



CIMAT

Centro de Investigación en Matemáticas, A.C.

**Estandarización del proceso de
pruebas en *DDSI-DGAI del INEGI:
Mejorando la calidad de los
productos de software generados**

T E S I S

Que para obtener el grado de
Maestro en Ciencias en Estadística Oficial

Presenta:

Andrés Francisco Rios Ramírez

Director de Tesis:

Dr. Cuauhtémoc Lemus Olalde

Co - Director de Tesis:

M. en E. Sergio Martín Nava Muñoz

Guanajuato, Gto. Diciembre de 2009

INDICE

RESUMEN	1
1.- INTRODUCCIÓN.....	4
1.1 ANTECEDENTES	4
1.2 RESPONSABILIDADES Y NECESIDADES DEL DEPARTAMENTO DE PRUEBAS MODULARES A SISTEMAS DE INFORMACIÓN (ESTADO ACTUAL).....	10
1.3 EVOLUCIÓN DEL PROCESO DE PRUEBAS A UN PROCESO DE PRUEBAS ESTÁNDAR	10
1.4 MOTIVACIÓN INICIAL DE HABER CONTACTADO AL ASESOR.	14
1.5 ORGANIZACIÓN DEL DOCUMENTO	14
2.- JUSTIFICACIÓN.....	17
2.1 HISTORIA DEL DEPARTAMENTO DE PRUEBAS MODULARES A LOS SISTEMAS DE INFORMACIÓN	17
2.2 ACTIVIDADES ACTUALES DEL DEPARTAMENTO DE PRUEBAS MODULARES A SISTEMAS DE INFORMACIÓN DE LA DDSI DE LA DGAI (INEGI)	18
2.3 CAMBIO DE RUMBO EN LA INVESTIGACIÓN.....	20
3.- PROCESO DE PRUEBAS DE SOFTWARE: PROPUESTA PARA LA DDSI DE LA DGAI(INEGI)	25
3.1 GUÍA METODOLÓGICA.....	25
3.2 APOYOS EN LA DOCUMENTACIÓN Y EVALUACIÓN DEL PROCESO DE PRUEBAS: CMAPTOOLS Y AMEF.....	26
3.2.1 Cmaptools [28]	27
3.2.2 AMEF	29
3.2.2.1 Limitaciones en la aplicación de la herramienta en esta tesis.....	35
4.-CASO DE ESTUDIO.	41
4.1 INFORMACIÓN DEL PROYECTO SISTEMA INTEGRAL DE ADMINISTRACIÓN (SIA) - MÓDULO DE CONTABILIDAD.....	42
4.1.1 El proceso aplicado al SIA - Módulo de Contabilidad.	42
4.2 INFORMACIÓN DEL PROYECTO CONTROL DE PRODUCCIÓN Y VENTAS DEL INEGI (CONPROVE).	47
4.2.1 El proceso aplicado al ConProVe (Control de Producción y Ventas del INEGI)....	49
5.- RESULTADOS, CONCLUSIONES Y TRABAJO FUTURO.....	54

5.1 RESULTADOS	54
5.2 CASO CONPROVE.....	57
5.3 CASO SIA - CONTABILIDAD:	61
5.4 CONCLUSIONES.....	65
5.5 TRABAJO FUTURO	66
5.6 BENEFICIOS.....	68
ANEXOS.....	69
ANEXO 1 – MONITOREO DEL PROCESO DE PRUEBAS.....	70
ANEXO 2 GUÍA METODOLÓGICA DEL PROCESO DE PRUEBAS PARA LA REVISIÓN DE SISTEMAS DE SOFTWARE. VERSIÓN 0.2, 04/06/2008.....	93
ANEXO 3.- DATOS ESTADÍSTICOS Y SCRIPTS DE R.....	138
ANEXO 4 JUSTIFICACIÓN DEL CAMBIO DE RUMBO EN LA INVESTIGACIÓN DEL TEMA DE TESIS.....	152
ANEXO 5. DATOS OBTENIDOS DE LAS PRUEBAS DEL VIII CENSO AGROPECUARIO 2007.....	153
ANEXO 6. ANALISIS DE MODOS Y EFECTOS DE FALLAS POTENCIALES (AMEF) [26].	156

INDICE DE FIGURAS

Figura 1.1 Estructura del INEGI.....	8
Figura 3. 1 Ejemplo de Mapas Conceptuales	28
Figura 4. 1 Sistema Integral de Administración del INEGI	42
Figura 4. 2 Prueba del Sistema (PS) del Proyecto SIA (Módulo de Contabilidad) producto de la aplicación del Proceso de Pruebas estandarizado basado en la Guía Metodológica del Proceso de Pruebas para la revisión de Sistemas de software y documentado con la herramienta Cmaptools.	46
Figura 4. 3 Prueba de Aceptación sistema del Proyecto SIA (Módulo de Contabilidad) producto de la aplicación del Proceso de Pruebas estandarizado basado en la Guía Metodológica del Proceso de Pruebas para la revisión de Sistemas de software y documentado con la herramienta Cmaptools.	47
Figura 4. 4 Módulos de ConProVe.....	49
Figura 4. 5 Prueba del Sistema (PS) del Proyecto ConProVe producto de la aplicación del Proceso de Pruebas estandarizado basado en la Guía Metodológica del Proceso de Pruebas para la revisión de Sistemas de software y documentado con la herramienta Cmaptools.....	52
Figura 4. 6 Prueba de Aceptación sistema del Proyecto ConProVe producto de la aplicación del Proceso de Pruebas estandarizado basado en la Guía Metodológica del Proceso de Pruebas para la revisión de Sistemas de software y documentado con la herramienta Cmaptools.....	53

INDICE DE TABLAS

Tabla 3. 1 Criterios de la evaluación y sistema de graduación sugeridos para la severidad de los efectos para un diseño AMEF	161
Tabla 3. 2 Criterios de la evaluación y sistema de graduación sugeridos para la severidad de los efectos para un proceso AMEF	162
Tabla 3. 3 Criterios de la evaluación y sistema de graduación sugeridos para la ocurrencia del incidente en un diseño AMEF	164
Tabla 3. 4 Criterios de la evaluación y sistema de graduación sugeridos para la ocurrencia del incidente en un proceso AMEF	165
Tabla 3. 5 Criterios de la evaluación y sistema de graduación sugeridos para la detección de una causa del incidente o del modo de fallo en un diseño AMEF	166
Tabla 3. 6 Criterios de la evaluación y sistema de graduación sugeridos para la detección de una causa del incidente o del modo de fallo en un proceso AMEF	167
Tabla 3. 7 Criterios de la evaluación y sistema de graduación sugeridos para la severidad de efectos en un proceso AMEF (Caso INEGI)	37
Tabla 5. 1 Información de defectos del ConProVe.....	57
Tabla 5. 2 Información de defectos del SIA-Contabilidad	61

RESUMEN

La ejecución de la fase de pruebas es uno de los procesos finales del desarrollo de software¹ [27], sin embargo este proceso no inicia cuando el producto de software ha sido terminado. Los probadores de software deben estar involucrados en todas las Fases del Desarrollo, desde el Análisis pasando por el Diseño y hasta su Liberación. Además en ocasiones cuando los proyectos están contra el tiempo, los procesos de pruebas se llevan a cabo de manera deficiente, meramente por cumplir con el requisito, esto tiene por consecuencia que el tratar de dar calidad al producto una vez que este ha sido terminado no se logre totalmente el objetivo. El proceso de pruebas es esencial y de impacto en la entrega de un producto de software con calidad, verificando el cumplimiento con los requerimientos funcionales de los clientes o usuarios, además de garantizar el desempeño de acuerdo a las especificaciones de Producción.

Cuando no se llevan a cabo las pruebas de manera correcta, los defectos que se han pasado por alto, aparecen durante la operación del producto, y por consiguiente se tiene un alto costo de mantenimiento y la insatisfacción del cliente o usuario.

Con una revisión formal basada en un proceso estable de pruebas, bien definido y estandarizado, personas diferentes al autor del producto de software, con la experiencia adecuada en pruebas de sistemas (testers), pueden ayudar a identificar los defectos que se han pasado por alto antes de ser liberado.

Los objetivos principales de este trabajo de investigación son en esencia:

- 1) Establecer las bases metodológicas para realizar un **Proceso de Pruebas** que estandariza las actividades de pruebas en el Departamento de Pruebas Modulares a los Sistemas de Información de la Dirección de Desarrollo de Sistemas de Información (DDSI) de la Dirección General de Innovación y Tecnologías de la Información (DGAI) dentro del INEGI (Instituto Nacional de Estadística y Geografía), para los sistemas o productos de software que sean recibidos en dicho departamento.

¹ Guía Metodológica para la Fase de Pruebas en el desarrollo de software versión 1.0 (GM_10_Pruebas, Iteración_07_Pruebas) del Proyecto TRITON (CIMAT).

2) Creación de la primera versión de una **Guía metodológica del Proceso de Pruebas** para la revisión de sistemas de software (referenciada por el **Proceso de Pruebas**). Esto ayudará al probador de productos de software a identificar el método de revisión adecuado y realizar las pruebas a los sistemas que se reciban en el Departamento de Pruebas Modulares a los Sistemas de Información. Lo anterior es con el afán de proveer las bases para generar productos de software con calidad, ya que se observó que no se seguía un proceso de pruebas estándar.

3) Como resultado de lo anterior, proponer la primera versión de un procedimiento de recolección de datos de defectos del software desarrollado en la Dirección de Desarrollo de Sistemas de Información (DDSI) de la Dirección General Adjunta de Informática de Información (DGAI). Esta primera versión tiene como fundamento el conteo de defectos encontrados por unidad de tiempo, durante las pruebas de un sistema, para que en un futuro se puedan aplicar técnicas estadísticas para obtener indicadores de calidad. Además, la metodología incluye la técnica AMEF (Análisis de Modo de Fallos)² [25], la cual se adaptó en algunos puntos para evaluar los defectos de software encontrados durante las pruebas, como resultado se priorizan los defectos para su atención por los encargados del sistema.

El alcance de este trabajo es:

Se pretende que todos los Sistemas que se reciban en el Departamento de Pruebas Modulares en la Dirección de Desarrollo de Sistemas de Información (DDSI) de la Dirección General de Innovación y Tecnologías de Información (DGAI), se revisen mediante un **Proceso de Pruebas** propuesto en este trabajo, el cual dependerá de los insumos proporcionados para su revisión y el tipo de sistema a probar y se basará en primer término en la documentación del proceso. Se utilizará un Mapa Conceptual que indique en forma general los pasos a seguir para realizar las Pruebas del Sistema (PS) y las Pruebas de Aceptación (PA).

En segundo término seguirá el Procedimiento más adecuado definido en la **Guía metodológica del Proceso de Pruebas**, mencionado anteriormente.

Cabe mencionar que las pruebas que se realizan se enfocan únicamente a la revisión de los sistemas y productos de software utilizando la técnica de **Caja Negra**, no a la

² Manual del AMEF, <http://www.monografias.com/trabajos27/manual-amef>, **Armando Hidalgo Mascorro** mando_hidalgo@yahoo.com.mx.

revisión de líneas de código (**Pruebas Caja Blanca**). Sin embargo, esta última técnica de revisión también se comenta en la Guía Metodológica del Proceso de Pruebas en el Anexo 2.

Además, con la inclusión de la técnica de Análisis de Modo de Fallos (AMEF) se sienta un precedente muy importante al incluir en los reportes de defectos del sistema que se está probando con cortes por ciclo de prueba, un indicador para la atención de los defectos por prioridad (npr – numero de prioridad), por parte de los encargados del proyecto, de acuerdo a tres cualidades de los defectos encontrados (las cuales la técnica maneja con tablas de calificación graduadas) : **severidad** (gravedad del defecto), **ocurrencia** (probabilidad de que ocurra un defecto en particular) y **detección** de los de los defectos por los mecanismos que se utilizan durante las pruebas.

CAPÍTULO 1

1.- INTRODUCCIÓN

1.1 Antecedentes

Responsabilidades del Instituto Nacional de Estadística y Geografía (INEGI).

Con la aprobación de la Ley del Sistema Nacional de Información Estadística y Geográfica (LSNIEG), publicada en el Diario Oficial de la Federación del 16 de abril de 2008, se dio cumplimiento al marco legal correspondiente del Artículo 26 de la Constitución Política de los Estados Unidos Mexicanos en su apartado B, con el objetivo de regular lo concerniente al Sistema Nacional de Información Estadística y Geográfica e instrumentar la constitución del **Instituto Nacional de Estadística y Geografía (INEGI)**, en su calidad de organismo constitucional encargado de normar y coordinar el Sistema. Así el Instituto Nacional de Estadística y Geografía –conforme a lo dispuesto en el apartado B del artículo arriba mencionado de la Constitución–, es un organismo público con autonomía técnica y de gestión, personalidad jurídica y patrimonio propios, responsable de normar y coordinar el Sistema Nacional de Información Estadística y Geográfica, así como de realizar los censos nacionales; integrar el sistema de cuentas nacionales, y elaborar los índices nacionales de Precios al Consumidor y de Precios Productor. Cabe mencionar que con fecha 7 de abril de 2006, se declararon reformados los artículos 26 y 73 de la Constitución Política de los Estados Unidos Mexicanos; al primero de los artículos citados se le adicionó un inciso B en el cual se precisa que la responsabilidad de normar y coordinar el Sistema Nacional de Información Estadística y Geográfica estará a cargo de un organismo con autonomía técnica y de gestión, personalidad jurídica y patrimonios propios, con las facultades necesarias para regular la captación, procesamiento y publicación de la información que se genere y proveer a su observancia, con ello lo que se brindó una respuesta positiva a la propuesta de dotar al Instituto Nacional de Estadística, Geografía e Informática, de autonomía en el ejercicio de sus funciones y con el propósito de fortalecer su carácter institucional, lo cual quedo reglamentado en la LSNIEG.

Misión

Generar, integrar y proporcionar información estadística y geográfica de interés nacional, así como normar, coordinar y promover el desarrollo de los Sistemas Nacionales Estadístico y de Información Geográfica, con objeto de satisfacer las necesidades de información de los diversos sectores de la sociedad.

Visión 2025

México pertenece al grupo de países que basan su desarrollo en el uso de la información y en el conocimiento organizado y diseminado electrónicamente al contar con un Sistema Nacional de Información Estadística y Geográfica sustentado en una Red Nacional de Información, que facilita la toma de decisiones de todos los sectores de la sociedad con base en información oportuna y confiable.

El INEGI es responsable de coordinar el Sistema Nacional de Información Estadística y Geográfica, así como la Red Nacional de Información.

Política de calidad

Todo producto o servicio que se genere en el INEGI debe tender a la plena satisfacción de las necesidades de información estadística y geográfica de la sociedad mexicana mediante el desarrollo de su personal y la mejora continua, privilegiando la integración de metodologías y tecnologías en sus procesos y proyectos.

Política Ambiental

El INEGI, coordinador y generador de información por medios electrónicos, ópticos e impresos, comprometido con el cumplimiento de la legislación ambiental vigente y otros requisitos que la organización suscriba, así como la prevención de la contaminación; establece para sus instalaciones los objetivos de promover el consumo responsable de materiales de oficina, energéticos, agua y el adecuado manejo de los residuos, procurando una mejora continua en el desempeño ambiental de sus tareas encomendadas.

Sistemas Nacionales Estadístico y de Información Geográfica

Se definen como el conjunto orgánico de instituciones oficiales e individuos que interactúan entre sí, en un contexto de marcos jurídicos, normativos y metodológicos armonizados, que permiten ordenar y ejecutar sus actividades a través de objetivos y estrategias comunes. Los SNEIG tienen como propósito generar e integrar la información estadística y geográfica de interés nacional, bajo normas y conceptos homogéneos para todos sus participantes, a fin de proporcionar el servicio público de

información que permita, tanto a los diferentes sectores de la sociedad como a los individuos, el conocimiento, análisis y la planeación, para coadyuvar en la toma de decisiones en apoyo al desarrollo del país.

Proyectos Censales tales como:

- Censo General de Población y Vivienda
- Censo Agropecuario
- Censo Económicos

Calidad en INEGI

En 1996, el Instituto decide instalar el Programa de Calidad INEGI con la participación organizada de todos los miembros de nuestra institución, buscando con ello la mejora permanente de sus procesos, productos y servicios, la satisfacción de los requerimientos de los usuarios con el menor costo social, así como la implementación de la calidad de vida en el trabajo. Se establecieron la Misión, Visión y la Política de Calidad que habrían de regir a partir de ese momento. Dichos preceptos se actualizaron en el 2001.

Objetivo del programa de Calidad

Implementar un sistema de mejora continua que permita incrementar la participación, sensibilización y capacitación en materia de calidad del personal, aumentar los niveles de productividad y mejorar las condiciones de trabajo en todas las áreas del Instituto.

Estructura y funciones del INEGI

Para prestar el servicio público de información estadística y geográfica a una población superior a los 100 millones de habitantes, el INEGI cuenta con una organización conformada por:

- La Junta de gobierno.
- Presidencia del INEGI.
- Unidades administrativas.

La Junta de gobierno:

La instauración de la Junta de Gobierno como Órgano Colegiado nos permite avanzar en la consolidación del proceso de autonomía, asegurando que las decisiones no

serán unipersonales, sino en beneficio de una mayor solidez de la información, bajo una visión de largo plazo. A la Junta de Gobierno le corresponden, entre otras, las siguientes atribuciones:

- Orientar sus actividades hacia la planeación, programación, producción y difusión de la información de interés nacional.
- Producir el Programa Estratégico del Sistema Nacional de Información Estadística y Geográfica; el Programa Nacional de Estadística y Geografía, y el Programa Anual de Estadística y Geografía, los cuales nos marcarán las directrices a seguir en diferentes plazos.
- Aprobar la imposición de sanciones administrativas por infracciones a la Ley del Sistema Nacional de Información Estadística y Geográfica y las políticas para la administración de los recursos humanos, financieros y materiales del Instituto, en términos de las disposiciones aplicables.

Presidencia del INEGI

La sede de la Presidencia del Instituto estará establecida en la ciudad de Aguascalientes, capital del estado del mismo nombre. A la Presidencia del Instituto le corresponden, entre otras, las siguientes atribuciones:

- Tener a su cargo la administración del Instituto, la representación legal de éste, y el ejercicio de sus funciones, sin perjuicio de las atribuciones que la Ley confiere a la Junta de Gobierno.
- Ejecutar los acuerdos de la Junta de Gobierno.
- Dar a conocer a los Poderes de la Unión y al público en general el calendario de publicación de Información de Interés Nacional una vez aprobado por la Junta de Gobierno.
- Aplicar las políticas para la administración de los recursos humanos, financieros y materiales del Instituto, de acuerdo a sus programas y objetivos.

Unidades Administrativas

Dirección General de Estadísticas Sociodemográficas

Dirección General de Estadísticas Económicas

Dirección General de Geografía y Medio Ambiente

Dirección General del Servicio Público de Información

Dirección General de Coordinación del Sistema Nacional de Información Estadística y Geográfica

Dirección General de Vinculación Estratégica

Dirección General de Administración

Contraloría Interna

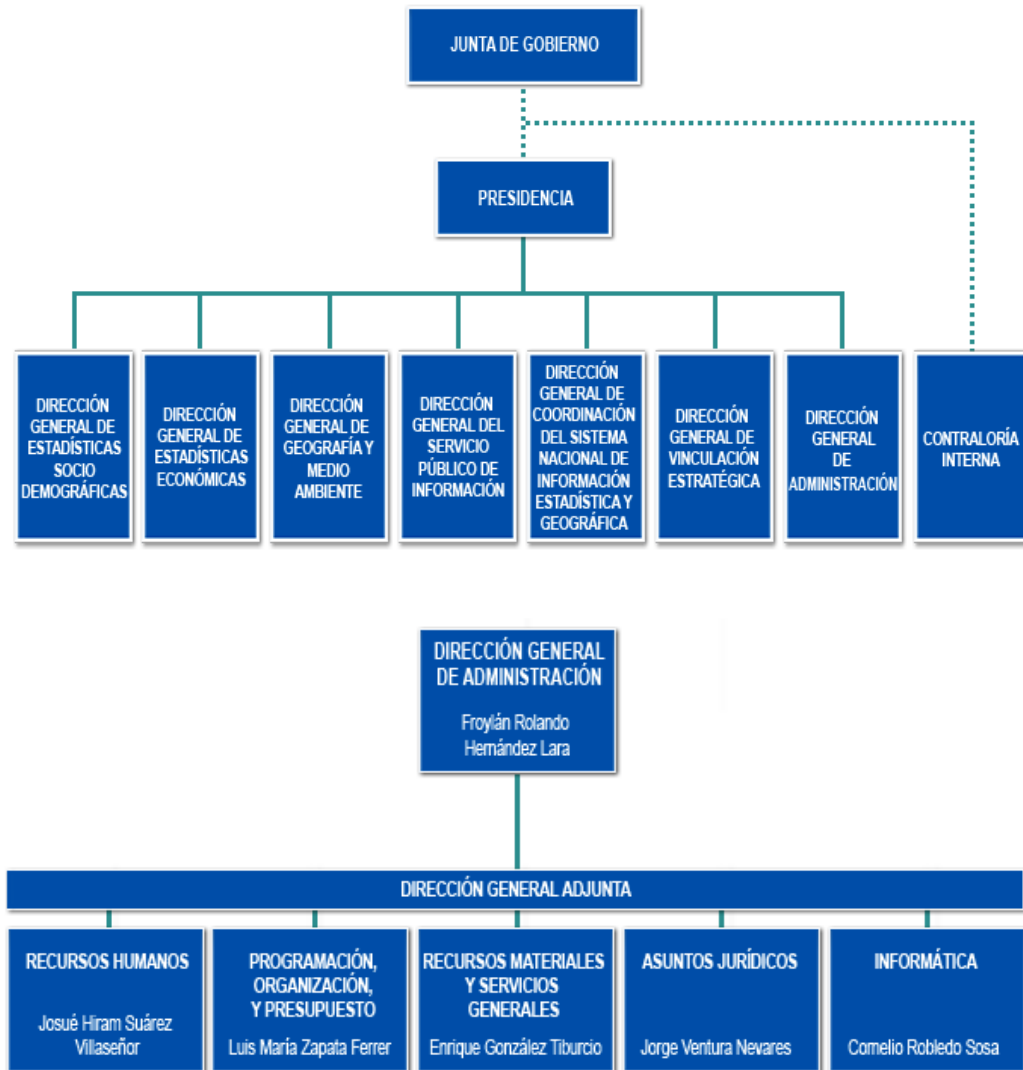


Figura 1.1 Estructura del INEGI³

³ Fuente: <http://intranet.inegi.gob.mx/default.aspx>, Agosto, 2008.

