsabilidad en relación a los sistemas (niveles gerenciales y direcciones de división y departamento), especialmente que conozcan la metodología y su papel en la misma, para que puedan apoyar al área informática durante el desarrollo de los sistemas.
- Se recomienda utilizar los estándares de programación e interfaz (ver anexo 2) actual. En este sentido, también se debe controlar que los estándares no solo se apliquen, sino que se haga correctamente.
- A medida que se avance en la aplicación de la metodología de desarrollo de sistemas propuesta, es necesario avanzar en el aspecto de la calidad. En este sentido, la fase de post-implementación debe ser la oportunidad para analizar concienzudamente los problemas que se presentaron en los sistemas y la forma en como deben enfrentarse a futuro. Por otra parte, es la oportunidad para

empezar a desarrollar métricas de calidad, no enfocadas únicamente a la funcionalidad, sino que se pueden reforzar otros aspectos del desarrollo de los sistemas. Esto sin embargo, debe comprenderse que se irá haciendo poco a poco, no es una tarea que en este momento sea posible realizar.

- La idea de establecer métricas es prever que se puede hacer ante ciertos problemas que se presentan. Es decir, tomar acciones correctivas que vengán a reforzar las áreas que presentan problemas.

- Se recomienda establecer un plan de verificación y validación en la organización. Esto es, que se desea revisar de los sistemas y cuales son los criterios por medio de los cuales se da por aceptado un producto. Esta tarea debe ser impulsada por la Intendencia Informática.

- Se recomienda en el futuro implementar auditorías a los sistemas en uso. Para esto es necesario establecer en conjunto con la Auditoría Interna de la SUPEN, una forma de revisar los requerimientos de los sistemas con lo que realmente esta implementado.

Capítulo VII

Bibliografía

1. **[BEDINI 89]** BEDINI. Inspecciones Formales: Un proceso para asegurar la calidad del software. 1989.
2. **[CADENA 96]** Cadena, Eduardo. ISO 9000: Una visión general. Revista Soluciones Avanzadas No 32, abril,1996.
3. **[CGR 95]** Contraloría General de la República. “Manual sobre Normas Técnicas de Control Interno Relativas a los sistemas de Información Computarizados”. Noviembre, 1995.
4. **[CRO 79]** Crosby,P. “Quality is Free”, McGraw-Hill, 1979.

5. **[DAICH 94]** Daich. “Software Process, Quality & ISO 9000”. ISSN: 1070-5457. Volumen 4, Número 5. Mayo, 1994.
(<http://www.integratise.com/spqsmp.html>)
6. **[Evenau 94]** R.G. Evenau, S.H. Strausss. “Software Inspections Process”. McGraw-Hill, 1994.
7. **[HUMP 95]** Humphrey W. “Software Standards”. Adison-Wesley, 1995.
Capítulo 9
8. **[IEEE 84]** Institute of Electrical and Electronics Engineers. “IEEE Guide to Software Requirements Specifications”. New York, USA, IEEE Standards Board, Julio, 1984.
9. **[IEEE 87]** Institute of Electrical and Electronics Engineers. “IEEE Standard for Software Project Management Plans”. New York, USA, IEEE Standards Board, Diciembre, 1987.
10. **[IEEE 87-1]** Institute of Electrical and Electronics Engineers. “IEEE Standard for Verification an Validation Plans”. New York, USA, IEEE Standards Board, 1987.
11. **[IEEE 87-2]** Institute of Electrical and Electronics Engineers. “IEEE Standard for Software Unit Testing”. New York, USA, IEEE Standards Board,

1987.

12. **[IEEE 88]** Institute of Electrical and Electronics Engineers. "IEEE Standard for Software Reviews and Audits". New York, USA, IEEE Standards Board, 1988.

13. **[IEEE 89]** Institute of Electrical and Electronics Engineers. "IEEE Standard for Software Quality Assurance Plans". New York, USA, IEEE Standards Board, Agosto, 1989.

14. **[IEEE 90]** Institute of Electrical and Electronics Engineers. "IEEE Standard for Software Configuration Management Plans". New York, USA, IEEE Standards Board, 1990.

15. **[IEEE 93]** IEEE Standards Collection. Software Engineering. 1993 Edition, IEEE Inc.

16. **[ISO 91]** ISO.ISO 9000-3. "Guidelines for the application of ISO 9001 to the Development, Supply and Maintenance of Software". ASQC Press, 1991.

17. **[ISO 93]** Colección de estándares ISO. Software Engineering. 1993. Edition. Normas ISO 9000

18. **[ISO 94]** ISO.ISO9001: "Quality Systems-Model for Quality Assurance in Design, Development, Production, Installation and Servicing". ASQC Press,1994.

19. **[JAGUADO 89]** Jaguado,R. ISO 9000: Algunas preguntas, respuestas y conceptos básicos. 1989.

20. **[JENKINS 96]** Jenkins, Marcelo. The User´s Role in Software Quality. Proceedings Sixth International Conference on Software Quality, Montreal, Canadá, 1996.

21. **[NASA-GB 97]** Nasa, Software Assurance Guidebook, 1997.

22. **[PAULK 95]** M.C. Paulk, C.V. Weber, B.Curtis, M.B. Chrissis. "The Capability Maturity Model". Adison-Wesly,1995.

23. **[PRESS 93]** Pressman, Roger . Ingeniería del Software. Un enfoque

práctico. Tercera edición, Editorial McGraw Hill/Interamericana de España, S.A., 1993.

24. **[PRESS 98]** Pressman, Roger. Ingeniería del Software. Cuarta Edición, Editorial McGraw Hill/Interamericana de España, 1998.

25. **[Soluciones Avanzadas 96]** Cadena Gómez, Eduardo. "ISO 9000, Una vision general". Revista Soluciones Avanzadas, No 32. Abril, 1996.

26. **[SUPEN 96]** Superintendencia de Pensiones. "Metodología para el Desarrollo de Sistemas". Versión 1.0. Julio 1996.

27. **[YOURDON 89]** Yourdon, E. Structured Walkthroughs, 4ta edición, Yourdon Press (Prentice-Hall), 1989. Capítulo 8: "Software Quality Assurance".