Ámbito regional y estatal

Dirección Regional. Tiene entre sus facultades y atribuciones las siguientes:

- Ejecutar, coordinar y supervisar los programas del Instituto, ajustándose a las disposiciones aplicables y a las normas, sistemas y procedimientos que al efecto establezcan las unidades administrativas centrales, y de acuerdo con los lineamientos que al efecto dicte el Presidente del propio Instituto.
- Representar al Instituto ante las autoridades estatales y municipales que correspondan a su jurisdicción, para la ejecución de los programas y concertaciones en que aquéllas participen conjuntamente con el Instituto.
- Supervisar y controlar el correcto ejercicio de las partidas que integran su presupuesto autorizado.
- Estas unidades administrativas contarán con coordinaciones estatales que ejecutarán los programas regionales, dentro de la entidad federativa correspondiente
- Para la consecución de sus objetivos en el país, el INEGI cuenta con diez direcciones regionales y 34 coordinaciones estatales.

El área de aplicación de este trabajo será la **Dirección General de Administración, Dirección General Adjunta de Informática (DGAI)** cuyas funciones generales se dan a conocer a continuación:

Facultades de la DGAI:

- Emitir los lineamientos que en materia de informática deberán observar las dependencias y entidades de la Administración Pública Federal, en su carácter de integrantes de los Servicios Nacionales de Estadística y de Información Geográfica y los Sistemas Nacionales Estadístico y de Información Geográfica.
- Promover y Coordinar tecnologías y metodologías que impulsen las competencias de los recursos humanos del Instituto; así como el Sistema de Gestión de Calidad, conforme al plan estratégico institucional y a los lineamientos que establezca el Presidente del INEGI.

1.2 Responsabilidades y necesidades del Departamento de Pruebas Modulares a Sistemas de Información (estado actual)

Actualmente en la Dirección de Desarrollo de Sistemas de Información (DDSI) de la Dirección General Adjunta de Informática (DGAJ), en el Departamento de Pruebas Modulares a Sistemas de Información se realizan actividades de Revisión y Pruebas a los Sistemas desarrollados en la misma DDSI, básicamente se trata de pruebas de **caja negra** enfocadas a la funcionalidad del producto, y aún se brinda apoyo en la **Revisión y Pruebas** de algunos Sistemas desarrollados en otras áreas de la DGAJ (como ocurrió para la etapa de Captura en Palm del VII Censo Agropecuario 2007). La problemática es que los sistemas que se reciben en el Departamento se encuentran en diferentes circunstancias: algunos sin documentación alguna, solo con una plática de uso y/o manual de funcionamiento, otros con funcionalidad y/o casos de uso, y algunos más (los pocos) con casos de prueba diseñados acompañados de un **checklist** propuesto por el programador del sistema, y el manual de usuario y/o requerimiento.

Aunado a lo anterior, se ha observado una necesidad de contar con indicadores de calidad en los avances del desarrollo, que se basen en la **detección de defectos** de estos sistemas revisados en el departamento, tanto en versiones preliminares como en la versión final del sistema bajo prueba, apoyando su correcto funcionamiento durante su producción y la calidad de la información generada por los mismos.

1.3 Evolución del proceso de pruebas a un Proceso de Pruebas estándar

En este apartado se presenta el trabajo de investigación realizada, el cual consiste en llevar el actual **proceso de pruebas ad hoc** (diseñado a la medida para cada sistema desarrollado), a un **Proceso de Pruebas** estable y bien definido.

Esta situación se puso de manifiesto el año 2007, al intentar monitorear el **Proceso de Pruebas** usando Control Estadístico de Procesos, mediante las mediciones periódicas consistentes en el conteo de los defectos encontrados por cada ciclo de prueba en el sistema para la Captura de VIII Censo Agropecuario 2007 en Palm. El Censo Agropecuario es uno de los censos más importantes y complejos que son atendidos en el Instituto Nacional de Estadística y Geografía, para la obtención de información agropecuaria del país, adicionado en esta ocasión con la innovación tecnológica de un dispositivo de captura móvil.

Este intento se realizó en la búsqueda de aportar indicadores de avance en la madurez de los sistemas bajo pruebas antes de ponerse en producción, al servicio del

usuario final. La investigación supone que el decaimiento de defectos o errores restantes del sistema, en versiones subsecuentes de pruebas, se puede aproximar mediante una distribución logarítmica y de esta manera usando Control Estadístico de Procesos indicar en qué momento el **Proceso de Pruebas** tiene características deseables como estabilidad y capacidad (detalladas en párrafos posteriores).

Es importante explicar al lector, que el resultado de este trabajo, fue madurando de manera paulatina, sorteando las dificultades que a continuación se describen:

En Noviembre del 2006, inicialmente la propuesta de trabajo de tesis fue presentada ante el Comité de la Maestría en Estadística Oficial con el título de:

“Control Estadístico de Calidad aplicado en el Proceso de Pruebas del Software”,
Basado en el artículo:

“Monitoreando el Proceso de Pruebas del Software (STP por sus siglas en inglés: Software Test Process) usando Control Estadístico del Proceso (SPC por sus siglas en inglés: Statistical Process Control): Un Acercamiento Logarítmico”, publicado en **Septiembre de 2003 por J. W. Cangussu, R. A. DeCarlo, and A. P. Mathur.[24]**

El objetivo era conocer en qué momento un proceso de desarrollo de software se encontraba bajo control ó estable (el número de defectos de las mediciones periódicas estaba dentro de los límites sugeridos por el mismo proceso y afinados por los encargados del sistema).

Además se estudiaba un segundo artículo con el título: “Generación Estadística de Casos Prueba basados en UML”, artículo Publicado en 2002 por Matthias Riebisch, ilka Philippow, Marco Götze[25].

El objetivo de esta otra investigación era automatizar la generación de casos de prueba con la documentación provista por los desarrolladores, aprovechando que los desarrollos de software en el INEGI son procesos iterativos e incrementales. La prueba automatizada es necesaria para establecer no sólo la evolución en términos de la funcionalidad, sino también en términos de la calidad del software. Esta propuesta tenía como limitante la adquisición de una herramienta de software llamada **UsageTester**, solicitada para uso académico a los autores del artículo, pero desafortunadamente no hubo respuesta a nuestra solicitud, y otro limitante fue el que se requerían casos de uso documentados en UML, como entrada a dicha herramienta y como se comentó en párrafos anteriores no todos los desarrolladores generan casos de uso en su documentación.

Retomando el primer enfoque provisto por el artículo de Cangussu: **“Control Estadístico de Calidad aplicado en el Proceso de Pruebas del Software ...”**, nos centramos en el número de errores encontrados por ciclo o unidad del tiempo y se realizaron por alrededor de 14 ciclos durante el seguimiento al desarrollo normal y ajustes al sistema en producción de 2 etapas: Enumeración de terrenos y Levantamiento de la información del Sistema de Captura en Palm del VIII Censo Agropecuario 2007.

Se propuso una variante del Control Estadístico de Procesos basado en una transformación logarítmica para permitir el monitoreo o supervisión y el control estadístico del Proceso de Pruebas de Software (STP) usando el decaimiento en el número de errores restantes en el producto bajo prueba.

Al analizar los datos el co-director de tesis M. en E. Sergio Nava refirió la siguiente problemática [ver Anexo 4]:

“Los ajustes realizados a la propuesta original de tesis de Andrés Rios se deben principalmente a la falta de un Proceso de Pruebas definido, estable, estandarizado y controlado. En la propuesta inicial (llamada originalmente: **“Control Estadístico de Calidad aplicado en el Proceso de Pruebas del Software, basada en el artículo: “Monitoreando el Proceso de Prueba del Software (STP) usando Control Estadístico del Proceso (SPC): Un Acercamiento Logarítmico”**[24], encaminada entre otras cosas a recolectar datos para aplicar técnicas estadísticas, se identificaron algoritmos de confiabilidad de software para determinar qué tan confiable era el proceso de pruebas, esto es, qué tan confiable era el proceso en la identificación de defectos en los sistemas sujetos a prueba. Las condiciones para aplicar dichos algoritmos no se cumplieron”.

A continuación se cita una parte del artículo [24] en que basó sus conclusiones el maestro Sergio Nava:

“3. 1 ¿ Presenta actualmente el STP una conducta exponencial ?

SPClog como se describe en la sección 3.3 se puede aplicar a cualquier proceso que presente una conducta exponencial. Aquí usamos STP (STP por sus siglas en inglés) para ejemplificar la aplicabilidad de esta técnica. Ya se sabe con amplia experiencia que los defectos son relativamente fáciles de encontrar durante las etapas tempranas de una fase de pruebas de un sistema. Quizás porque hay muchos de ellos y no es difícil filtrarlos [18]. Encontrar defectos se hace difícil con el paso del tiempo quizás

porque hay menos de ellos y con frecuencia requieren de combinaciones específicas de eventos para ser filtrados. Esta conducta implica una pendiente exponencial por el decaimiento en el número de defectos acumulados, y como los defectos son quitados, la intensidad de la falla y la confiabilidad también presentan una conducta exponencial [12]. Sin embargo, en algunas situaciones el STP presenta una pendiente S (shape) más que una pendiente exponencial. Esta conducta puede deberse a factores tales como el aprendizaje y la adaptación presentes al inicio de un STP [19]. En ambos casos, la suposición aquí es que el STP es un proceso bien definido, no ad hoc, Nuestro interés es con procesos donde el aprendizaje y la adaptación están minimizados y la conducta dominante puede ser caracterizada como una exponencial".

Sus conclusiones a partir de este análisis es que en ese momento no había una metodología bien implantada para hacer las pruebas, esta conclusión fue determinada al analizar los datos disponibles en ese momento (ver Anexo 5). Tal parece que las pruebas para cada sistema desarrollado eran a la medida. Considera el incumplimiento de este supuesto algo muy relevante, ya que este es un supuesto importante, y el cumplirlo sería el equivalente a tener el **Proceso de Pruebas** bajo control o estable, por lo cual, el **Proceso de Pruebas** debería estar estandarizado y bien establecido.

Estas condiciones requieren fundamentalmente de un **Proceso de Pruebas** bien definido, con límites de operación estables, uso de plantillas para reportar defectos, y sobre todo la disponibilidad a la postre de mecanismos y datos para determinar si el proceso estaba bajo control o fuera de control.

Por lo anterior, se tomó la decisión de ajustar la propuesta original para establecer las bases del **Proceso de Pruebas** y en un futuro implementar algoritmos de confiabilidad en dicho proceso.

La metodología a implementar para obtener un **Proceso de Pruebas** estandarizado fue:

- Estandarizar el **Proceso de Pruebas** mediante la generación de la Guía Metodológica del Proceso de Pruebas. Esto garantizará que independientemente de los insumos proporcionados para realizar las pruebas, se siga la misma metodología, generando resultados estándares. Los resultados serán reportados en plantillas y/o reportes que permitan evaluarse fácilmente y en trabajos futuros determinar si el proceso era estable.

- Documentar el **Proceso de Pruebas**, de manera que se pueda entrar en un ciclo de mejora del mismo y de la Guía metodológica del **Proceso de Pruebas** para la revisión de sistemas de software.
- Pilotear el **Proceso de Pruebas** con algunos de sus diferentes escenarios, considerando los escenarios que se presenten en la realidad.
- Recolectar datos para actualizar el **Proceso de Pruebas**.
- Actualizar el **Proceso de Pruebas**.

1.4 Motivación inicial de haber contactado al asesor.

La inquietud original de haber contactado al asesor es en principio aplicar alguna técnica estadística al trabajo de pruebas a sistemas de software que se realiza en el Departamento de Pruebas Modulares a los Sistemas de Información (DPMSI) de la DDSI de la DGAI del INEGI. De esta manera contar con un indicador que sirviera de referencia del avance y calidad de los sistemas probados, y que está directamente relacionada con la Ingeniería de software (en cuya área el Dr. Lemus es ampliamente reconocido). Esto condujo a que al tratar de aplicar alguna técnica estadística a los conteos de defectos de un sistema se descubrió que no era posible ya que se carecía de un **procedimiento estandarizado de pruebas y revisión de sistemas**. Por lo anterior se decidió trabajar en consolidar un **Proceso de Pruebas estable y estandarizado** e iniciar la creación de archivos de conteos de defectos o fallas que se presentan tanto en el desarrollo de sistemas de software como durante su producción. De esta manera en un futuro, como producto de la continuidad de esta investigación, se contará con indicadores de la confiabilidad y avance de los sistemas. Lo anterior en conformidad con el requerimiento inicial del cliente o usuario y de los cambios inevitables en estas dos etapas principales. Para posteriormente pensar en hacer más ágil, mediante la automatización, el proceso de pruebas mediante la Generación Estadística de Casos Prueba para los sistemas. Un primer intento de esto último fue basarse en los Casos de Uso escritos en UML, sin embargo no se contó con la herramienta requerida UsageTester y actualmente no se documentan los casos de uso mediante la herramienta UML en INEGI.

1.5 Organización del documento

La organización del presente documento es la siguiente: en el **Capítulo 2** se presenta la justificación del proyecto, explicando el por qué del trabajo realizado, el problema identificado y la estrategia o metodología del trabajo de tesis desarrollado. Comienza con la historia del Departamento de Pruebas Modulares a los Sistemas de Información

y el estado actual, incluyendo los escenarios de entrada al Proceso de Pruebas, identificados de acuerdo a los insumos proporcionados para su revisión.

En el **Capítulo 3** se describe el **Proceso de Pruebas** propuesto para la DDSI de la DGAI y sus aplicaciones en este trabajo basándose en los procedimientos descritos en la **Guía metodológica para el Proceso de Pruebas** (Anexo 2). Se comenta la importancia de la Técnica de la herramienta de mapas Conceptuales: **Cmptools** para documentar cada Proceso de Pruebas de los sistemas probados. También se explica la técnica de **AMEF** (Análisis de Modo de Fallos) y las modificaciones realizadas para acomodar su aplicación en apoyo a las pruebas de software y las limitaciones en la aplicación de la herramienta en esta tesis.

En el Anexo 1, se presenta la traducción del artículo: **“Monitoreando el Proceso de Pruebas del Software (STP por sus siglas en inglés: Software Test Process) usando Control Estadístico del Proceso (SPC por sus siglas en inglés: Statistical Process Control): Un Acercamiento Logarítmico”**, publicado en Septiembre de 2003 por **J. W. Cangussu, R. A. DeCarlo, and A. P. Mathur.[24]**, en este trabajo se explica cómo los autores proponen monitorear un **Proceso de Pruebas** de Software. Después de establecer un **Proceso de Pruebas** estandarizado este artículo se ha tomado como base para iniciar los primeros pasos para aplicar dicha técnica en el Departamento de Pruebas Modulares a Sistemas de Información (DPMSI) como se explica en el Capítulo 5.

En el **Capítulo 4** se introduce a la información de cada uno de los dos casos de estudio que se estarán tomando como base para aterrizar lo realizado en la tesis (piloteo del **Proceso de Pruebas**), así como la manera de utilizar los métodos definidos en la **Guía Metodológica del Proceso de Pruebas** (Anexo 2) llamado **narrativa del proceso de pruebas** para realizar dichas pruebas Se muestran los **Procesos de Pruebas** (Mapas Conceptuales de cada Sistema de software que se prueba), indicando que se tendrán dos grandes pruebas: Prueba del Sistema (PS) y Prueba de Aceptación (PA), Además se aplicó la técnica denominada Análisis de modo de Fallos (AMEF), se explican los antecedentes para la industria automovilística y su interés de aplicación en otras ramas como lo es el caso del **Proceso de Pruebas** de sistemas de software, finalmente su aplicación a dos proyectos a los cuales se les aplica el **Proceso de pruebas**.

En el **Capítulo 5** se muestran los resultados obtenidos en el transcurso de la investigación así como las bondades, beneficios, y limitaciones de la propuesta del **Proceso de Pruebas** que se estandarizó.

Se presentan los problemas que ocurrieron durante la investigación, las decisiones que se tomaron en discrepancia con la propuesta inicial de la tesis y cómo se lograron solucionar o adaptar. En el Anexo 1, se explica un primer intento de monitorear el **Proceso de Pruebas** mediante la aproximación logarítmica, que permite aplicar las técnicas de Control Estadístico de Procesos, basándose en el artículo correspondiente. Se inicia con la aplicación de la técnica estadística a los primeros datos de defectos recolectados en los casos de estudios que se mencionan en el Capítulo 4, dejando como trabajo futuro la investigación para la aplicación de la técnica SPClog, para determinar si el **Proceso de Pruebas** es estable y capaz en algún momento determinado.

Asimismo se explica la utilización de todos los artefactos de apoyo al proceso y su rol en el proceso, se incluyen los artefactos, actividades y roles para un **proceso unificado** y los que utilizamos para el **Proceso de Pruebas** que ocupó esta investigación.

En el apartado de **Conclusiones del Capítulo 5** se habla del impacto de haber aplicado el proceso propuesto en los casos de Estudio, del trabajo futuro que ha detonado esta aplicación y se comentan brevemente los trabajos que se pueden realizar tomando como base la investigación realizada, tales como la generación de Casos de prueba a partir de los casos de uso (documentados en notación UML).

Se comenta brevemente el perfil del personal de prueba en el Departamento de Pruebas Modulares a Sistemas de Información con fines de contratación y capacitación de personal que desee adherirse al mismo.

Se menciona la importancia de contar con un sitio de colaboración (Sharepoint), en donde se puedan dejar las diferentes versiones del sistema a probar, así como los reportes con los resultados de las pruebas correspondientes, mejorando la comunicación entre probadores, desarrolladores y usuarios.

Asimismo se comenta de la posibilidad de compensar los escenarios donde la documentación recibida de los sistemas a probar sea limitada, escasa, o ausente, mediante el análisis y aprovechamiento de la experiencia de los probadores a fin de generar los casos de prueba incluso solo con el instrumento de aplicación del evento estadístico o propiciando reuniones con los clientes o usuarios para conocer sus expectativas y necesidades.

CAPITULO 2

2.- JUSTIFICACIÓN

Este trabajo se realiza en atención a la problemática detectada al momento de realizar las pruebas a los sistemas de desarrollo de software en la DDSI de la DGAI del INEGI, se observó que las pruebas se realizaban de acuerdo a los insumos que se proporcionaban por los desarrolladores de los diferentes sistemas, pero no seguían un proceso estandarizado de revisión o pruebas, mismo que fue estandarizado en este trabajo llamándolo **Proceso de Pruebas** estandarizado.

Asimismo, en atención a las solicitudes de la parte directiva sobre información de indicadores de calidad y avances de las pruebas de los sistemas durante las distintas versiones emitidas, así como durante los ajustes de los sistemas (se pensó en obtener un indicador de confiabilidad) y antes de pasarlos a consideración de los usuarios o clientes a evaluación y ponerlos en producción.

Se comentan los antecedentes del grupo de pruebas de los sistemas de desarrollo para ubicar la problemática:

2.1 Historia del Departamento de Pruebas Modulares a los Sistemas de Información

Desde Julio de 1986 se creó un grupo de pruebas para atender actividades relacionadas con la revisión de los sistemas, desde entonces se ha participado en diversos proyectos censales y algunos de encuestas en el INEGI, realizando estas actividades, entre las que se encontraban:

- Recibir el Requerimiento del sistema a desarrollar.
- Analizar el requerimiento del sistema.
- Asistir a las reuniones con el usuario para resolver dudas conceptuales del sistema solicitado.
- Servir de enlace entre el usuario y los desarrolladores para aclaración de conceptos.
- Generar, aplicar y revisar pruebas exhaustivas a cada uno de los Módulos o componentes del sistema, para garantizar su correcto funcionamiento.

- Generar, aplicar y revisar pruebas de Integración para verificar el correcto funcionamiento de los enlaces entre los módulos que conforman el sistema, así como el correcto funcionamiento del medio ambiente de operación del sistema.
- Aplicar y revisar pruebas al sistema con archivos que contienen información Real, realizándose éstas en un ambiente de producción para garantizar el correcto funcionamiento del sistema y, al mismo tiempo, verificar el comportamiento de la información durante su paso por el sistema.
- Aplicar nuevamente las pruebas necesarias al sistema cada vez que fuera modificada alguna parte de su programación, para garantizar su correcto funcionamiento.
- Llevar un registro y control de los cambios y acuerdos, durante el desarrollo y producción del sistema, para así entregar un requerimiento actualizado al finalizar la producción.

Los resultados de estas actividades se veían reflejados en Reportes de revisión, donde se explicaba la problemática para cada defecto encontrado.

Cabe mencionar que dentro de estas actividades en primer término se contaba con la participación del grupo de pruebas en todas las reuniones con el cliente o usuario, desde el análisis y diseño del sistema hasta su liberación y en segundo término la participación en reuniones en que participaba el usuario, sirviendo de enlace entre los usuarios y los desarrolladores, y registrando y dando seguimiento a los acuerdos derivados de dichas reuniones, con el beneficio de que se podían filtrar algunos⁴ errores que seguramente aparecerán durante su construcción y producción.

2.2 Actividades actuales del Departamento de Pruebas Modulares a Sistemas de Información de la DDSI de la DGAI (INEGI)

Actualmente se reciben sistemas desarrollados para pruebas y revisión en diferentes circunstancias, a las que llamaremos escenarios, dichos sistemas se encuentran casi totalmente desarrollados o son versiones preliminares:

- Escenario 1: Sólo con manual de operación o requerimiento,
- Escenario 1A: Con manual de operación o requerimiento y servidor de pruebas actualizado.
- Escenario 2: Con funcionalidad y/o casos de uso

⁴ Administración de la Calidad del Software, Reporte técnico: TMM Un enfoque extendido, Dr. Cuauhtémoc Lemus Olalde, I. Alfonso Murillo C.363363, última fecha de modificación 30 Septiembre 2006.

- Escenario 3: Con funcionalidad y/o casos de uso y casos prueba en un checklist propuesto por el desarrollador del sistema
- Escenario 4 (deseado): Con manual de operación o requerimiento, funcionalidad y/o casos de uso, casos prueba en un checklist y servidor de pruebas actualizado.

Así mismo los resultados de dichas pruebas y revisiones se plasman en un reporte de revisión con imágenes tomadas como evidencias y/o en un checklist de casos prueba (ver Anexo 2 dentro de la Guía Metodológica).

En la Dirección de Desarrollo de Sistemas de Información el Departamento de Pruebas Modulares a Sistemas de Información trabaja directamente con los desarrolladores, se perdió comunicación con el usuario o cliente de los sistemas solicitados, esto debido tal vez a la premura que ahora acompaña a los desarrollos solicitados. El cliente o usuario se comunica directamente con el desarrollador y a su vez el jefe de proyecto de desarrollo solicita al grupo de pruebas su revisión oficial antes de “liberarlo” al usuario o cliente que lo solicitó, sin embargo en ocasiones se hacen pruebas paralelas con el usuario. Actualmente el Departamento de Pruebas Modulares a Sistemas de Información se enfoca en dar a conocer al jefe de Proyecto de desarrollo el estado de conformidad de un sistema con los requerimientos del usuario o cliente, apoyándose en un proceso de pruebas y revisión no estandarizado.

Con la implementación del **Proceso de Pruebas** estandarizado se propone realizar una revisión que independientemente de los insumos proporcionados arroje resultados los cuales son reportados en formatos o plantillas de reportes de fallas o defectos, que indiquen el grado de conformidad de los productos de software con los requerimientos de los sistemas de desarrollo. Asimismo proveer los rudimentos para realizar los conteos de defectos que lleven en un futuro a permitir el cálculo de indicadores estadísticos de confiabilidad.

En este tenor se realizaron las siguientes actividades de investigación:

Inicialmente se invirtió tiempo en tratar de Monitorear el Proceso de Pruebas mediante control estadístico de procesos para defectos restantes en el sistema de software probado en sus diferentes versiones o unidades de tiempo. Esto con la finalidad de observar en que tiempo un cierto desarrollo se encontraba bajo control y proponer mejoras, es decir si los defectos o fallas se encontraban dentro de los límites de confianza en una gráfica de control. Esta investigación se basó en el análisis de el artículo: "**Monitoring the Software Test Process Using Statistical Process Control: A Logarithmic Approach**". J. W. Cangussu, R. A. DeCarlo, and A. P. Mathur.[33]

Se llegó a la conclusión que el proceso de pruebas actual no era estable, por lo que no era susceptible a mejora, y que debería generarse un **Proceso de Pruebas** estandarizado que garantizara el mismo tratamiento de pruebas para los sistemas sometidos al Proceso de Pruebas, a partir de los insumos con que se contara al inicio de la revisión.

Al mismo tiempo se analizó un segundo artículo de investigación llamado: **“Generación Estadística de Casos Prueba basados en UML”**. Basado en el artículo del mismo nombre, publicado por Matthias Riebisch, ilka Philippow, Marco Götze, el cual consideraba la documentación de la funcionalidad de los sistemas de software a revisar en forma de casos de uso en UML.

Éste artículo considera la utilización de una herramienta de generación estadística de casos de prueba a partir de casos de uso. El impacto de dicha herramienta se vería opacado por el hecho de que el proceso donde se incrustaría dicha herramienta no ofrecería un ambiente para explotar los beneficios de automatización. Asimismo se consideró que son pocos los desarrolladores que documentan sus casos de uso, sobretodo en UML.

Además esta última opción requería de una herramienta llamada **Usagetester**, para generar los casos de prueba a partir de los casos de uso, cuyas probabilidades de transición están basados en el uso de cadenas de Markov. La herramienta se solicitó a finales del año pasado 2007 y aún no hay respuesta.

2.3 Cambio de rumbo en la investigación

Se presenta el trabajo del año pasado (2007) que justifica el cambio de dirección del trabajo a realizar en la tesis, ya que se identificó una necesidad de mayor peso que es la estandarización del proceso de pruebas:

El 17 de Noviembre del 2006 se presentó ante el comité de la Maestría en Estadística Oficial CIMAT – INEGI, durante el foro de avances de trabajos de tesis la propuesta de trabajo de Maestría llamada: “Control Estadístico de Calidad aplicado al Proceso de Pruebas de Software” cuyos objetivos principales eran:

- Mejorar el proceso actual de pruebas en la Dirección de Desarrollo de Sistemas.
- Identificar metodología de recolección de datos (conteos de errores o defectos).

- Proveer información para evaluar la calidad de los sistemas sujetos a prueba.

En ese tiempo en el área de Desarrollo de Sistemas no se contaba con una metodología que brindara información de la calidad del funcionamiento de Sistemas, ni de un registro de recolección de los datos de pruebas del Sistema que permitiera evaluaciones sustentadas y formales.

Asimismo se tenía como segunda opción la posibilidad de Generar casos de prueba mediante casos de uso, de manera automática. Dichos casos de prueba se basan en la idea de que diferentes porciones del código no necesariamente se deben de probar con el mismo rigor (principio de Pareto: regla 90-10). Se enfoca en identificar estas porciones de código que causan o contienen la mayor cantidad de defectos y ajustar los casos de prueba de manera acorde, probando de manera más rigurosa aquellas secciones de código que se ejecutan con mayor frecuencia.

Se presentó un programa de trabajo que incluía los siguientes puntos:

Plan de trabajo:

- Revisión de Literatura (Dic. 2006)
- Entender artículos seminales (Ene. de 2007)
- Adecuar artículos seminales (Feb.-Abr.2007)
- Seleccionar sistema de caso de estudio (Mar.-Jun.2007)
- Aplicar técnica al caso de estudio (May.-Jun.2007)
 - Preparar ambiente trabajo (Jun.07)
 - Obtener y analizar resultados (Jun.-Jul.2007)
 - Conclusiones (Jul.-Ago.2007)
 - Escritura de tesis(Ago.-Sep.2007)
 - Revisiones de borradores(Sep.2007)
- Tiempos
 - Fecha de Inicio: Diciembre 2006
 - Fecha tentativa de terminación: Septiembre 2007

Al participar en las pruebas del Censo agropecuario 2007 a partir de marzo del 2007, se pensó en aplicar la técnica de Control Estadístico de Calidad al Proceso de Pruebas del software. Esta idea se vio apoyada por el hecho de que el Departamento de Pruebas Modulares a Sistemas de Información fue el encargado de organizar y administrar a 34 estatales que forman parte del INEGI a lo largo de la república, por lo

cual fueron capacitados en las labores de pruebas de sistemas, mediante una metodología de pruebas de captura en el dispositivo electrónico Palm, utilizando diagramas de árbol, que contemplaban todas las posibles caminos que se podían seguir durante una entrevista a un productor capturada con una Palm. A finales de octubre del 2007 se prepararon y entregaron los datos de 14 ciclos de pruebas realizadas para el Sistema de Captura en Palm del VIII Censo agropecuario 2007 al M. en E. Sergio Nava, quien después de analizar dichos datos concluyó:

“Los ajustes realizados a la propuesta original de tesis de Andrés Rios se deben principalmente a la falta de un Proceso de Pruebas definido, estable, estandarizado y controlado. En la propuesta inicial (llamada originalmente: **“Control Estadístico de Calidad aplicado en el Proceso de Pruebas del Software, basada en el artículo: “Monitoreando el Proceso de Prueba del Software (STP) usando Control Estadístico del Proceso (SPC): Un Acercamiento Logarítmico”**”, encaminada entre otras cosas a recolectar datos para aplicar técnicas estadísticas, se identificaron algoritmos de confiabilidad de software para determinar qué tan confiable era el proceso de pruebas, esto es, qué tan confiable era el proceso en la identificación de defectos en los sistemas sujetos a prueba. Las condiciones para aplicar dichos algoritmos no se cumplieron”.

Así mismo, el 28 de Enero del 2008, durante el Foro de Avances de los trabajos de tesis de los alumnos de la Maestría en Estadística Oficial, ya se presentó un avance tomando como base ya solo el segundo enfoque de la tesis propuesta originalmente denominado: **Generación Estadística de Casos Prueba de Software**, con los mismos objetivos iniciales, pero direccionando esfuerzos hacia el apoyo al Centro de Pruebas en la actividad de Testing (pruebas). Este Centro de Pruebas tiene entre otras funciones: La realización de pruebas de caja blanca (se observa que el valor de una variable sea el óptimo), pruebas de caja negra (se introducen cambios, errores, virus a propósito y se exige que el Sistema lo detecte). En estos dos tipos de pruebas ellos trabajan con versiones beta y con versiones (como nosotros lo hacemos con versiones preliminares).

Además se realizan pruebas de volumen (como todos estos sistemas operan en Web se prueba conectividad, “stress” del Sistema, al introducir un número creciente de usuarios, simulado en una PC mediante un script la conectividad en LAN).

Desafortunadamente ésta opción de tesis también fue descartada ante la imposibilidad de obtener el software UsageTester para lograr la generación automática de casos prueba a partir de los casos de uso. Aunado al hecho de que dicha herramienta por sí

sola no sería de impacto al no contar con un proceso de pruebas definido, establecido y estandarizado.

Con todo lo anteriormente explicado la investigación cambió de dirección hacia la búsqueda de la información de apoyo que sirviera para la estandarización del **Proceso de Pruebas**, este cambio se dio en Febrero del 2008, aproximadamente, con los consabidos reconsideraciones en la investigación. Sin embargo, ya que el proceso de pruebas es una labor institucional que pretende mejorar la calidad a los sistemas desarrollados en la DDSI de la DGAI del INEGI, se tuvo la ganancia de estandarizar el proceso de Pruebas, documentar el trabajo que se ha hecho por décadas, por primera vez definiendo conteos de defectos y la posibilidad de seguir investigando en relación a la aplicación de técnicas estadísticas que apoyen y den luz acerca de la confiabilidad del proceso de pruebas.

Un **Proceso de Pruebas** estandarizado tiene muchos beneficios, como lo es el tener un proceso estable, es decir, que garantiza que no existen otras fuentes de variación implícitas en el proceso de pruebas como lo son el aprendizaje del uso del sistema y la capacitación en las etapas tempranas del **Proceso de Pruebas**. Con la anterior consideración se podrá probar si la distribución de los errores restantes tiene una distribución exponencial y solo entonces tratar de monitorear el proceso mediante una aproximación logarítmica. Metodológicamente el beneficio es que se podrá tener una Guía Metodológica del Proceso de Pruebas que permitirá guiar la realización correcta del método a seguir durante el **Proceso de Pruebas** (Mapa Conceptual de los sistemas que se pasen a revisión al Departamento de Pruebas Modulares a los Sistemas de Información) de manera sustentada.

Así mismo vale la pena mencionar que la problemática encontrada durante esta investigación en ningún momento desmotivó los objetivos primarios que son los de mejorar el **Proceso de Pruebas** mediante la estandarización de las actividades independientemente de los insumos (entradas) proporcionados para la realización de las pruebas. Pero si detonó la búsqueda de una metodología adecuada a cada uno de los sistemas para garantizar la detección confiable de defectos de los productos de software (salidas). Con esta idea se decidió además de contar los defectos encontrados en cada corte para ilustrar el decaimiento en el número de errores restantes (como inicio de los conteos de defectos), incluir un indicador denominado **npr** (número de prioridad en la atención de defectos) calculado al aplicar la técnica llamada AMEF, ampliamente utilizada en la industria automotriz para la prevención y detección de fallas o defectos de manufactura (tanto en la etapa de Diseño como de Procesos).

Esto se logró adecuando al ámbito de pruebas de software, las calificaciones correspondientes a tres tablas que califican una falla o defecto encontrado mediante las variables de **Severidad** (gravedad), **Ocurrencia** (Probabilidad de ocurrencia del defecto) y **Detección** (que califica los niveles de detección de defectos de los mecanismos de prueba), de esta manera este indicador se calcula como: **NPR = S*O*D**. Cada variable tiene un rango de 1 a 10 de calificación, lo que permite tener un rango de [1,1000], y se podrán ordenar los defectos o fallas del producto probado en la plantilla o formato correspondiente de acuerdo a la recomendación de prioridad de atención del encargado del desarrollo, siendo los números de **npr** mayores los que requieren mayor atención.

CAPITULO 3

3.- Proceso de Pruebas de Software: Propuesta para la DDSI de la DGAI(INEGI)

En este capítulo se pone de manifiesto que el **Proceso de Pruebas** del Departamento de Pruebas Modulares a Sistemas de Información de la DDSI de la DGAI deberá realizarse de acuerdo con la información con que se cuente al inicio del mismo, pero siguiendo un procedimiento estándar para obtener resultados consistentes que puedan analizarse fácilmente. Este procedimiento se encuentra definido y sustentado en la **Guía Metodológica para el Proceso de Pruebas** del DPMSI de la DDSI de la DGAI, la cual se irá mejorando y actualizando, produciendo nuevas versiones. Esta guía marcará el camino hacia la realización de la revisión y las pruebas de manera estándar.

Este trabajo plasma de manera explícita toda la experiencia de años en la realización de las pruebas a los sistemas desarrollados en ésta área y se pretende que éste **Proceso de Pruebas** sirva de base para definir y aplicar indicadores estadísticos que nos den información acerca de éste proceso de desarrollo de sistemas, basados en el número de errores o defectos encontrados en el software. A la fecha, los proyectos desarrollados en ésta Dirección se encuentran en su mayoría sujetos a revisiones antes de su puesta en operación para generar la información que se genera en el INEGI, y se requiere conocer el grado de avance y confiabilidad en cierto momento del desarrollo del proyecto.

Se pretende que este **Proceso de Pruebas** sea seguido por el personal del DPMSI, a quien llamaremos probadores o testers, cada vez que tenga en sus manos los insumos para realizar las pruebas de un sistema de desarrollo de software, terminado o una versión preliminar para revisarla, con el apoyo de esta Guía Metodológica. Como se mencionó inicialmente, este **Proceso de Pruebas** estandariza el procedimiento de revisión y pruebas independientemente de los insumos que reciba de parte de los programadores.

3.1 Guía metodológica.

A las diferentes condiciones en que se reciben los insumos para revisión y pruebas de diferentes sistemas se les denota como “escenarios” y a través de la **Guía metodológica para el Proceso de Pruebas**. Se documentó dicho **Proceso de Pruebas**,

versión 0.1, y actualmente se tiene disponible la versión 0.2, después del piloteo de dos proyectos que se mencionarán más adelante. Dichos documentos formalizan y estandarizan estos procedimientos, y los documentan técnicamente con los últimos avances de Ingeniería de software en materia de pruebas de software. En particular se tomaron como guía los avances en cuanto a Proceso de Pruebas: “Guía metodológica para la Fase de Pruebas en el Desarrollo de Software” versión 1.0 del 20 de Junio del 2005 (en su iteración_07_Pruebas), plantillas o formatos de revisión, etc. del Proyecto TRITÓN, el cual se está desarrollando en la ciudad de Zacatecas por alumnos de la Maestría en Ingeniería de Software del CIMAT.

En el Anexo 2 se presenta la **Guía Metodológica para el Proceso de Pruebas** en la Dirección de Desarrollo de Sistemas de Información de la DGAI.

El documento real de consulta para establecer la dinámica de trabajo para realizar el **Proceso de Pruebas** de cada proyecto bajo pruebas será la **Guía Metodológica para el Proceso de Pruebas**, el proceso en sí será variante y dinámico dependiendo del escenario que represente. Además la Guía Metodológica que se usará también recibirá actualizaciones y se manejará mediante versiones (reflejadas en una plantilla), que integren las innovaciones a cuestiones tecnológicas y proyectos que no se encuentren tipificados en el documento presente en su momento.

3.2 Apoyos en la documentación y evaluación del Proceso de Pruebas: Cmaptools y AMEF

Cabe mencionar que además se optó por definir y documentar el proceso de Pruebas con la herramienta denominada Cmaptools (Mapas conceptuales) para los diferentes escenarios de prueba, a fin de documentar los procesos para cada uno de los mismos. En el presente capítulo se presentan algunos detalles de la herramienta Cmap (Mapas Conceptuales) y se dan a conocer sus beneficios y los artefactos de prueba que utiliza para documentar este y todo tipo de procesos, por último se discute la inclusión de la herramienta AMEF (Análisis del Modo de Fallas) para priorizar cuantitativamente las fallas encontradas durante la fase de pruebas de los sistemas de software desarrollados, esto ya forma parte del presente trabajo de investigación, se incluye como parte de los reportes de observaciones para los desarrolladores, sugiriéndoles que atiendan prioritariamente los errores o defectos que tengan un mayor valor en la columna npr (número de prioridad) del mismo reporte (esta variable se mueve en un rango de 1 a 1000).

En el siguiente capítulo se aplicarán todos estos instrumentos de análisis y evaluación, piloteando **el Proceso de Pruebas definido en el presente trabajo**, en dos grandes proyectos: Sistema Integral de Administrador (SIA) – Módulo de Contabilidad, correspondiendo al Escenario 1 de acuerdo con la Guía Metodológica, y Sistema de Control de Producción y Ventas del INEGI (ConProVe), correspondiendo al Escenario 3.

3.2.1 Cmaptools [28]

Importancia de los Mapas conceptuales en la Gestión de Proyectos de Software

Hoy en día uno de los factores más importantes para gestionar correctamente un proyecto de software es la creación de un marco de trabajo donde los distintos actores que participan en el proyecto puedan colaborar en las distintas tareas del proyecto. Para ello se suelen utilizar, en muchos casos implícitamente, procesos de trabajo y aplicaciones con objeto de facilitar el trabajo en grupo como pueden ser: el correo electrónico, tormentas de ideas, foros, pizarrones, etc. Estas técnicas son agrupadas bajo el término Groupware. En este trabajo presentamos una herramienta Groupware que utiliza como **estrategia de colaboración: los mapas conceptuales**. Esta técnica aunque no muy conocida ha sido considerada como revolucionaria en grandes empresas en sus procesos de aprendizaje. A su vez analizaremos las implicaciones de utilizar esta estrategia de aprendizaje como herramienta para la colaboración, tanto desde el punto de vista de su influencia en el factor humano, como desde el punto de vista de las aportaciones a la gestión de proyectos de software [28].

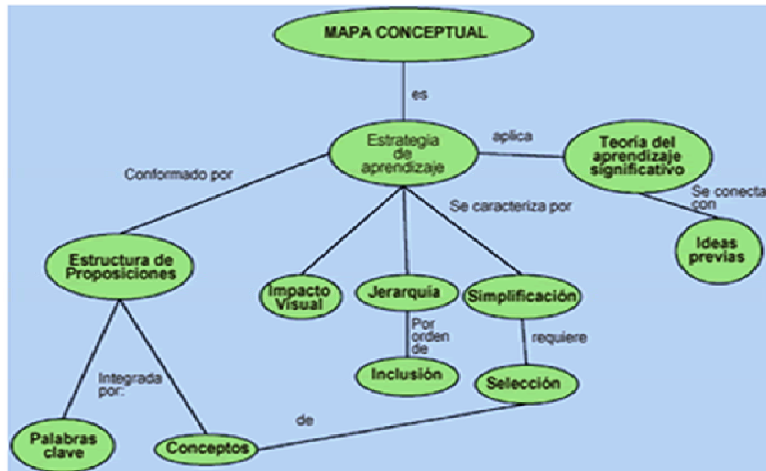


Figura 3. 1 Ejemplo de Mapas Conceptuales⁵

Características Fundamentales.

Idea central “concepto”.

- Estructura de acuerdo a su importancia.
- Utiliza términos específicos de acuerdo a la idea central.
- Interrelación de conceptos utilizando palabras o frases de enlace.

Actualmente tenemos instalada la versión 4.12 de la herramienta, misma que utilizamos en nuestro equipo de trabajo.

En éste trabajo de investigación, que consiste en la creación y documentación del Proceso de Pruebas, es importante que los probadores creen su propio esquema (mapa conceptual) de acuerdo al análisis de los insumos proporcionados para la fase de Pruebas y lo documenten compartiéndolo en un sitio sharepoint (trabajo futuro). Este sitio puede ser consultado por los demás integrantes del Departamento de Pruebas Modulares de Sistemas de Información (DPMSI) y por los programadores y servir como herramienta de colaboración en la fase de Pruebas.

⁵ Institute for Human and Machine Cognition (IHMC)

<http://www.ihmc.us>

3.2.2 AMEF

Análisis de Modos y Efectos de Fallas potenciales (AMEF) [26].

En este apartado se reseña el trabajo relativo a la investigación para cuantificar la magnitud de los defectos o errores encontrados en los sistemas de software desarrollado, en un primer intento para poder aplicar en un trabajo futuro indicadores estadísticos de confiabilidad a los desarrollos informáticos sujetos a pruebas.

Introducción

Tradicionalmente, en los procesos de comercialización de bienes y servicios, y con el objetivo de satisfacer al cliente, las empresas se han visto en la obligación de ofrecer garantías, es decir, de comprometerse con el cliente por un período determinado a reparar o sustituir de manera total o parcial los productos que presenten defectos operacionales o de construcción.

Aun cuando este compromiso representa tranquilidad para el consumidor, el hecho de no poder disponer del producto durante un período de reparación o sustitución, o que éste se averíe con mucha frecuencia; representa un motivo de insatisfacción, el cual se traduce como una pérdida de prestigio para el proveedor.

De igual manera, en aquellos casos en que el producto o servicio es utilizado en lugares remotos o en condiciones muy críticas, la garantía pasa a un segundo plano y el interés principal del cliente recae en que el producto no falle.

Por estos motivos, es deseable colocar en el mercado un producto o servicio que no presente defectos, y para tal fin en el presente trabajo se expone el Análisis de Modos y Efectos de Fallas potenciales (AMEF) como un procedimiento de gran utilidad para aumentar la confiabilidad y buscar soluciones a los problemas que puedan presentar los productos de software y los procesos de desarrollo antes de que estos ocurran (cultura de prevención de fallas) ver Anexo 6.

Actualmente, el AMEF se ha popularizado en todas las empresas automotrices americanas y ha empezado a ser utilizado en diversas áreas de una gran variedad de empresas a nivel mundial.

Requerimientos de AMEF

Para hacer un AMEF se contó con lo siguiente:

[Escribir texto]

- Un equipo de personas con el compromiso de mejorar la capacidad de diseño para satisfacer las necesidades del cliente.
- Diagramas esquemáticos y de bloque de cada nivel del sistema y el sistema completo.
- Especificaciones funcionales de módulos, sistema integral, etc. o manual de operación del sistema.
- Requisitos explícitos e implícitos de manufactura y detalles de los procesos que se van a utilizar.

Además de Formatos de Reporte de defectos encontrados incluyendo las columnas relativas al AMEF. Estos reportes ya se usaban para reportar defectos de sistemas, solo fueron modificados.

[Escribir texto]

¿Qué es AMEF?

Por lo tanto, AMEF puede ser considerado como un método analítico estandarizado para detectar y eliminar problemas de forma sistemática y total, cuyos objetivos principales son:

- Reconocer y evaluar los modos de fallas potenciales y las causas asociadas con el diseño y manufactura de un producto
- Determinar los efectos de las fallas potenciales en el desempeño del sistema
- Identificar las acciones que podrán eliminar o reducir la oportunidad de que ocurra la falla potencial
- Analizar la confiabilidad del sistema
- Documentar el proceso

Aunque el método AMEF generalmente ha sido utilizado por las industrias automotrices, éste es aplicable para la detección y bloqueo de las causas de fallas

Beneficios de AMEF

La eliminación de los modos de fallas potenciales tiene beneficios tanto a corto como a largo plazo. A corto plazo, representa ahorros de los costos de reparaciones, las pruebas repetitivas y el tiempo de paro. El beneficio a largo plazo es mucho más difícil de medir puesto que se relaciona con la satisfacción del cliente con el producto y con su percepción de la calidad; esta percepción afecta las futuras compras de los productos y/o servicios, por lo que es decisivo para crear una buena imagen de los mismos.

Formato y elementos de AMEF

Como se comentó, se utilizó el reporte de observaciones que se venía usando con anterioridad para reportar observaciones de defectos de sistemas, ver anexo de la Guía Metodológica.

Modo de falla potencial.

Se define como la manera en que una parte o ensamble puede potencialmente fallar en cumplir con los requerimientos específicos del proceso.

Se hace una lista de cada modo de falla potencial para la operación en particular; para identificar todos los posibles modos de falla, es necesario considerar que estos pueden caer dentro de una de cinco categorías:

[Escribir texto]

Falla Total
Falla Parcial
Falla Intermitente
Falla Gradual
Sobrefuncionamiento

Efectos de falla potencial.

El siguiente paso del proceso de AMEF, luego de definir la función y los modos de falla, es identificar las consecuencias potenciales del modo de falla; ésta actividad debe de realizarse a través de la tormenta de ideas y una vez identificadas estas consecuencias, deben introducirse en el modelo como efectos.

Severidad.

El primer paso para el análisis de riesgos es cuantificar la severidad de los efectos, éstos son evaluados en una escala del 1 al 10 donde 10 es lo más severo.

A continuación les presentare las tablas con los criterios de evaluación para diseño y para el proceso:

Cualquiera característica con una severidad de 9 o 10 que requiera un control especial asegurar la detección es una característica crítica.

Los ejemplos del producto o de los requisitos del proceso que podrían ser características críticas incluyen dimensiones, especificaciones, pruebas, secuencias de ensamblaje, los útiles, los empalmes, los esfuerzos de torsión, las autógenas, las conexiones, y los usos componentes. Las acciones o los controles especiales necesarios para resolver estos requisitos pueden implicar la fabricación, ensamblaje, un surtidor, envío, el vigilar, o examen.

Las características significativas requieren controles especiales porque son importantes para la satisfacción de cliente. Los grados de la severidad entre 5 y 8 se juntaron con una ocurrencia que clasificaba mayor de 3 indican características significativas. En un diseño AMEF, son potenciales Características Significativas. En el proceso AMEF, si un control especial se requiere para asegurar la detección entonces una característica significativa real existe. Las compañías no han estandarizado un método para agrupar y denotar características especiales del producto. La nomenclatura y la notación variarán.

[Escribir texto]

Ocurrencia.

Las causas son evaluadas en términos de ocurrencia, ésta se define como la probabilidad de que una causa en particular ocurra y resulte en un modo de falla durante la vida esperada del producto, es decir, representa la remota probabilidad de que el cliente experimente el efecto del modo de falla.

EL valor de la ocurrencia se determina a través de las siguientes tablas, en caso de obtener valores intermedios se asume el superior inmediato, y si se desconociera totalmente la probabilidad de falla se debe asumir una ocurrencia igual a 10.

Controles actuales.

Los controles actuales son descripciones de las medidas que previenen que ocurra el modo de falla o detectan el modo de falla en caso de que ocurran. Los controles de diseño y proceso se agrupan de acuerdo a su propósito:

Tipo 1: Estos controles previenen la causa o el modo de falla de que ocurran, o reduce su ocurrencia

Tipo 2: Estos controles detectan la causa del modo de falla y guían hacia una acción correctiva

Tipo 3: Estos controles detectan el modo de falla antes de que el producto llegue al cliente

Detección.

La detección es una evaluación de las probabilidades de que los controles del proceso propuestos (listados en la columna anterior) detecten el modo de falla, antes de que la parte o componente salga de la localidad de manufactura o ensamble.

No es probable que verificaciones de control de calidad al azar detecten la existencia de un defecto aislado y por tanto no resultarán en un cambio notable del grado de detección. Un control de detección válido es el muestreo hecho con bases estadísticas.

NPR

El número de prioridad de riesgo (NPR) es el producto matemático de la severidad, la ocurrencia y la detección, es decir:

$$NPR = S * O * D$$

Este valor se emplea para identificar los riesgos más serios para buscar acciones correctivas.

[Escribir texto]

Acción (es) recomendada (s).

Cuando los modos de falla han sido ordenados por el NPR, las acciones correctivas deberán dirigirse primero a los problemas y puntos de mayor grado e elementos críticos. La intención de cualquier acción recomendada es reducir los grados de ocurrencia, severidad y/o detección. Si no se recomienda ninguna acción para una causa específica, se debe indicar así.

Un AMEF de proceso tendrá un valor limitado si no cuenta con acciones correctivas y efectivas. Es la responsabilidad de todas las actividades afectadas el implementar programas de seguimiento efectivos para atender todas las recomendaciones.

Área/individuo responsable y fecha de terminación (de la acción recomendada)

Se registra el área y la persona responsable de la acción recomendada, así como la fecha meta de terminación.

Acciones tomadas.

Después de que se haya completado una acción, registre una breve descripción de la acción actual y fecha efectiva o de terminación.

Npr resultante.

Después de haber identificado la acción correctiva, se estima y registra los grados de ocurrencia, severidad y detección finales. Se calcula el NPR resultante, éste es el producto de los valores de severidad, ocurrencia y detección.

El ingeniero en proceso es responsable de asegurar que todas las acciones recomendadas sean implementadas y monitoreadas adecuadamente. AMEF es un documento viviente y deberá reflejar siempre el último nivel de diseño.

Conclusión

Mediante la realización del presente informe, se establece la gran importancia y el alcance de los beneficios que proporciona el Análisis de Modo y Efectos de Falla Potencial como una herramienta para examinar todas las formas en que un producto o proceso pueda fallar; además se hace una revisión de la acción que debe tomar para minimizar la probabilidad de falla o el efecto de la misma.

Dado que para la mayoría de los productos y procesos no es económico llevar a cabo AMEF para cada componente, se hace necesaria la realización de los elementos críticos que deben ser sometidos al mismo.

Aunque AMEF es muy valioso como una técnica de advertencia temprana, la prueba definitiva viene dado por el uso del producto por parte del cliente.

[Escribir texto]

Sin embargo la experiencia de campo llega demasiado tarde, y es aquí donde resalta la importancia de que ésta sea precedida por AMEF para que las empresas puedan simular el uso de sus productos y procesos en el campo de trabajo.

3.2.2.1 Limitaciones en la aplicación de la herramienta en esta tesis

De las reuniones sostenidas con:

- Los probadores del Departamento de Pruebas Modulares para Sistemas de Información.
- El M. en E. Sergio Nava.
- La reunión entre probadores del Departamento.

Se obtuvo lo siguiente:

- Se tomarán solo las ponderaciones de los tipos de errores para un **proceso**, ya que actualmente no participamos en el **diseño** de los sistemas, recibiendo los sistemas solo en la Fase de Pruebas.
- Uno de los mayores obstáculos a vencer será el hecho de que los defectos encontrados en una versión del sistema que se esté probando (versión) no se conoce en qué momento serán corregidos por el autor con una nueva versión, por lo cual persisten en el reporte correspondiente por varias revisiones.
- Seguiremos aplicando el formato para revisión: FORMATOPRUEB.doc comentado en la **Guía Metodológica para Procesos de Pruebas**, solo incluiremos cuatro columnas más llamadas: Severidad(S),, en la cual incluiremos la calificación de la severidad (gravedad) del defecto encontrado, esta se decidirá con la técnica de “Lluvia de ideas” con la participación de los probadores involucrados y con el consentimiento del autor o encargado del sistema, Ocurrencia(O), en la cual pondremos la probabilidad de que ocurra la

[Escribir texto]

falla del sistema y por último Detección(D), en la cual calificaremos el nivel de eficiencia de los controles que actualmente usamos para detectar una falla o defecto y por último el producto de ellas: **npr**.

Como se comentó arriba solo aplicaremos las ponderaciones o calificaciones para los tipos de errores de un proceso, en particular solo para la tabla 2, la cual modificamos con situaciones propias del **Proceso de Pruebas** de sistemas que se sigue en el INEGI (en la DDSI de la DGAI). A continuación incluimos esta tabla a la cual le llamamos **Tabla 3.7 Criterios de evaluación y sistema de graduación sugeridos para la severidad de los efectos en un proceso AMEF (Caso INEGI)**, las tablas correspondientes a **Ocurrencia y Detección** seguirán aplicándose **sin cambio**, por el momento.

[Escribir texto]

Severidad en el proceso AMEF para el proceso de Pruebas en INEGI.

Tabla3. 1 Criterios de la evaluación y sistema de graduación sugeridos para la severidad de efectos en un proceso AMEF (Caso INEGI).

Efecto	Criterios: Severidad del efecto para DDSI, DGAI del INEGI	Tipo	Calif.
Peligroso; sin alarma	Afecta la conformidad de la funcionalidad (no hace lo que debe hacer) y al ocurrir la falla NO emitirá mensaje alguno (P.ej.; No graba los datos que debe grabar en la Base de Datos y no manda mensaje).	Funcional	10
Peligroso; con alarma	Afecta la conformidad de la funcionalidad (no hace lo que debe hacer) y al ocurrir la falla SI emitirá mensaje.	Funcional	9
Muy Arriba	La observación interrumpirá la ejecución del sistema o echará a perder la información producida. El sistema se vuelve inoperable en caso de que se inhiba (ya no se puede seguir probando).	Funcional	8
Alto	La observación interrumpirá la ejecución del sistema pero en menor grado. El sistema es operable (se puede seguir probando) pero en un nivel más reducido de su funcionalidad (.P.ej.: No presenta datos capturados con anterioridad).	Funcional	7
Moderado	La interrupción será de menor importancia. El sistema es operable (se puede seguir probando), pero cierto(s) funcionamiento(s) del sistema relacionado(s) con la comodidad y la conveniencia NO opera(n)(P.ej.: no genera una gráfica o pantalla, no habilita un checkbox que debería habilitar o no muestra las opciones, íconos o ligas no funcionan ,no navega correctamente a través del sistema).	Funcional	6
Bajo	La interrupción será de menor importancia. El sistema es operable (se puede seguir probando), pero cierto(s) funcionamiento(s) del sistema relacionado(s) con la comodidad y la conveniencia opera(n) a un nivel reducido) (P.ej.: genera una gráfica o pantalla incompleta, habilita un checkbox pero no muestra las opciones, no valida algunos campos, entra en ciclos repetitivos, se desactiva al pasar de una opción a otra del menú).	Funcional	5
Muy Bajo	La interrupción será de apreciación. El sistema es operable (se puede seguir probando y hasta liberarlo a producción), aunque la mayoría de los usuarios notarán el defecto.(P.ej.: en algunas pantallas habrá faltas de ortografía, palabras incompletas, colores inadecuados, no muestra barras de avance).	Funcional	4
De menor importancia	La interrupción será de apreciación. El sistema es operable (se puede seguir probando y hasta liberarlo a producción), aunque los usuarios medios notarán el defecto.P.ej.: en algunas pantallas habrá tipos de letra y tamaños no estandarizados con todas las demás, mostrará porciones de código al ejecutarse).	No Funcional	3

[Escribir texto]

Muy de menor importancia	La interrupción será de menor importancia. El sistema es operable (se puede seguir probando y hasta liberarlo a producción), aunque los usuarios exigentes notarán el defecto. P.ej.: en algunas pantallas habrá mala redacción de los ítem's, y faltas de corrección de estilo, en resultados de consultas habrá: falta de subtítulos completos y correctos).	No Funcional	2
Ninguno	La observación NO tiene ningún efecto en la producción de información del sistema. (P.e.: Falta de atención en el servidor de pruebas o producción, se dilata al lanzar un proceso).	No funcional	1

[Escribir texto]

Tabla 3. 2 Criterios de la evaluación y sistema de graduación sugeridos para la ocurrencia del incidente en un proceso AMEF

Probabilidad del incidente	Incidente Tarifas	Pk de C	Fila
Muy Arriba: El incidente es casi inevitable	1 en 2 \geq	< 0.33	10
	1 en 3	0.33 IV	9
Alto: Asociado generalmente a los procesos similares que han fallado anteriormente	1 en 8	0.51 IV	8
	1 en 20	0.67 IV	7
Moderado: Asociado generalmente a los procesos similares previos que han experimentado incidentes ocasionales, pero no en proporción es importantes	1 en 80	0.83 IV	6
	1 en 400	1.00 IV	5
	1 de 2,000	1.17 IV	4
Bajo: Los incidentes aislados se asocian a procesos similares	1 en 15,000	1.33 IV	3
Muy Bajo: Solamente los incidentes aislados se asocian a procesos casi idénticos	1 en 150,000	1.50 IV	2
Telecontrol: El incidente es inverosímil.	1 en 1,500,000 IV	1.67 IV	1

[Escribir texto]

Tabla 3. 3 Criterios de la evaluación y sistema de graduación sugeridos para la detección de una causa del incidente o del modo de fallo en un proceso AMEF.

Detección	Probabilidad de la detección por el control del proceso.	Fila
Casi imposible	Ninguno de los controles disponibles detecta el incidente de Modo o causa.	10
Muy Alejado	Los controles actuales tienen una probabilidad muy alejada de detectar modo o causa de fallo.	9
Alejado	Los controles actuales tienen una probabilidad alejada de detectar modo o causa de fallo.	8
Muy Bajo	Los controles actuales tienen una probabilidad muy baja de detectar modo o causa de fallo.	7
Bajo	Los controles actuales tienen una probabilidad baja de detectar Modo o causa de fallo.	6
Moderado	Los controles actuales tienen una probabilidad moderada de detectar modo o causa de fallo.	5
Moderadamente Alto	Los controles actuales tienen una probabilidad moderadamente alta de detectar modo o causa de fallo.	4
Alto	Los controles actuales tienen una alta probabilidad de detectar modo o causa de fallo.	3
Muy Alto	Los controles actuales tienen una probabilidad muy alta de detectar modo o causa de fallo.	2
Casi Seguro	Los controles actuales detectan casi seguros el modo o la causa de fallo. Los controles confiables de la detección se saben con procesos similares.	1

[Escribir texto]

4.-CASO DE ESTUDIO.

Se ha aplicado el **Proceso de Pruebas** estandarizado a los proyectos que se someten a pruebas en el Departamento de Pruebas Modulares a los Sistemas de Información. Se han tomado como piloteo dos de ellos, a fin de retroalimentar y actualizar la **Guía Metodológica del Proceso de Pruebas** (Anexo 2) para la Revisión de Sistemas de Software, actualmente se cuenta con la versión 0.2. Esta Guía Metodológica indica los procedimientos que deben seguirse una vez que se ha definido un **Proceso de Pruebas** (Mapa Conceptual de la Prueba del Sistema (PS) y Prueba de Aceptación (PA) para cada proyecto que es sometido a pruebas). Los proyectos que han sido tomados como piloteo son: Sistema Integral de Administrador del INEGI (SIA) - Módulo de Contabilidad, correspondiendo al Escenario 1 que de acuerdo con la **Guía Metodológica del Proceso de Pruebas** se aplica cuando solo se cuenta con un manual de usuario del sistema o requerimiento, y Sistema de Control de Producción y Ventas de INEGI (ConProVe), correspondiendo al Escenario 3 en donde de acuerdo al mismo documento se cuenta con funcionalidad y/o casos de uso y casos prueba acompañados con checklist de casos prueba propuesto por el desarrollador del sistema.

[Escribir texto]

4.1 Información del proyecto Sistema Integral de Administración (SIA) - Módulo de Contabilidad.

Sistema Integral de Administración (SIA), es el sistema informático integral del INEGI que conforma una base de datos única como resultado de las actividades relacionadas con el manejo de recursos humanos, recursos materiales y recursos financieros, a partir del cual se dispone de información para la toma de decisiones a fin de lograr una mayor eficiencia en el manejo de los recursos del Instituto.

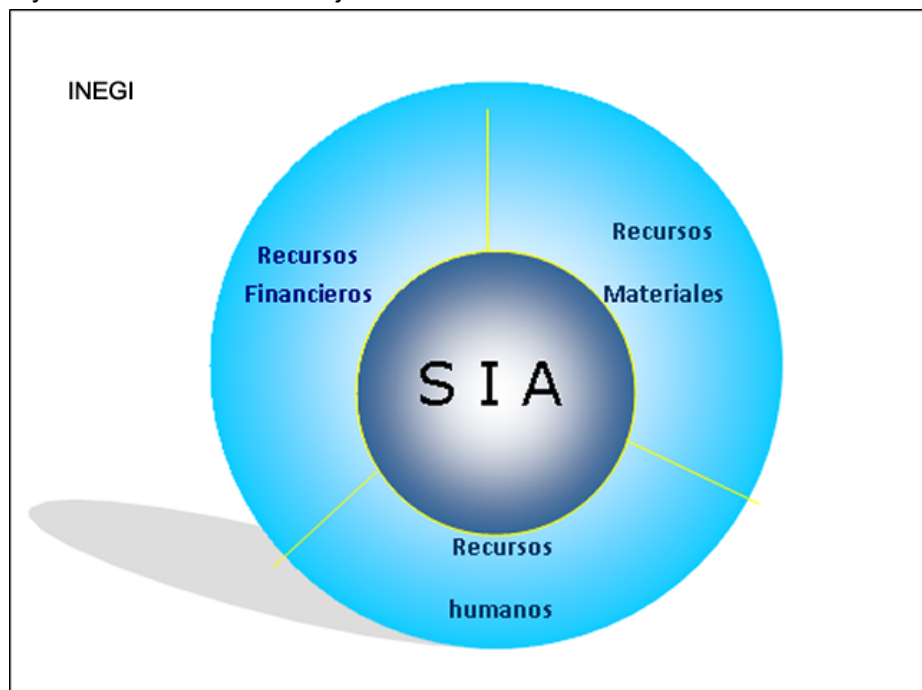


Figura 4. 1 Sistema Integral de Administración del INEGI

4.1.1 El proceso aplicado al SIA - Módulo de Contabilidad.

En este apartado se explica el procedimiento aplicado al proyecto SIA como piloteo del funcionamiento del sistema y a continuación se muestra el resultado de la documentación del **Proceso de Pruebas**, mediante el uso de la herramienta Cmaptools, con dos figuras referentes a la Prueba de Sistema (PS) y a la Prueba de Aceptación del sistema (PA). En primer lugar se incluye la **narrativa del Proceso de**

[Escribir texto]

Pruebas de acuerdo a lo referido en la **Guía Metodológica del Proceso de Pruebas** (Anexo 2) para la Revisión de Sistemas de Software.

- Escenario 1: Sólo con manual de operación o requerimiento: *Caso Sistema Integral de Administración (SIA): Módulo Contabilidad.*

Narrativa del Proceso de Pruebas:

Se formó un **equipo de inspección o revisión** (de 3 a 6 personas dependiendo de disponibilidad y competencia con otros proyectos) y se realizaron las Pruebas del Sistema (funcionalidad) (PS) de las diferentes versiones o preliminares del SIA, posteriormente la Prueba de Aceptación (PA).

Se optó por aplicar el Método de Revisión en Equipo (menos riguroso que el Método de Inspección ó Revisión, consultar Anexo 2 **Guía Metodológica del Proceso de Pruebas**). Las revisiones en equipo son un tipo de inspección más ligera, que aunque están planeados y estructurados, son menos formales y menos comprendidos. El equipo de revisión tiene varios nombres como el de “walkthrough estructurado” o simplemente “revisión”. Estas revisiones siguen varios de los pasos encontrados en una inspección. Los pasos fueron:

- 1.- Los participantes reciben el material a revisar, con días de anticipación a la reunión para que lo puedan estudiar y analizar por ellos mismos.
- 2.- El equipo reúne datos sobre el esfuerzo dedicado a la revisión y el número de defectos encontrados (se usará el formato de revisión del manual o requerimiento, de la sección A2.11 del Anexo 2 **Guía Metodológica para el Proceso de Pruebas**). Sin embargo, se simplifican o se omiten la descripción y las etapas de inspección, y algunos roles de los participantes pueden ser combinados.
- 3.- El autor puede conducir al equipo de revisión, de no ser así, el **Administrador de Pruebas** lo hará, mientras que en una inspección el autor no puede funcionar como asesor. A diferencia de la mayoría de las inspecciones, se omite el papel del lector.
- 4.- En lugar de tener un lector que vaya describiendo por partes el producto, el líder de la revisión (**Administrador de Pruebas**), sólo

pregunta a los participantes (**testers**) si tienen comentarios sobre una sección o página específica del trabajo.

5.- Es más costoso tener un equipo de revisión que tener a un colega realizando la revisión con ayuda de un checklist detallado, la diferencia es que los participantes del equipo (**testers**) encontrarán otros o más errores.

6.- Como en cualquier reunión, pueden presentarse discusiones u opiniones diferentes o bien, acuerdos que puedan ser olvidados más adelante, por lo tanto la persona que dirige la revisión (**Administrador de Pruebas**), deberá registrar todo lo que se trató durante la junta. La persona que registra o escribe, captura todo lo que sucede durante la reunión pero usando formatos estándar que la organización ha definido o adoptado (en este caso no están definidos, pero se anotan en un cuaderno exclusivo para el proyecto).

7.- Al final de la reunión, el equipo lleva a cabo una valoración del producto de trabajo al autor o Líder de Proyecto y decide cómo verificar los cambios que el autor realizará durante la reanudación del trabajo.

8.- Se utilizarán los reportes de revisión de la sección A2.11 llamados **Formatoprueb.doc** e **imágenes.doc** del Anexo 2 **Guía Metodológica para el Proceso de Pruebas**, los reportes de cada iteración (versión) de las inspecciones con sus respectivas pruebas se van depurando quitando/agregando los defectos corregidos/agregados para las nuevas versiones de Sistema o Producto de Software, Se cuenta el número de defectos encontrados en el reporte de revisión correspondiente en cada versión del sistema o aplicación puesto a revisión, obteniéndose el promedio de defectos encontrados en el sistema desarrollado por ciclo o periodo de prueba y se colocan en una tabla, mismos que se grafican mediante una Gráfica de Control X-barras, para observar si el proceso es estable y tenía la capacidad de cumplir los requerimientos en el tiempo esperado por el encargado del sistema, ésta grafica la realiza el **Administrador de Pruebas** (mediante una función de R llamada **qcc** véase sección A2.12 Programas en R), la interpreta y entrega este material al encargado del sistema acompañado de los reportes de observaciones de la sección A2.11, (actualmente solo se entregan los reportes correspondientes). Asimismo, en estos reportes se encuentran

[Escribir texto]

las calificaciones de las cualidades de: Severidad, Ocurrencia y Detección de cada una de las observaciones o defectos encontrados, de acuerdo a las tablas del AMEF incluidas en el Capítulo 3 de este trabajo, obteniéndose el número de prioridad de atención: npr, sugerida para las observaciones o errores.

9.- El ciclo continúa hasta liberar el Producto de Trabajo, el ciclo se cierra al entregar un reporte compendio con todas las observaciones atendidas y corregidas por fecha.

Los resultados de este proceso fueron documentados con la herramienta Cmaptools como se muestra en las siguientes figuras:

[Escribir texto]

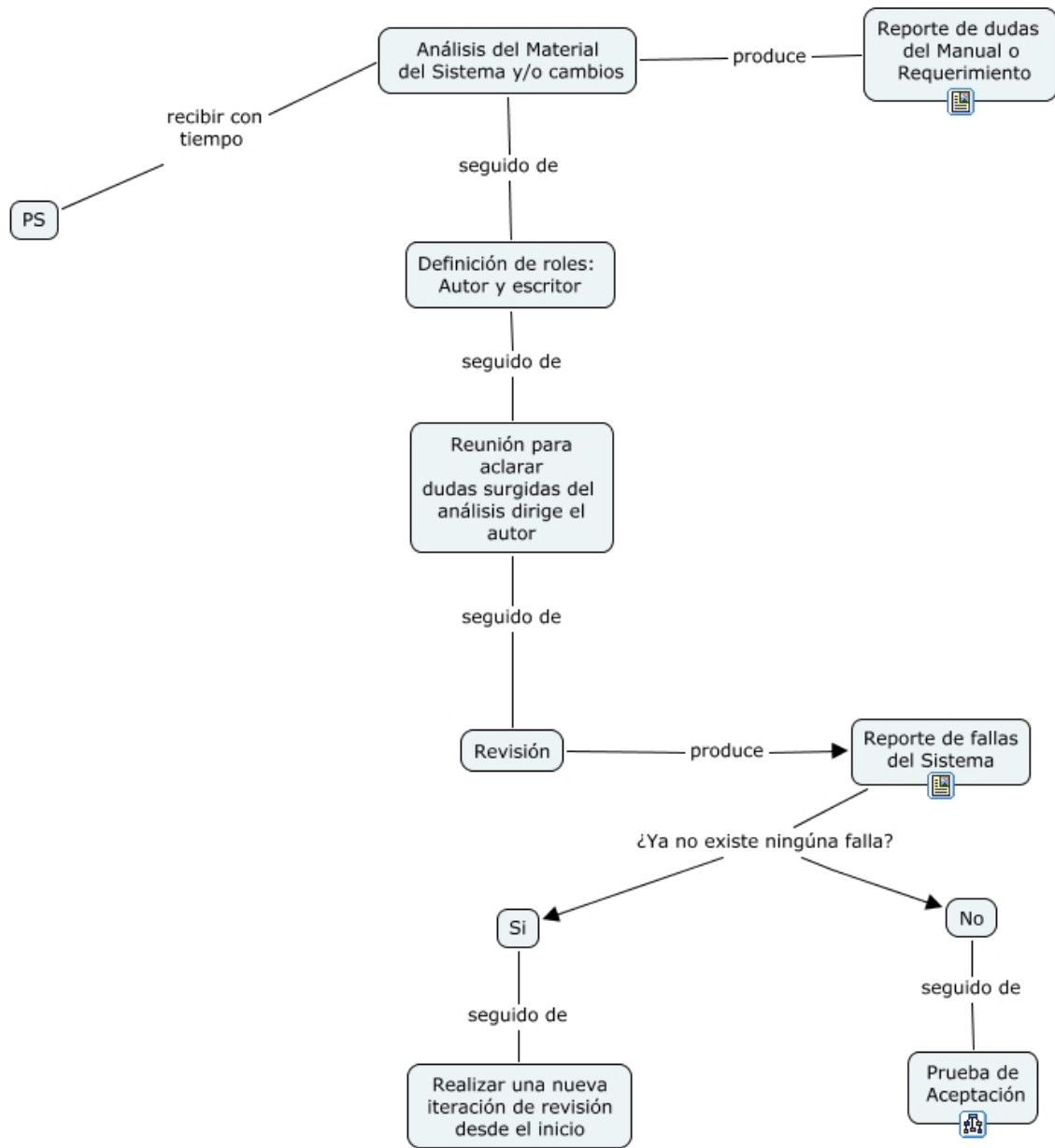


Figura 4. 2 Prueba del Sistema (PS) del Proyecto SIA (Módulo de Contabilidad) producto de la aplicación del Proceso de Pruebas estandarizado basado en la Guía Metodológica del Proceso de Pruebas para la revisión de Sistemas de software y documentado con la herramienta Cnaptools.

[Escribir texto]

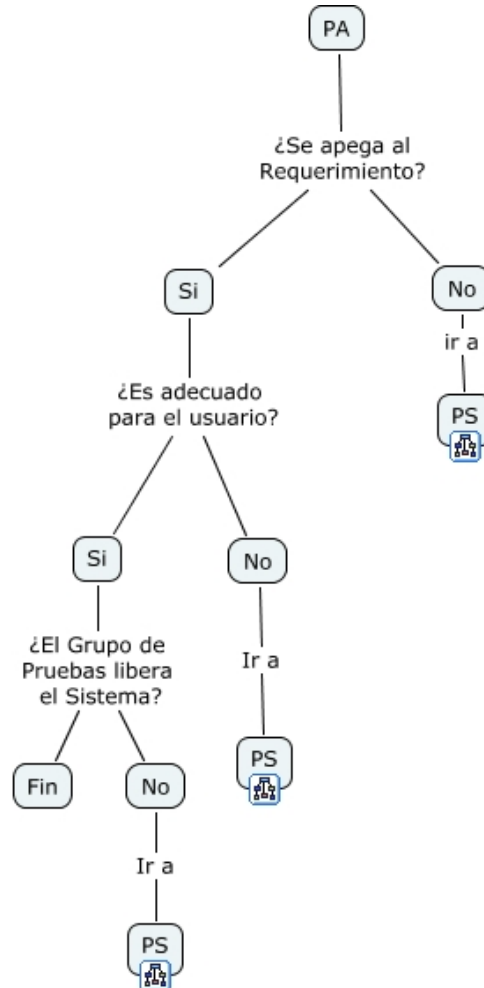


Figura 4. 3 Prueba de Aceptación sistema del Proyecto SIA (Módulo de Contabilidad) producto de la aplicación del Proceso de Pruebas estandarizado basado en la Guía Metodológica del Proceso de Pruebas para la revisión de Sistemas de software y documentado con la herramienta Cmaptools.

4.2 Información del proyecto Control de Producción y Ventas del INEGI (ConProVe).

ConProVe

Dentro del proceso de modernización y siempre en búsqueda de la vanguardia en tecnologías, el INEGI pone en marcha en 2002 el proyecto ConProVe como parte integral del SIA, cuyo objetivo es el Control de la Producción y Ventas del INEGI, que proporciona las herramientas para la administración de la planeación, producción, distribución y comercialización de los productos del Instituto, en una plataforma web con arquitectura de tres capas y una base de datos centralizada. El sistema consta de 14 subsistemas, cuyo desarrollo se ha priorizado de acuerdo a las necesidades de los usuarios del mismo, de la secuencia y de la dependencia de los procesos que

[Escribir texto]

intervienen desde la planeación hasta la venta de los productos. Se han liberado hasta la fecha 10 subsistemas, los cuales ya se encuentran en producción; se tienen 3 más en pruebas para su liberación y restan dos más para su desarrollo: comercialización y ventas, fijación de precios y registros contables.

El Sistema ConProVe en su parte liberada y en producción, se alimenta de la información proporcionada por las áreas involucradas en los procesos y asegura la disponibilidad, oportunidad e integridad de la información en tiempo real. Integrado el subsistema de comercialización y ventas, proporcionará y compartirá a su vez esta información sustentando la toma de decisiones.

OBJETIVOS

General

Proporcionar en el mes de octubre de 2007, una aplicación integrada al Sistema de Control de la Producción y Ventas (ConProVe), que permita el control de los procesos de ventas de productos y servicios que presta el INEGI para brindar un mejor servicio a la sociedad.

Específicos

Recabar los requerimientos del área de comercialización y administración.

Analizar, diseñar, construir, probar e implementar, así como proporcionar la capacitación necesaria para la operación de las opciones del Subsistema de Ventas.

Opciones del Subsistema de ventas

Catálogos: clientes, servicios, productos

Reportes

Cotización

Facturación

Nota de crédito

i

[Escribir texto]

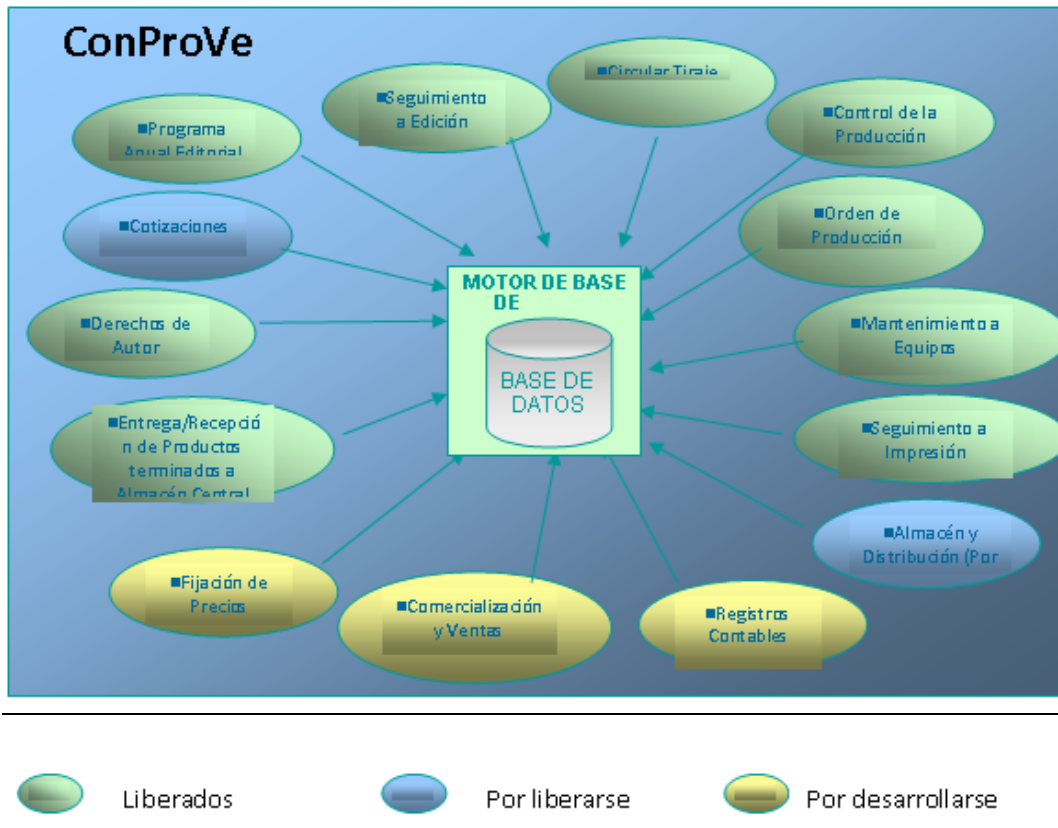


Figura 4. 4 Módulos de ConProVe

El módulo que nos compete es el de comercialización y Ventas, submódulo Facturación.

4.2.1 El proceso aplicado al ConProVe (Control de Producción y Ventas del INEGI).

En este apartado se explica el procedimiento aplicado al proyecto ConProVe como piloteo del **Proceso de Pruebas** y a continuación se muestra el resultado de la documentación del proceso mediante dos figuras referentes a la prueba de funcionalidad del sistema (PS) y a la prueba de aceptación del sistema (PA), al igual que en proyecto anterior.

- Escenario 3: Con funcionalidad y/o casos de uso y casos prueba acompañados con checklist propuesto por el desarrollador del sistema. Caso ConProVe

Se forma un **equipo de inspección o revisión** (de 3 a 6 personas dependiendo de disponibilidad y competencia con otros proyectos) y se realizan pruebas de

integración ya que el sistema se estaba desarrollando aún, una vez que los desarrolladores terminaron sus pruebas unitarias. Se prueba funcionalidad Pruebas del Sistema (PS) y la Prueba de Aceptación (PA). Dado que presentan además casos de uso, se podrán generar algunos casos prueba que no estén incluidos en la batería de casos de pruebas entregados en el checklist correspondiente.

Desgraciadamente no se contó con un servidor para realizar las pruebas exclusivamente del departamento, se probó en el mismo servidor junto con las pruebas de los desarrolladores y de los usuarios.

Se utilizó el Método de Revisión Formal más riguroso (de acuerdo con la Guía **Metodológica del Proceso de Pruebas**, véase Anexo 2): La Inspección se realizó de la siguiente manera:

- 1.- se asignaron roles específicos a cada integrante del equipo de trabajo.
- 2.- El proceso de inspección más común incluyó siete etapas que son: planeación, visión general, preparación, reunión, reanudación de trabajo, dar seguimiento y análisis.
- 3.- Contamos con checklist de defectos comúnmente encontrados en diversos tipos de trabajo de software, reglas para construir estos productos.
- 4.- Los participantes a excepción del autor del producto, condujeron la reunión (puede ser el **Administrador de Pruebas** o uno de los desarrolladores), presentan el material al equipo de inspección (demás **testers**) y registran los puntos que se hayan mencionado en la reunión (el registro lo puede hacer el **Administrador de Pruebas**).
- 5.- Los participantes se prepararon para la reunión de la inspección, examinando el material por ellos mismos para entenderlo y para encontrarle detalles que pudieran estar dando problemas. El equipo reúne datos sobre el esfuerzo dedicado a la revisión y el número de defectos encontrados (se usará el formato de revisión del manual o requerimiento, de la sección A2.11 del Anexo 2 **Guía Metodológica para el Proceso de Pruebas**. Durante la reunión, el lector (**Administrador de Pruebas** o uno de los **testers**) presenta el material a los demás

inspectores (**testers**), quienes hacen más comentarios e identifican posibles defectos. El lector ayudo al equipo a darle la misma interpretación a cada parte del producto para que los inspectores comparen lo que ellos han comprendido, con lo que el lector había expresado.

6.- Al final de la reunión, el equipo llevó a cabo una valoración del producto de trabajo al autor o Líder de Proyecto.

7.- Y se decidió cómo verificar los cambios (en el servidor de prueba) que el autor iba realizando durante la reanudación del trabajo.

8.- Los reportes de cada iteración de las inspecciones con sus respectivas pruebas se iban depurando quitando/agregando los defectos corregidos/agregados para las nuevas versiones del Sistema o Producto de Software. Se cuenta el número de defectos encontrados en el reporte de revisión correspondiente en cada versión del sistema o aplicación puesto a revisión, obteniéndose el promedio de defectos encontrados en el sistema desarrollado por ciclo o periodo de prueba y se colocan en una tabla, mismos que se grafican mediante una Gráfica de Control X-barras, para observar si el proceso es estable y tenía la capacidad de cumplir los requerimientos en el tiempo esperado por el encargado del sistema, , ésta grafica la realiza el **Administrador de Pruebas** (mediante una función de R llamada **qcc** véase sección A2.12 Programas en R), la interpreta y entrega este material al encargado del sistema acompañado de los reportes de observaciones de la sección A2.11, (actualmente solo se entregan los reportes correspondientes). Asimismo se calificaron las cualidades de: Severidad, Ocurrencia y Detección de cada una de las observaciones o defectos encontrados, de acuerdo a las tablas del AMEF incluidas en el Capítulo 3 de este trabajo, obteniéndose el número de prioridad de atención: npr, sugerida para las observaciones o errores.

9.- El ciclo continuó hasta liberar el Producto de Trabajo, , el ciclo se cierra al entregar un reporte compendio con todas las observaciones atendidas y corregidas por fecha.

[Escribir texto]

Los resultados de este proceso fueron documentados con la herramienta Cmaptools como se muestra en las siguientes figuras:

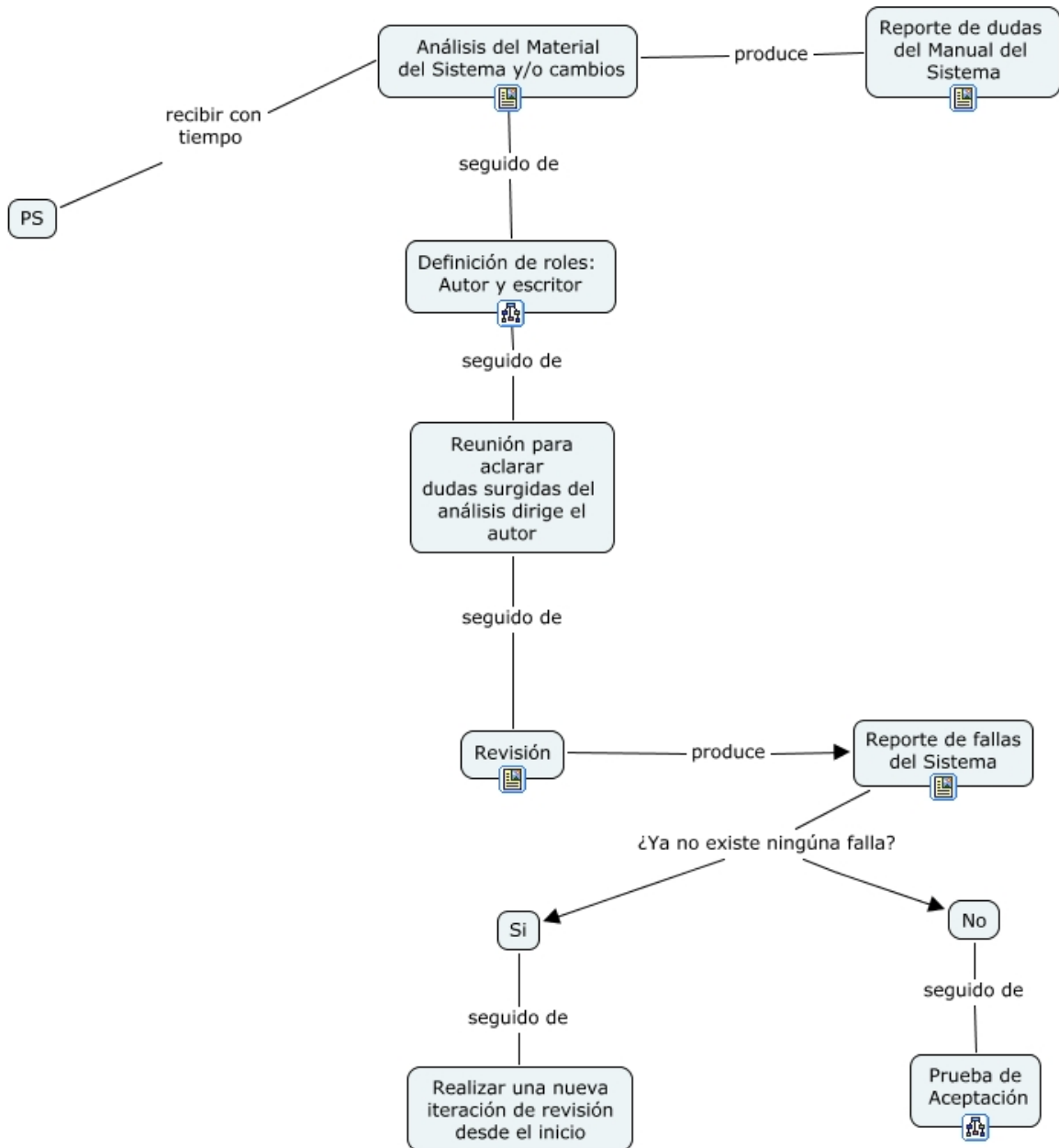


Figura 4. 5 Prueba del Sistema (PS) del Proyecto ConProVe producto de la aplicación del Proceso de Pruebas estandarizado basado en la Guía Metodológica del Proceso de Pruebas para la revisión de Sistemas de software y documentado con la herramienta Cmaptools.

[Escribir texto]

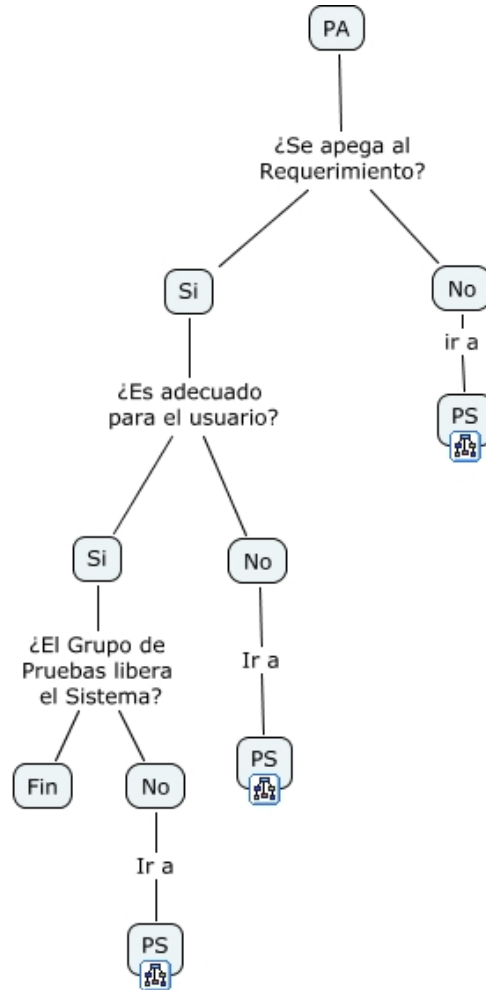


Figura 4. 6 Prueba de Aceptación sistema del Proyecto ConProVe producto de la aplicación del Proceso de Pruebas estandarizado basado en la Guía Metodológica del Proceso de Pruebas para la revisión de Sistemas de software y documentado con la herramienta Cmaptools.

Cabe mencionar que con la aplicación del **Proceso de Pruebas** estandarizado se actualizó el procedimiento reflejado en la **Guía Metodológica del Proceso de Pruebas** para la revisión de Sistemas de software.

CAPITULO 5

5.- RESULTADOS, CONCLUSIONES y TRABAJO FUTURO

5.1 Resultados

Los resultados principales del presente trabajo de investigación son:

- Se logró establecer un **Proceso de Pruebas** estandarizado para la Dirección de Desarrollo de Sistemas de Información (DDSI) de la DGAI, cuyos sistemas y productos de software son revisados y probados por el Departamento de Pruebas Modulares a Sistemas de Información (DPMSI). Dicho proceso es estable y estándar consistió en seguir un procedimiento estable y estándar en el cual independientemente de los insumos proporcionados para las pruebas de los sistemas se obtengan resultados predecibles que permitan conocer el estado del Proceso de Pruebas en un momento determinado. Esto se logró documentando esquemáticamente el Proceso de Pruebas mediante un Mapa Conceptual (usando la herramienta Cmaptools), apoyando las actividades específicas con la **Guía Metodológica para el Proceso de Pruebas**, la cual finca las bases metodológicas que respaldan las actividades del **Proceso de Pruebas**, en el marco de lo más actual en términos de Ingeniería de software. Asimismo se piloteó dicho **Proceso de Pruebas** con dos casos de estudio (SIA – Contabilidad y ConProVe) mediante la generación de sus correspondientes Mapas Conceptuales, en los cuales se distinguen las relaciones entre las diferentes actividades, roles y artefactos de apoyo para efectuar las Pruebas del Sistema (PS) y las Pruebas de Aceptación (PA) de esos sistemas
- Se logró alcanzar la versión 0.2 de la **Guía Metodológica para el Proceso de Pruebas**, ya que la versión 0.1 creada inicialmente fue actualizada después de pilotear el proceso de pruebas con los casos de estudio mencionados en el resultado anterior. Dicha guía es referenciada por el **Proceso de Pruebas**, que cada probador utiliza para cada sistema o producto de software que le sea presentado para probarlo. El probador utiliza dicho proceso con los procedimientos de pruebas más actualizados para preparar el ambiente de pruebas adecuado y escoger el método de pruebas y revisión más adecuado para efectuar su labor. Todo lo anterior con la finalidad que mediante las pruebas de software se liberen a producción sistemas y productos de software con una menor cantidad de defectos, con la consiguiente satisfacción de los clientes o usuarios.

- Este esfuerzo es pionero en el Departamento en el intento de evaluar numéricamente los resultados del proceso de pruebas (idea inicial aún antes de darnos cuenta de que se debía estandarizar el **Proceso de Pruebas**). Ya que para evaluar cualquier fenómeno es necesario tener datos de su desempeño en el tiempo a fin de poder entender su comportamiento. Normalmente, en nuestro ámbito se realizaban pruebas cuyos resultados solo se plasmaba en un papel o se comunicaban verbalmente a los encargados del sistema, programadores o usuarios, pero en ningún momento se pensaba en registrar datos de los defectos o errores encontrados durante las pruebas con el fin de evaluar de alguna manera el desempeño de dicho proceso de pruebas y menos se consideraba en conocer la calidad de los resultados esperados de dichos sistemas, Esto es, en esos momentos la prioridad y preocupación era la de comunicar la problemática encontrada para su atención, terminar los desarrollos (incluyendo las pruebas) en el tiempo esperado o solicitado por el usuario, “liberar” el sistema para entregarlo al cliente o usuario y ponerlo en producción, etc. Esta primera versión de propuesta de recolección de datos tiene como finalidad que **en un futuro** se puedan aplicar técnicas estadísticas para obtener indicadores de calidad. Además se incluye la técnica de prevención en modo de fallos (AMEF) la cual se adaptó en algunos puntos para evaluar los defectos del software encontrados durante las pruebas, como resultado se priorizan los defectos o fallas encontradas para su atención por parte de los encargados del sistema. Todo establece un historial del desempeño de los principales sistemas en el tiempo, estableciendo antecedentes para pruebas de futuros proyectos.

Otros logros.

- Otro logro importante de este trabajo consiste en reutilizar el conocimiento y experiencia que se tiene el Departamento de Pruebas Modulares para Sistemas de Información, ya que desde hace ya mucho tiempo y proyectos atrás, se han emitido reportes en diferentes formatos, resguardándolos en medios magnéticos, pero al revisarlos tratando de encontrar un patrón de comportamiento que ayude a abordar las pruebas de proyectos similares que se presentan actualmente no es tan fácil aprovechar las lecciones aprendidas, reutilizando los procesos y pruebas aplicadas, en parte debido a la escasa utilidad de esta documentación o a que los procesos de pruebas eran ambiguos y hechos a la medida, por lo que no se pudieron aprovechar y/o

adaptar al sistema bajo prueba. Esto se convirtió en un ciclo vicioso, ocasionando que esta actividad de pruebas no se beneficiara de un conocimiento claro del comportamiento real del fenómeno de Pruebas a Sistemas de software. Al encontrar esta forma de documentar y obtener resultados estándares de pruebas con conteos de defectos el **Proceso de Pruebas** se ve enriquecido.

Reseña de la decisión del cambio de rumbo de la investigación inicial.

- A pesar de haberse presentado múltiples obstáculos durante esta investigación, como se ha reseñado desde el Capítulo 1 de este trabajo, entre ellos el hecho de querer aplicar técnicas estadísticas al Proceso de Pruebas de software cuando no se contaba con un proceso de Pruebas estable y estándar (que impedía el entendimiento de este fenómeno por medio del análisis de la distribución de los errores a través del tiempo) se decidió replantear a abordar el Proceso de Pruebas como temática fundamental de investigación.
- Al observar el número promedio de errores por ciclo de prueba de un proyecto, inicialmente no presentaba un patrón específico de comportamiento que permitiera suponer que la distribución de los errores restantes durante algunos periodos o ciclos de pruebas (agrupados) pudiera aproximarse por medio de una distribución logarítmica [1] y aplicar indicadores estadísticos que nos ayudaran, en determinado momento, a entender si el proceso era estable (estaba dentro de los límites permisibles de errores) y en un futuro proponer una medida estadística de confiabilidad que garantizara su uso por el cliente o usuario que lo solicitó y durante su producción. Ante este inconveniente, por asesoría y recomendación director y co – director de tesis, se replanteó la tesis orientándola hacia su estado actual, se enfocaron los esfuerzos en estandarizar las actividades de pruebas, mediante un **Proceso de Pruebas** estándar en el cual independientemente de los insumos proporcionados para su revisión por los encargados del sistema, el cual al aplicarlo se obtuvieran resultados confiables y que posteriormente mediante conteos de errores se observara si reflejaban un patrón de comportamiento en el cual el número de errores restante (que fuera quedando sin resolver en las diferentes versiones de desarrollo entregadas a pruebas por los programadores) reflejara una comportamiento decreciente, sin la variación inicial incluida por el aprendizaje de los probadores (testers).

Incursión en la Técnica y primeros resultados.

A continuación se presentan los primeros resultados obtenidos al aplicar el nuevo Proceso de Pruebas, estandarizado y enriquecido mediante la aplicación de las innovaciones tecnológicas en materia de Ingeniería de software aportadas por el Dr. Lemus, director de la tesis, a dos proyectos que han sido atendidos en el Departamento para ejemplificar el **trabajo futuro** en la aplicación de algunas de estas técnicas:

5.2 Caso ConProVe

Como recordaremos el caso del sistema de Control de Producción y Ventas(ConProVe), módulo de Ventas – Facturación, fue incluido como ejemplo del escenario 3 de pruebas, en el cual se cuenta con funcionalidad y/o casos de uso, casos de prueba acompañados con checklist, es conveniente aclarar que este sistema se clasificó como escenario 3, porque no se entregó el requerimiento o el manual de funcionamiento, debido a que los requisitos fueron obtenidos por el encargado del proyecto en reuniones con los usuarios y plasmados directamente en la funcionalidad del sistema, asimismo no se contó con un servidor de pruebas actualizado y exclusivo para el DPMSI.

Los datos obtenidos de los conteos de errores o defectos por ciclo durante las pruebas de este proyecto se encuentran en la siguiente tabla:

Tabla 5. 1 Información de defectos del ConProVe

Ciclo	Número de ciclo	Defectos por ciclo
22/01/2008	1	10
29/01/2008	2	6
07/01/2008	3	6
14/02/2008	4	10
18/02/2008	5	3
28/02/2008	6	0

Para conocer los datos correspondientes a los reportes de observaciones (fallas o defectos) del sistema en cuestión acudir al **ANEXO 4 Datos estadísticos**.

[Escribir texto]

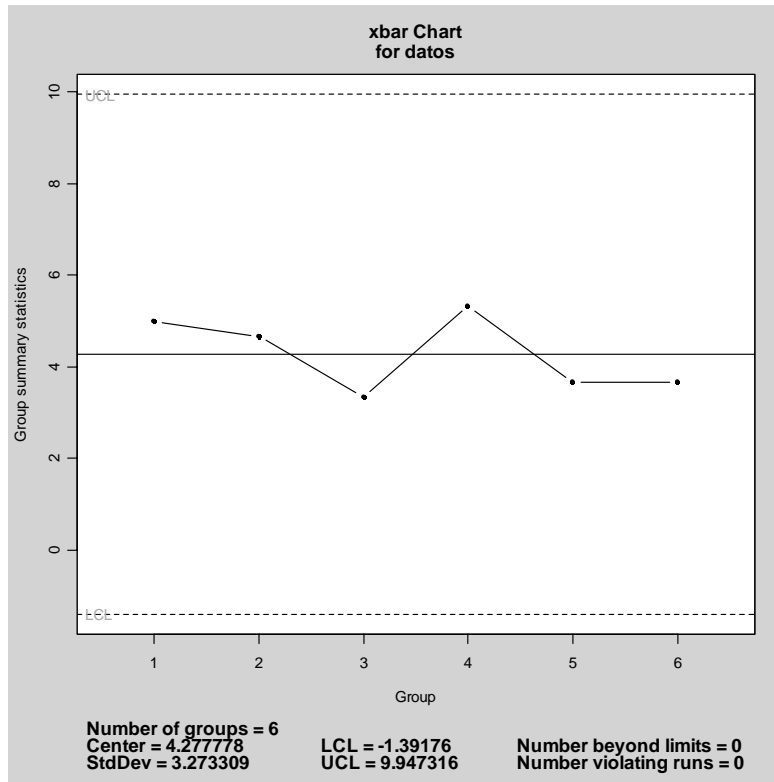
Volviendo al **Capítulo 1.- Introducción, sección 1.3 Evolución del proceso de pruebas a un Proceso de Pruebas estándar**, como ahí se menciona: en Noviembre del 2006, la propuesta de trabajo de tesis fue presentada ante el Comité de la Maestría en Estadística Oficial Tesis con el título de : **“Control Estadístico de Calidad aplicado en el Proceso de Pruebas del Software”**, basado en el artículo: **“Monitoreando el Proceso de Pruebas del Software (STP por sus siglas en inglés: Software Test Process) usando Control Estadístico del Proceso (SPC por sus siglas en inglés: Statistical Process Control): Un Acercamiento Logarítmico”**, publicado en Septiembre de 2003 por **J. W. Cangussu, R. A. DeCarlo, and A. P. Mathur.[24]**.El objetivo era conocer en qué momento un proceso de desarrollo de software se encontraba bajo control ó estable (el número de defectos de las mediciones periódicas estaba dentro de los límites sugeridos por el mismo proceso y afinados por los encargados del sistema).

Con la finalidad de conocer la aplicabilidad de la técnica planteada originalmente: Control estadístico de Procesos aplicado al Proceso de Pruebas y apoyados en la observación de los datos de errores o defectos restantes (agrupados por ciclo de prueba) durante la fase de pruebas de los sistemas tomados como casos de estudio, nos anima a dar los primeros pasos en la aplicación de las técnicas propuestas en el **ANEXO 3 Monitoreo del Proceso de Pruebas**. Esto se debe a la tendencia natural de los defectos a ser muchos al inicio de la fase de pruebas e ir disminuyendo con el paso del tiempo, ya que para encontrar los errores restantes en ese momento el probador tiene que realizar combinaciones complejas de funciones, siempre guiado por el plan de pruebas que incluye los casos prueba planeados en conjunto con el desarrollador y los no planeados que surgen durante el proceso de pruebas y revisión y que están directamente con la experiencia y habilidad del probador. Mediante la utilización de la función **qcc** se graficó la Carta de Control X barra para el número de errores restantes para el módulo del ConProVe llamado Ventas – Facturación:

De acuerdo al artículo que motivó el tema original de tesis denominado: **“Monitoreando el Proceso de Prueba del Software (STP) usando Control Estadístico del Proceso (SPC): Un Acercamiento Logarítmico”**, el análisis es el siguiente:

[Escribir texto]

Con la ayuda de la función `qcc` de R se obtuvo la gráfica X-barra:



Interpretación: Los puntos promedio correspondientes a las 6 ciclos se encuentran dentro de los límites de control (LCL y UCL), por lo cual de acuerdo al **Anexo 1 Monitoreo del Proceso de Pruebas**, y aplicando el Control Estadístico de Procesos (SPC) independientemente de la restricción de perturbaciones correlacionadas que inducen a la transformación logarítmica referida en el artículo que ahí se detalla el **Proceso de Pruebas** del caso ConProVe se considera estable:

Estabilidad [15]: Un proceso es estable si varía dentro de límites predichos. Si la variación está más allá de estos límites, el proceso se considera inestable (y no bajo control).

Aún si el proceso se saliera de los límites de control especificados (LCL y UCL, se debe considerar los siguiente regla:

El análisis de una carta de la X-barra determina estabilidad de proceso. Un proceso se dice para estar fuera del control (inestable) si cualquiera de las pruebas siguientes ocurre [15]:

Test 1: Un punto está fuera de los límites de LCL y de UCL.

Test 2: Al menos dos de tres valores sucesivos caen en el mismo lado y más de 2σ unidades alejados de la línea central.

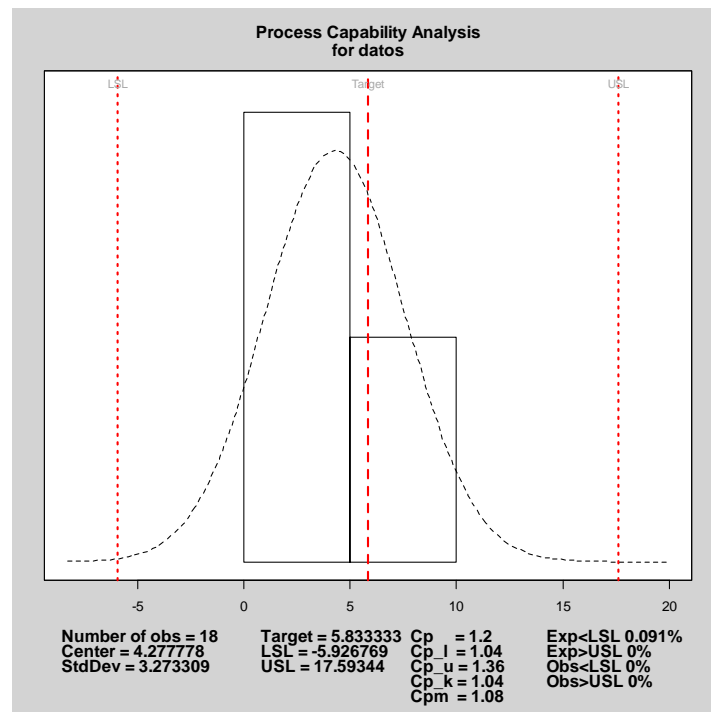
[Escribir texto]

Test 3: Al menos cuatro de cinco valores sucesivos caen en el mismo lado de, y más allá de 1σ unidad lejos de, la línea central.

Test 4: Al menos ocho puntos sucesivos caen del mismo lado de la línea central.

Observando la anterior, según las pruebas definidas arriba el proceso bajo observación está bajo control, es decir, el **proceso es estable**.

A continuación se obtuvo un Histograma de capacidad para ConProVe utilizando la misma herramienta, esta herramienta se utiliza una vez que se ha observado que un proceso es estable y se desea conocer si es capaz de terminar dentro del tiempo esperado por el encargado del proyecto:



Interpretacion: De acuerdo al **Anexo 1 Monitoreo del Proceso de Pruebas**. Se aplicó un Histograma de Capacidad:

Capacidad [15]: Un proceso es capaz si puede alcanzar los resultados previstos dentro de un tiempo especificado. La capacidad se basa en la probabilidad de que el proceso puede alcanzar los resultados dentro del tiempo especificado. Esto presupone claramente que el proceso es estable.

[Escribir texto]

Un histograma de capacidades tal como el de la figura anterior se puede usar para completar esta tarea determinando si un proceso está fuera de los límites esperados. El estar fuera de los límites esperados representa una alta probabilidad de generar resultados no conformables e indica que el proceso debería ajustarse para lograr esos resultados.

Como puede observarse en la figura anterior el proceso es Capaz, dado que el histograma se encuentra dentro de los límites de especificación mostrados (LSL y USL)

Conclusión: El proceso de Pruebas del ConProVe se observó que es un Proceso estable al observarse que los errores observados nunca salieron de los límites de control en la Carta X-barra y es Capaz de obtener los resultados en el tiempo esperado por el encargado del Sistema dado que siempre se mantuvo dentro de los límites de especificación del histograma de Capacidad correspondiente.

5.3 Caso SIA - Contabilidad:

El caso del módulo de contabilidad del SIA, fue incluido como ejemplo del escenario 1 de pruebas, en el cual se cuenta con requerimiento o manual de funcionamiento del sistema únicamente como insumos proporcionados para realizar las pruebas.

Los datos obtenidos de los conteos de errores o defectos encontrados por ciclo durante las pruebas de este proyecto se encuentran en la siguiente tabla:

Tabla 5. 2 Información de defectos del SIA-
Contabilidad

Ciclo	Defectos por ciclo	Media de defectos por ciclo
1	26	13
2	46	11.5
3	20	6.66666667
4	27	5.4
5	35	7
6	38	8,5
7	42	10.5
8	41	10.25
9	34	8.5

[Escribir texto]

10	11	5.5
11	17	5.66666667
12	24	6
13	29	3.33333333
14	4	4
15	6	3
16	2	2
17	4	2
18	2	2
19	3	3
20	2	2

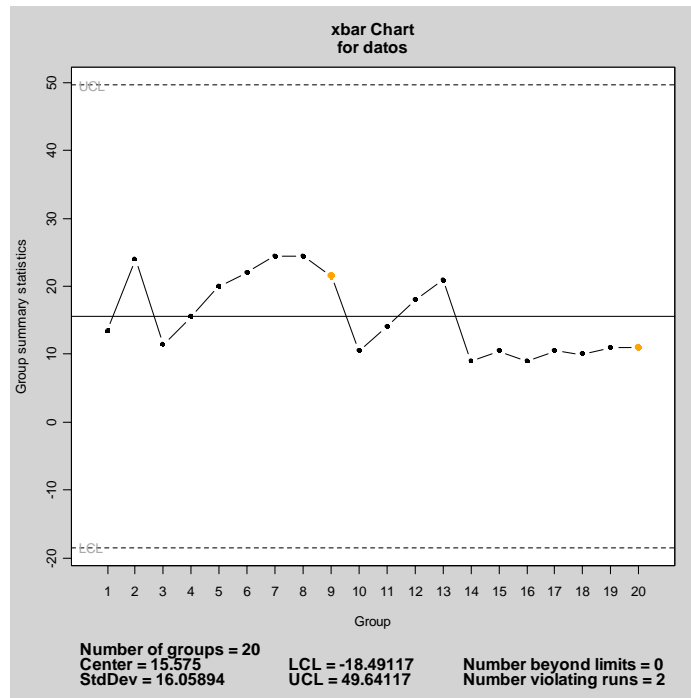
Para conocer los datos correspondientes a los reportes de observaciones (fallas o defectos) del sistema en cuestión acudir al **ANEXO 3 Datos estadísticos**.

Como en el caso del análisis correspondiente al sistema ConProVe, también para el sistema SIA-Contabilidad se desea conocer la aplicabilidad de la técnica originalmente planteada en el presente trabajo de tesis, el cual se denominó: "Control estadístico de Procesos aplicado al Proceso de Pruebas" y apoyados en la observación de los datos de errores o defectos restantes (agrupados por ciclo de prueba) durante la fase de pruebas de los productos de software, nos anima a dar los primeros pasos en la aplicación de las técnicas propuestas en el **ANEXO 1 Monitoreo del Proceso de Pruebas**. Mediante la utilización de la misma función **qcc** se graficó la Carta de Control X barra para el número de errores restantes para éste sistema, con los siguientes resultados.

De acuerdo al artículo que motivó el tema original de tesis denominado: **"Monitoreando el Proceso de Prueba del Software (STP) usando Control Estadístico del Proceso (SPC): Un Acercamiento Logarítmico"**, el análisis es el siguiente:

[Escribir texto]

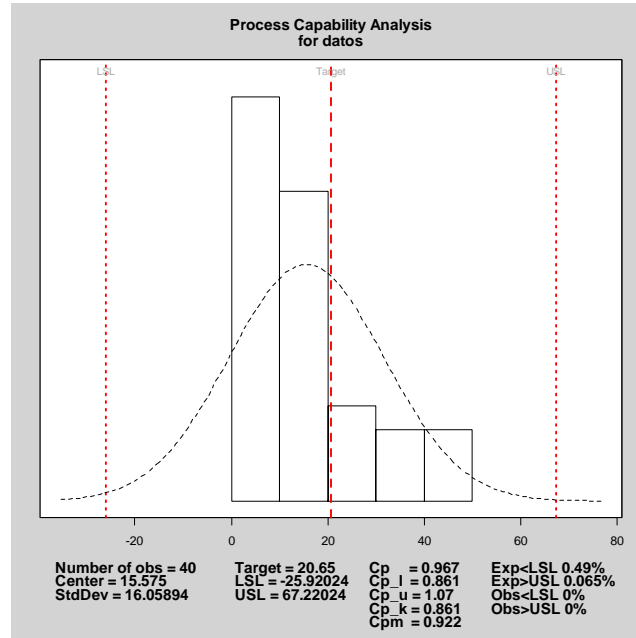
Con la ayuda también de la función **qcc** de R se obtuvo la gráfica X-barra:



Interpretación: Los puntos promedio correspondientes a los 20 ciclos se encuentran dentro de los límites de control (LCL y UCL), por lo cual de acuerdo al **Anexo 1 Monitoreo del Proceso de Pruebas**, y aplicando el Control Estadístico de Procesos (SPC) independientemente de la restricción de perturbaciones correlacionadas que inducen a la transformación logarítmica referida en el artículo que ahí se detalla el **Proceso de Pruebas** del caso SIA – Módulo Contabilidad se considera estable ya que el número promedio de errores varía dentro de límites predichos. Si la variación está más allá de estos límites, el proceso se considera inestable (y no bajo control).

A continuación se obtuvo un Histograma de capacidad para SIA –Módulo Contabilidad utilizando la misma herramienta, esta herramienta se utiliza una vez que se ha observado que un proceso es estable y se desea conocer si es capaz de terminar dentro del tiempo esperado por el encargado del proyecto:

[Escribir texto]



Interpretación: De acuerdo al **Anexo 1 Monitoreo del Proceso de Pruebas**. Se aplicó un Histograma de Capacidad:

El estar fuera de los límites esperados representa una alta probabilidad de generar resultados no conformables e indica que el proceso debería ajustarse para lograr esos resultados.

Como puede observarse en la figura anterior el proceso es Capaz, dado que el histograma se encuentra dentro de los límites de especificación mostrados (LSL y USL)

Conclusión: El proceso de Pruebas del SIA - Contabilidad se observó que es un Proceso estable al observarse que los errores observados nunca salieron de los límites de control en la Carta X-barra y es Capaz de obtener los resultados en el tiempo esperado por el encargado del Sistema dado que siempre se mantuvo dentro de los límites de especificación del histograma de Capacidad correspondiente.

5.4 Conclusiones

Ya que se trata de una labor institucional que da garantías a los sistemas desarrollados en la DDSI de la DGAI en el INEGI, se tuvo la ganancia de estandarizar el proceso de Pruebas, documentar el trabajo que se ha hecho por décadas, por primera vez definiendo conteos de defectos y la posibilidad de seguir investigando en relación a la aplicación de técnicas estadísticas que apoyen y den luz acerca de la confiabilidad del proceso de pruebas de cada sistema probado.

Un proceso de pruebas estandarizado tiene muchos beneficios, como lo es el tener un conjunto de actividades repetibles de pruebas, es decir, el cual garantiza que dicha actividades sean predecibles y con pocas fuentes de variación implícitas en el proceso de pruebas como lo son el aprendizaje del uso del sistema y la capacitación en las etapas tempranas del Proceso de Pruebas, garantizando así que la distribución de los errores restantes sea exponencial mediante una aproximación logarítmica. Metodológicamente el beneficio es que se podrá tener una guía de pruebas para realizar correctamente el proceso de pruebas de los sistemas que se pasen a revisión al Departamento de Pruebas Modulares a los Sistemas de Información.

Asimismo vale la pena mencionar que la problemática encontrada durante esta investigación en ningún momento desmotivó los objetivos primarios que son los de mejorar el proceso de Pruebas mediante la estandarización de las actividades independientemente de los insumos (entradas) proporcionados para la realización de las pruebas, pero si aplicando la metodología adecuada a cada uno de ellos para garantizar la detección confiable de defectos de los productos de software (salidas). Con esta idea se decidió además de contar los defectos encontrados en cada corte para ilustrar el decaimiento en el número de errores restantes (como inicio de la metodología de conteos de defectos), incluir un estadístico denominado NPR (número de prioridad de defectos) calculado al aplicar la técnica llamada AMEF, ampliamente utilizada en la industria automotriz para la prevención y detección de fallas o defectos de manufactura (tanto en la etapa de Diseño como de Procesos). Esto se logró adecuando al ámbito de pruebas de software, las calificaciones correspondientes a tres tablas que califican una falla o defecto encontrado mediante las variables de Severidad (gravedad del defecto), Ocurrencia (probabilidad de ocurrencia del defecto) y Detección (que califica los niveles de detección de defectos de los mecanismos de prueba), de esta manera este estadístico se calcula como: $NPR = S * O * D$. Cada

[Escribir texto]

variable tiene un rango de 1 a 10 de calificación, lo que permite tener un rango de [1,1000],y se podrán ordenar los defectos o fallas del producto probado en la plantilla o reporte correspondiente de acuerdo a la recomendación de prioridad de atención del encargado del desarrollo, siendo los números de NPR mayores los que requieren mayor atención.

5.5 Trabajo Futuro

- A partir de la presente investigación se continuará con el análisis y estudio de la técnica estadística SPClog, para aplicarla a los datos obtenidos de los Procesos de Pruebas tomados como casos de estudio y otros proyectos de pruebas en puerta.
- Se buscará compartir este trabajo con la Direcciones Regionales y Coordinaciones Estatales donde se pueda apoyar con pruebas a Proyectos de grandes dimensiones como los Censos nacionales a fin de garantizar que los productos de software y los sistemas funcionen correctamente durante su producción y produzcan información de calidad y conformidad con los requerimientos de los clientes y usuarios.
- Se seguirá investigando sobre software alternativo para la generación de casos de prueba a partir de los casos de uso documentados en UML, insistiendo en que este tipo de documentación se convierta en una cultura institucional, incluso promoviendo el desarrollo de herramientas que permitan generar casos de prueba a través de la aplicación de probabilidades de transición calculadas mediante Cadenas de Markov a los diagramas de secuencia.
- Perfil de personal en la Jefatura de Pruebas (Contratación, Capacitación)

El personal que desee ingresar al Departamento de Pruebas Modulares a Sistemas de Información de la DDSI deberá cumplir con los siguientes requisitos y someterse a la comprobación de sus conocimientos:

1.- Capacidad de Análisis y Crítica. Se requiere tener:

Pensamiento técnico (modelar tecnología, predecir el comportamiento de los sistemas).

Creatividad (generar ideas y posibilidades, se buscan los problemas que uno imagina que existen).

Pensamiento crítico (evaluar ideas y hacer deducciones, detectar y eliminar errores de nuestro pensamiento, vincular lo observado con los criterios de calidad).

[Escribir texto]

Pragmatismo (poner en práctica las ideas, adecuar las técnicas y el esfuerzo al alcance del proyecto).

2.- Aptitud de servicio y gestión y cultura de prevención.

3.- Conocimiento de Metodologías de desarrollo y Documentación (MÉTRICA,UML, etc.).

4.- Conocimiento de Metodologías de Pruebas (generación de Pruebas Modulares, Integrales, de Aceptación, etc.) y generación de Pruebas de Caja Negra.

5.- Experiencia en la aplicación de pruebas de sistemas de software.

Funciones:

- Analizar los requerimientos o requisitos solicitados por el usuario.
 - Definir detalles de Pruebas a partir del análisis de los requisitos del usuario.
 - Analizar detalles de pruebas.
 - Generar casos de prueba para los diferentes Casos de Uso (Funcionalidad) del Sistema.
 - Revisar productos de software terminados que garanticen el cumplimiento de los requerimientos de los usuarios.
 - Determinar y documentar los resultados de las pruebas.
 - Establecer mecanismos de comunicación para difundir los resultados de las pruebas y dar seguimiento a la corrección de errores.
 - Evaluar el funcionamiento de los sistemas y aplicaciones de Base de Datos Censales.
 - Identificar Metodologías de Pruebas.
 - Liberar los Sistemas.
-
- Se insistirá en la creación de sitios de colaboración (Sharepoint, sitios web, etc.) en los cuales se comparta tanto software para prueba como reportes de defectos o fallas de los sistemas que se encuentren en fase de pruebas.
 - Se seguirá trabajando en la detección de otras herramientas automáticas de apoyo al proceso, además de la herramienta solicitada denominada Usagetester, para agilizar el proceso e pruebas de sistemas y no constituir un cuello de botella al momento de liberar los sistemas a producción al cliente o usuario.
 - Se tratará de compensar los escenarios donde la documentación recibida de los sistemas a probar sea limitada, escasa, o ausente, mediante el

[Escribir texto]

aprovechamiento de la experiencia de los probadores para que en estas ocasiones puedan generar casos de prueba con solo el instrumento de captación, e incluso realizando reuniones con clientes o usuario para conocer sus expectativas de interés principales y esclarecer objetivos confusos.

5.6 Beneficios

- Gracias a los conocimientos adquiridos durante la etapa de preparación que se nos brindó durante la Maestría en Estadística Oficial (MCEO) del CIMAT, que nos formó y nos proveyó de herramientas necesarias para analizar los fenómenos del mundo real y aplicar las técnicas adecuadas con el debido apoyo y asesoría de nuestros maestros y asesores y a la inquietud de conocer más de los fenómenos que ocurren en nuestro ámbito. Emprendimos la aventura de tratar de obtener más información de los procesos institucionales en que nos vemos inmersos dentro de nuestro trabajo cotidiano.

[Escribir texto]

ANEXOS

ANEXO 1 – MONITOREO DEL PROCESO DE PRUEBAS.

A continuación se tradujo el artículo tomado como base como primer intento para realizar el monitoreo del proceso de pruebas de software mediante el Control Estadístico de Procesos, con la finalidad de aclarar cuáles fueron los esfuerzos por aplicar una técnica estadística al Proceso de Pruebas de Software en la Dirección de Desarrollo de Sistemas de Información de la DGAI en el INEGI, aunque desde un principio se nos aclaró por parte del asesor de tesis que se requería evaluar cuantitativamente los errores (defectos o fallas) encontradas durante una revisión típica de un software, ya que solamente se reportaban errores comentados. En un principio se pensó en salvar este escollo pidiendo ayuda de un experto en Control Estadístico de Calidad, solicitándole la revisión típica de un sistema de desarrollo al Censo Agropecuario 2007. Al tratar de “ponerle números” a los errores encontrados durante la revisión en las diferentes versiones del Sistema desarrollado hasta su puesta en marcha, nos encontramos con que el Proceso de Pruebas se trataba de un proceso NO estable, como nos comentó el co-asesor el M. en E. Sergio Nava Flores y es reseñado desde la introducción del presente trabajo.

Este material proporcionado por el asesor, se basa en el artículo llamado: **“Monitoreando el Proceso de Prueba del Software (STP) usando Control Estadístico del Proceso (SPC): Un Acercamiento Logarítmico”**, publicado por **J. W. Cangussu, R. A. DeCarlo, and A. P. Mathur.**

El objetivo era conocer en qué momento un proceso de desarrollo de software, sujeto a un Proceso de Pruebas de software, se encontraba bajo control o estable (dentro de los límites de control especificados) y así poder después pensar en buscar y hacer uso de un indicador de confiabilidad del software durante su producción.

A partir del análisis de este artículo, que motivó la aplicación, a continuación se detalla la técnica que se deseaba aplicar y que no fue posible aplicarlo por el hecho de no contar con un Proceso de Pruebas Estable (en el cual no se siguiera el mismo procedimiento independientemente de los insumos que se entregan para su revisión):

El control estadístico de procesos (SPC por sus siglas en inglés) es una herramienta de gran alcance para controlar la calidad de procesos. Asiste al personal de la gerencia en la identificación de los problemas y de las acciones que se tomarán para

[Escribir texto]

llevar un proceso a un estado estable. El control estadístico de procesos se ha aplicado en varios campos, incluyendo el proceso del desarrollo del software. Sin embargo, algunos procesos son caracterizados mejor por los factores que exhiben un comportamiento exponencial. El uso de esos factores sirve para calcular los límites de control del proceso, en la aplicación de las técnicas tradicionales del proceso estadístico. El Proceso de Prueba de Software (STP por sus siglas en Inglés) caracterizado por el decaimiento en el número de errores restantes, de la intensidad de la falla, y de un aumento en cobertura del código, es uno de esos procesos. Se propone una variante de la técnica tradicional de Control Estadístico del Proceso (SPC). Esta variante utiliza la transformación logarítmica para permitir el control estadístico de los procesos (SPC) cuyo comportamiento dominante es descrito mucho mejor por una exponencial. Una evaluación de la transformación propuesta realizada usando simulación y un estudio de caso de un proyecto industrial, anima el uso de la variante propuesta al STP.

A1.1 INTRODUCCIÓN

El SPC es un método del control de calidad basado en la supervisión de un proceso experimental. El proceso estadístico fue idealizado para el control de calidad de las cadenas de producción [1]. Por ejemplo, cuando las instancias múltiples de un producto, digamos un remache, van a ser producidos dentro de límites especificados de tamaño, el SPC se puede utilizar para comprobar si el número de casos dentro de estos límites es aceptable y si el proceso entero alcanza la producción deseada dentro de los límites de control para una duración especificada. El éxito del SPC para controlar la calidad en líneas de producción condujo a su uso en otras áreas tales como química, electrónica, alimentos, empaquetado, y desarrollo del software [2, 3, 4, 5, 6]. El enfoque de las seis Sigma para el mejoramiento de la calidad de negocios y de procesos del software es otro ejemplo de una metodología basada en el proceso estadístico [7, 8, 9].

La aplicación del SPC y de sus derivaciones se restringe a los procesos cuyas medidas de la calidad deseadas **no varían en el tiempo**. Sin embargo, la calidad y el progreso de algunos procesos es analizada mucho mejor monitoreando y midiendo los factores que presentan un decaimiento o crecimiento exponencial. El Proceso de Prueba del Software (STP) [10, 11, 12] y el decaimiento de la eficacia de algunos filtros usados en una máquina de la diálisis renal son dos ejemplos de tales procesos.

[Escribir texto]

Como se explicó antes, **las técnicas tradicionales de SPC no permiten el control de procesos que presentan tal comportamiento.**

El uso de promedios móviles ponderados [13] tampoco trata el control estadístico de procesos con las características que son exponenciales.

En este artículo nos centramos en el control del STP usando los factores de calidad clave/dominantes. Sin embargo, la técnica propuesta se puede utilizar para monitorear cualquier proceso con los factores que exhiben comportamiento exponencial. La cobertura del código, el número de errores restantes, y la intensidad de la falla son ejemplos de tres factores de calidad en el STP que exhiben un comportamiento exponencial [12, 14].

Para un análisis del enfoque presentado aquí, nos centramos en el número de errores encontrados por unidad del tiempo.

Se propone una variante del SPC basado en una transformación logarítmica para permitir la supervisión y el control estadístico del STP usando el decaimiento en el número de errores restantes en el producto bajo prueba. Puesto que el enfoque de este artículo es sobre el STP, uno podría preguntarse porque no usar los modelos de crecimiento de Confiabilidad existentes, en vez del SPC, para monitorear el proceso. Los modelos de crecimiento de Confiabilidad se enfocan en la predicción y no en la supervisión. Es decir, no tratan los resultados del tiempo para hacer cambios en el proceso.

Suponga que una desviación del 1% de la confiabilidad prevista es observada. ¿Debe el supervisor realizar cambios en el proceso o es la variación demasiado pequeña para ser tomada en consideración? Ésta es una pregunta entre otras, no tratadas por los modelos del crecimiento de Confiabilidad, justificando el uso de la variación del proceso estadístico descrita aquí.

El resto de este trabajo se organiza como sigue. Por motivos de ofrecer una mayor claridad y sentido de congruencia, se presenta una breve introducción al SPC en la sección A1.2. Las dificultades en la aplicación del SPC al STP, y una variante del SPC para hacer su uso posible, se presentan en la sección A1.3. La sección A1.4 describe un método usado para evaluar la variante propuesta. La sección A1.5 presenta un análisis de los resultados obtenidos usando el método de la evaluación. Un resumen de este trabajo y las conclusiones del mismo se encuentran en la sección A1.6.

[Escribir texto]

A1.2 Control Estadístico de Calidad.

Control Estadístico de Calidad, como lo describe Florac y Carleton [15], es una herramienta del análisis para mejorar el control del proceso del desarrollo del software (SDP). Los dos conceptos de la estabilidad y de la capacidad en el proceso estadístico se aplican directamente al control del SDP.

Estabilidad [15]: Un proceso es estable si varía dentro de límites predichos.

Si la variación está más allá de estos límites, el proceso se considera inestable (y no bajo control).

Una carta de control se utiliza comúnmente para determinar si un proceso está bajo control o no, es decir, estable. Existen varios tipos de cartas de control: X-Barra, Range Charts, U Charts, y Z Charts son algunos ejemplos. Cada carta se diseña para ser utilizada basada en las condiciones específicas de los datos disponibles. Ejemplificamos el uso de una carta del control describiendo una carta de la X-Barra [16]. En la figura A1.1 se muestra el ejemplo de la medición de los errores encontrados durante 14 ciclos de prueba, observándose que se encuentran dentro de los límites de control establecidos.

No. de errores
Normalizados

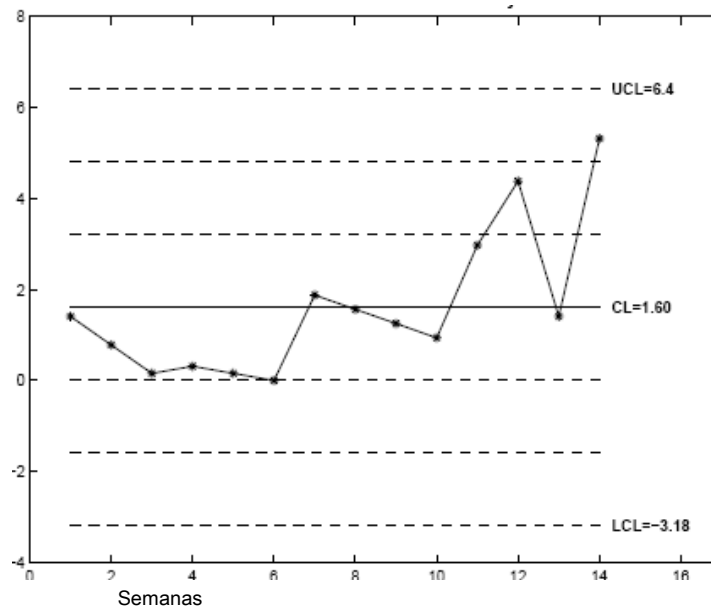


Figura A1.1: Grafica de control X-barra para un proyecto Comercial

Una carta X-barra se basa en promedios de los datos agrupados. Requiere que los datos estén agrupados en subgrupos por lo menos de tamaño dos. Los datos para

[Escribir texto]

cada subgrupo i se promedian (\bar{X}_i) y se calcula la desviación de estándar (σ_i) para el subgrupo. Entonces, el promedio de los promedios de los subgrupos ($\bar{\bar{X}}$) se calcula junto con la desviación de estándar media ($\bar{\sigma}$). Basadas en estos valores, se dibujan tres líneas en la carta de la X-Barra: (CL) la línea de centro que es $\bar{\bar{X}}$, (UCL) el límite de control superior $\bar{\bar{X}} + 3\bar{\sigma}$, y (LCL) el límite de control inferior $\bar{\bar{X}} - 3\bar{\sigma}$.

Un ejemplo de la carta de la X-Bar se muestra en la figura A1.1. En este trabajo asumimos el límite de control estándar ($\bar{\bar{X}} \pm k\bar{\sigma}$) para $k = 3$. Sin embargo, según lo precisado por Jalote y Saxena [17], un límite de control optimizado se puede establecer para mejorar el empate de las características específicas de un proceso de software. Este cambio en el límite de control no afectará la variante del SPC presentada aquí.

El análisis de una carta de la X-barra determina estabilidad de proceso. Un proceso se dice para estar fuera del control (inestable) si cualquiera de las pruebas siguientes ocurre [15]:

Test 1: Un punto está fuera de los límites de LCL y de UCL.

Test 2: Al menos dos de tres valores sucesivos caen en el mismo lado y más de 2σ unidades alejados de la línea central.

Test 3: Al menos cuatro de cinco valores sucesivos caen en el mismo lado de, y más allá de 1σ unidad lejos de, la línea central.

Test 4: Al menos ocho puntos sucesivos caen del mismo lado de la línea central.

Observamos en la figura A1.1 que según las pruebas definidas arriba el proceso bajo observación está bajo control, es decir, el proceso es estable. Aunque el proceso es estable, el número promedio de errores quitados es cerca de 1.8% debajo de su valor esperado. Los datos del proyecto de Razorfish se presentan aquí normalizados a unas 100 unidades. Por lo tanto, cada unidad representa el 1%. La línea central, determinada por el Supervisor es 3.4, mostrando un 1.8% de desviación comparado con la línea central en la figura A1.1, entonces necesitamos analizar si el proceso es capaz o no de completar la tarea predeterminada cuando es restringido a un costo

[Escribir texto]

predeterminado y a un esquema. La desviación de la línea central no provee suficiente evidencia que indique que el proceso no es capaz.

Capacidad [15]: Un proceso es capaz si puede alcanzar los resultados previstos dentro de un tiempo especificado. La capacidad se basa en la probabilidad de que el proceso puede alcanzar los resultados dentro del tiempo especificado. Esto presupone claramente que el proceso es estable.

En la figura A1.2 se muestra un histograma de Capacidad para el proyecto Razorfish:

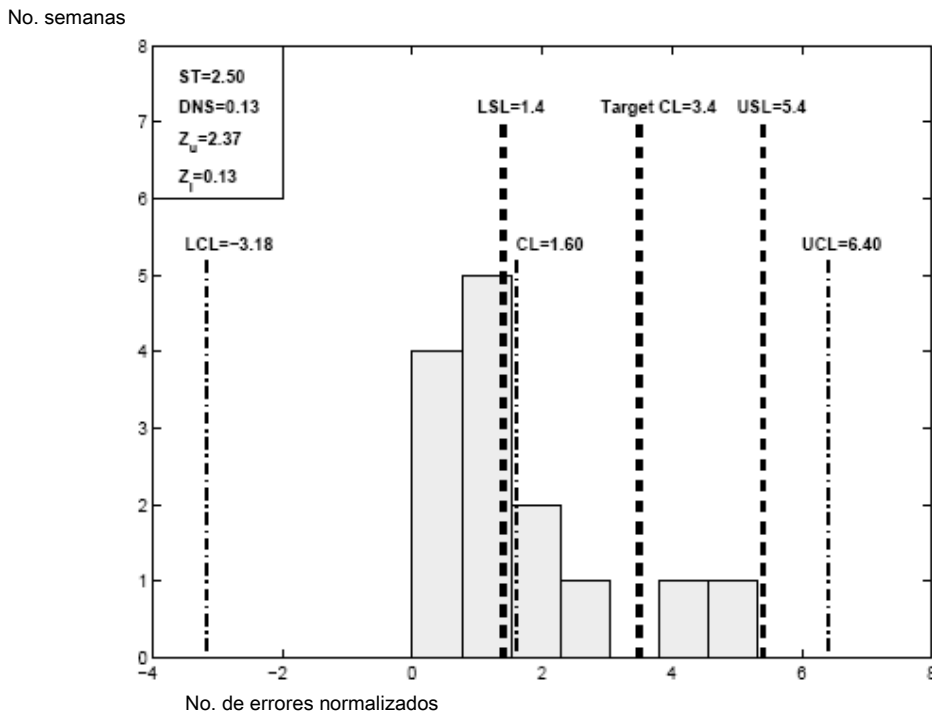


Figura A1.2: Histograma de Capacidad del Proceso para un proceso Industrial

Un histograma de capacidades tal como el de la figura A1.2 se puede usar para completar esta tarea determinando si un proceso está fuera de los límites esperados. El estar fuera de los límites esperados representa una alta probabilidad de generar resultados no conformables e indica que el proceso debería ajustarse para lograr esos resultados.

Las ecuaciones para determinar la capacidad del proceso son presentadas ahora:

$$ST = \frac{USL - LSL}{\sigma} \quad (1)$$

[Escribir texto]

$$DNS = \min\{Z_u, Z_l\} \quad (2)$$

$$Z_u = \frac{USL - \bar{X}}{\sigma} \quad (3)$$

$$Z_l = \frac{\bar{X} - LSL}{\sigma} \quad (4)$$

Donde \bar{X} es el promedio de los valores observados durante el periodo bajo consideración. ST es la Tolerancia de especificación, USL es el límite superior de especificación, LSL es el límite inferior de especificación⁶, DNS es la distancia a la especificación más cercana, Z_u es la distancia en σ unidades entre el promedio del proceso y USL; y Z_l es la distancia en σ unidades entre el promedio del proceso y LSL. Wheeler dice: "...sigma unidades expresa el número de unidades de medida que corresponde a una unidad estándar de dispersión" [1], las sigma unidades se usan para caracterizar cuantos de los datos está dentro de una distancia alfa de la media observada [1].

En la Figura A1.3 se presentan los pasos para el Control Estadístico de Procesos:

⁶ Ambos USL y LSL son límites determinados por el supervisor de acuerdo a sus expectativas.

[Escribir texto]

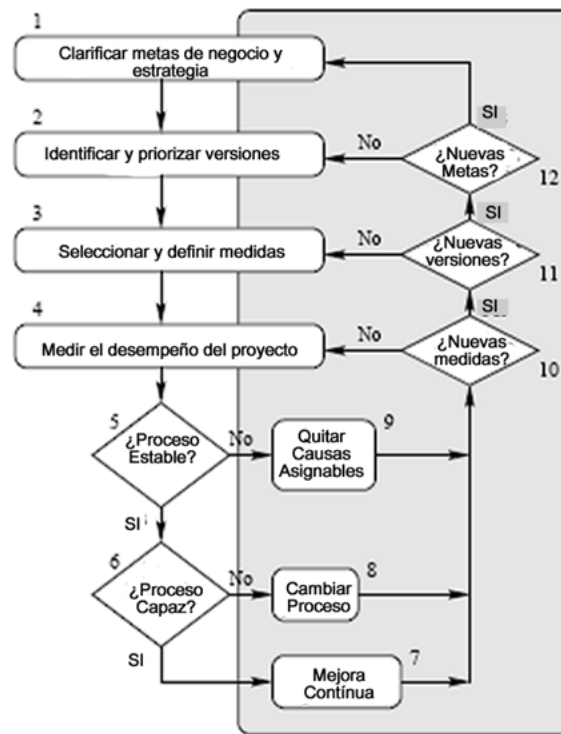


Figura A1.3: Secuencia de pasos para el Control Estadístico de Procesos

En la figura A1.3 observamos los pasos que siguen a la supervisión de un proceso cuando se usa SPC [15]. El paso 5 representa una de las pruebas de estabilidad listadas anteriormente. La prueba de Capacidad es representada por el paso 6. Los detalles de cómo SPC determina si un proceso puede o no puede ser capaz se proveen en la figura A1.4.

[Escribir texto]

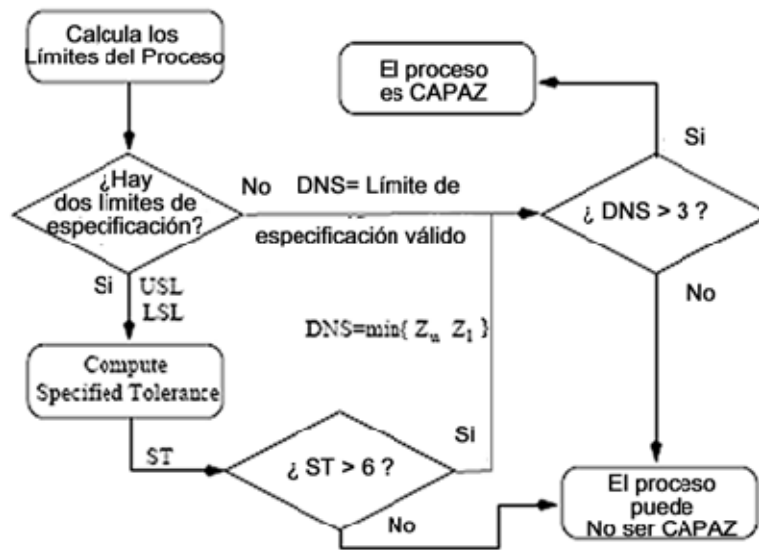


Figura A1.4: Verificación de la Capacidad para un Control Estadístico de Procesos[15]

A1.3 Aplicando SPC a STP

Ahora explicaremos el porqué un STP es caracterizado por una conducta exponencial y entonces derivaremos las razones del porqué un SPC tradicional no se aplica para controlar el STP. Siguiendo esta discusión introducimos en la sección A1.3.3 una variante del SPC basada en la transformación logarítmica, que a partir de aquí llamaremos SPClog, la cual hace posible controlar el STP y otros procesos que presentan una conducta exponencial.

A1.3.1 ¿ Presenta actualmente el STP una conducta exponencial ?

SPClog como se describe en la sección A1.3.3 se puede aplicar a cualquier proceso que presente una conducta exponencial. Aquí usamos STP para ejemplificar la aplicabilidad de esta técnica. Ya se sabe con amplia experiencia que los defectos son relativamente fáciles de encontrar durante las etapas tempranas de una fase de pruebas de un sistema. Quizás porque hay muchos de ellos y no es difícil filtrarlos [18]. Encontrar defectos se hace difícil con el paso del tiempo quizás porque hay menos de ellos y con frecuencia requieren de combinaciones específicas de eventos para ser filtrados. Esta conducta implica una pendiente exponencial por el decaimiento en el número de defectos acumulados, y como los defectos son quitados, la intensidad de la falla y la confiabilidad también presentan una conducta exponencial [12]. Sin embargo, en algunas situaciones el STP presenta una forma S más que una forma exponencial. Esta conducta puede deberse a factores tales como el aprendizaje y la adaptación presentes al inicio de un STP [19]. En ambos casos, la suposición aquí es que el STP es un proceso bien definido, no ad hoc, Nuestro interés es con procesos donde el aprendizaje y la adaptación están minimizados y la conducta dominante puede ser caracterizada como una exponencial.

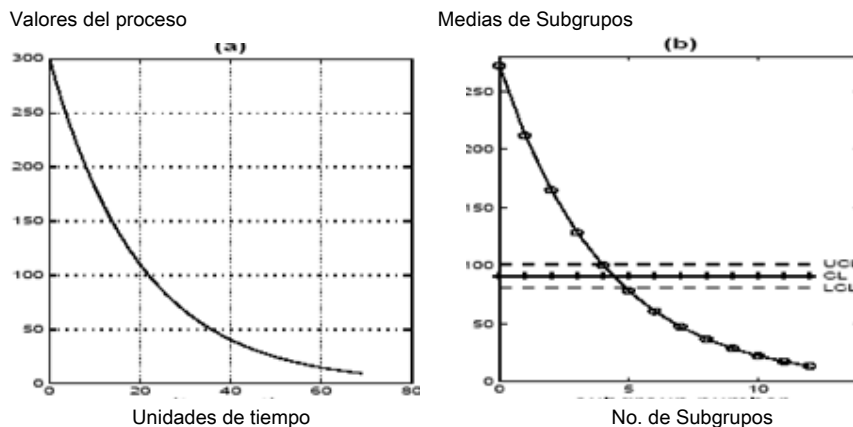


Figura A1.5: Ejemplo de un proceso que presenta un decaimiento exponencial.

[Escribir texto]

Los investigadores han proporcionado la evidencia de que los procesos de prueba presentaban un comportamiento exponencial. El registro de errores divulgado por Knuth tiene un decaimiento exponencial claro durante la fase de prueba [20]. El número acumulado de averías planteado en el trabajo de Dalal [21] mostró un comportamiento exponencial que también se observa en el proyecto de Razorfish [18]. El modelo de crecimiento de la confiabilidad de Jacoby y de Tohma [22] presenta casos de ambas curvas: exponencial y S. Además, Ehrlich [23] presentó ejemplos de fallas acumuladas exponencialmente y decaimiento de la intensidad de la falla.

A1.3.2 Dificultades en la aplicación de técnicas tradicionales de SPC al control de calidad del STP.

El uso del SPC no asume ninguna distribución específica de los valores del factor que es observado para el control de proceso y es aplicable a una variedad de distribuciones tales como la uniforme, la exponencial, la normal, la triangular, y la Chi-cuadrada [1].

Este hecho es justificado por los experimentos que revelan que por lo menos el 95% de los valores observados caen entre los límites 6σ [1], es decir, entre los límites de control más bajo y más alto. La suposición en todos los casos es que los valores del factor observado están distribuidos aleatoriamente. Esta suposición plantea la pregunta siguiente: "¿qué sucede si un proceso produce valores correlacionados que no se distribuyen aleatoriamente?".

Como un ejemplo de valores correlacionados considere un proceso que presenta un decaimiento exponencial esperado según lo demostrado en la figura A1.5a. Por "esperado" nos referimos a los valores (v_j) que se deben producir por un proceso casi perfecto. Asumiendo subgrupos de tamaño δ ($\delta = 5$ en la figura A5) y calculando la media (\bar{X}_j) y la desviación estándar (σ_j) de cada subgrupo produce los resultados demostrados en la figura A1.5b. Del análisis de cada subgrupo se observa que la desviación estándar es más pequeña que la diferencia entre el primer valor (el más grande) y el último valor (el más pequeño) de cada subgrupo. La desviación estándar más grande será entonces más pequeña que la más grande diferencia entre δ puntos

[Escribir texto]

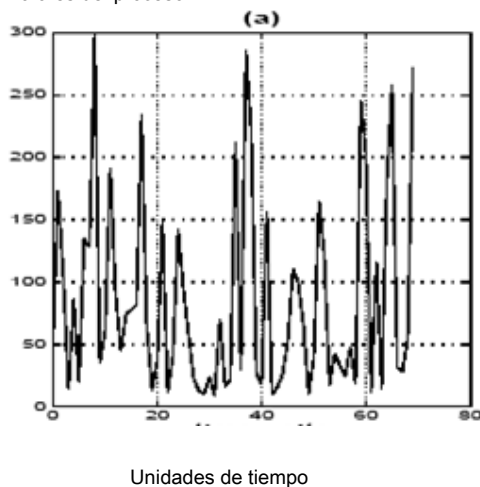
consecutivos ($\sigma_i < v_{ix\delta} - v_{(i-1)x\delta+1}$), la desviación estándar más grande entonces será más pequeña que la diferencia más grande entre los δ puntos consecutivos:

$$(\sigma_{max} < Max(v_{ix\delta} - v_{(i-1)x\delta+1}), para i = 1, .., n_s)$$

, donde n_s es el número de subgrupos. Entonces, la desviación estándar promedio será menor que el valor máximo, $\bar{\sigma} < \sigma_{max}$. Los límites de control del proceso son $\bar{X} \pm 3 \times \bar{\sigma}$.

Se puede mostrar que si el número de subgrupos es mayor que 2: ($n_s > 2$), muchos puntos caerán fuera de estos límites como se observó en la figura A1.5b. En este caso, de acuerdo a la carta X-barra de la figura A1.5b, el proceso no es estable mientras que presenta de hecho un decaimiento exponencial esperado perfecto. Incluso el uso de una técnica media móvil cargada exponencialmente [13] para calcular los límites de control no soluciona el problema. Asuma ahora que los valores (v_j) de la figura A1.5a se distribuyen aleatoriamente como se ve en la figura A1.6a. Construyendo la carta de la X-barra con la misma configuración que arriba conduce a los resultados demostrados en la figura A1.6b. Puede observarse de esta figura que todas las pruebas definidas en la sección A1.2 son aceptadas y por lo tanto el proceso es estable.

Valores del proceso



Medias de Subgrupos

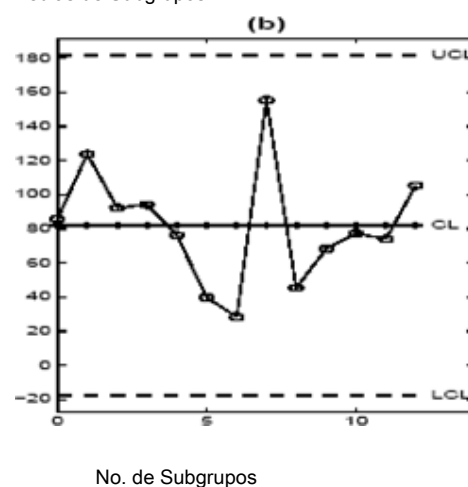


Figura A6: Permutación aleatoria de los valores del proceso de la Figura A5.

[Escribir texto]

La conclusión que dibujamos de la discusión anterior es que la suposición de una distribución aleatoria de valores prohíbe el uso de un SPC a un proceso que presenta un comportamiento exponencial. El STP puede ser caracterizado y/o ser monitoreado analizando el número de los errores encontrados por unidad del tiempo o el aumento en cobertura del código. En ambos estos casos, se observa un comportamiento exponencial, un decaimiento en el caso del número de los errores encontrados por unidad de tiempo y un aumento en el caso de la cobertura del código. Aunque el STP no presentará muy probablemente un decaimiento exponencial perfecto, el comportamiento será bien aproximado por una exponencial y presentará las características que provienen el uso del SPC.

A1.3.3 SPClog

Según lo mencionado anteriormente, el SPC espera que la variación del proceso ocurra sobre una línea horizontal. Sean $v_i, i = 1, \dots, n$ los valores observados para un proceso. Una transformación logarítmica permite más tarde el uso del SPC para controlar un proceso con comportamiento exponencial. El primer paso es calcular el logaritmo natural de los datos observados para cada valor $\log(v_i), i = 1, \dots, n$. Esta transformación genera un decaimiento lineal total más bien que un decaimiento exponencial. El segundo paso es calcular un ajuste de mínimos cuadrados de $\log(v)$ produciendo $LSF_i, i = 1, \dots, n$. Esta aproximación captura un decaimiento lineal promedio de los datos. La variación se puede ahora observar sobre una línea con una pendiente negativa pero no cero (línea horizontal) según lo esperado por SPC. Ahora preguntamos si o no los valores del $\log(v)$ están bajo control según las pruebas definidas en la sección 2. Para obtener una respuesta utilizamos la distancia euclidiana (δ) en la Ecuación 5 para calcular la distancia entre cada punto del $\log(v)$ y el punto respectivo en el ajuste de mínimos cuadrados LSF_i como en la Ecuación 6. Los dos puntos bajo consideración aquí están siempre en la misma posición respecto al eje del tiempo y la distancia por lo tanto se mide considerando solamente el eje vertical como en la ecuación 5.

$$\delta_i = \sqrt{(x(i)_1 - x(i)_2)^2 + (y(i)_1 - y(i)_2)^2} \quad x(i)_1 = x(i)_2$$

→

$$\delta_i = \sqrt{(y(i)_1 - y(i)_2)^2} \quad (5)$$

[Escribir texto]

Calculando la distancia euclidiana entre los \log valores y el ajuste de mínimos cuadrados, como en la ecuación 6, genera una variación lineal con un decaimiento con pendiente cero permitiendo el uso de las técnicas tradicionales del SPC.

$$x_i = \sqrt{(LSF_i - \log(v_i))^2}, i = 1, \dots, n \quad (6)$$

Asumiendo subgrupos de tamaño m , el promedio y la desviación estándar se pueden ahora calcular para cada subgrupo.

$$\bar{x}_i = \frac{\sum_{j=i}^{i+m-1} x_j}{m} \quad \text{para } i = 1, 1 + m, 1 + 2m, \dots, n \quad (7)$$

$$\sigma_i = \sqrt{\frac{\sum_{j=i}^{i+m-1} (x_j - \bar{x}_i)^2}{m}}$$

para $i = 1, 1 + m, 1 + 2m, \dots, n \quad (8)$

Finalmente, la línea central ($\bar{\bar{x}}$) y los límites de control son calculados como en el SPC y el proceso puede ser comprobado para la estabilidad. Un listado compacto de los pasos para comprobar la estabilidad de un proceso que presenta un decaimiento exponencial se proporciona en la tabla 1.

Los pasos anteriores determinan si un proceso es estable o no. En el caso de que se observe estabilidad, el proceso necesita ser analizado para capacidad. Un histograma de capacidad se puede trazar a partir de los promedios de los subgrupos \bar{x}_i junto con los límites de control calculados.

Bajo la suposición de que el proceso tiene un comportamiento exponencial esperado, el decaimiento esperado (Target o Línea Central) y un valor inicial se debe proporcionar por el encargado de proceso. Los límites, superior e inferior, USL y LSL, respectivamente, se deben también proporcionar por el encargado. Asumamos que los valores previstos son definidos por $e_v(t) = v_0 e^{-\lambda t}$. Entonces el valor inicial v_0 y el parámetro λ de decaimiento esperado son proporcionados por el encargado de proceso.

[Escribir texto]

Un primer intento para generar los límites de especificación superior e inferior fue derivado directamente del decaimiento exponencial esperado. Asuma que el encargado acepta una variación Δ para el parámetro del decaimiento y el valor inicial. Bajo estas condiciones, los límites superior e inferior se pueden calcular como en las ecuaciones 9 y 10.

$$e_v^{up}(t) = \log((1 + \Delta) * v_0 e^{-((1-\Delta)*\lambda)*t}) \quad (9)$$

$$e_v^{low}(t) = \log((1 - \Delta) * v_0 e^{-((1+\Delta)*\lambda)*t}) \quad (10)$$

Como se muestra en la figura A1.7 este enfoque produce límites de especificación divergentes. El cálculo del DNS en la ecuación 2 y ST en la ecuación 1 se vuelve más sensible al uso de este enfoque a medida que el valor de t aumenta. Es decir, debido a los límites de especificación divergentes la distancia euclidiana promedio con el tiempo y así lo hacen los valores del DNS y del ST.

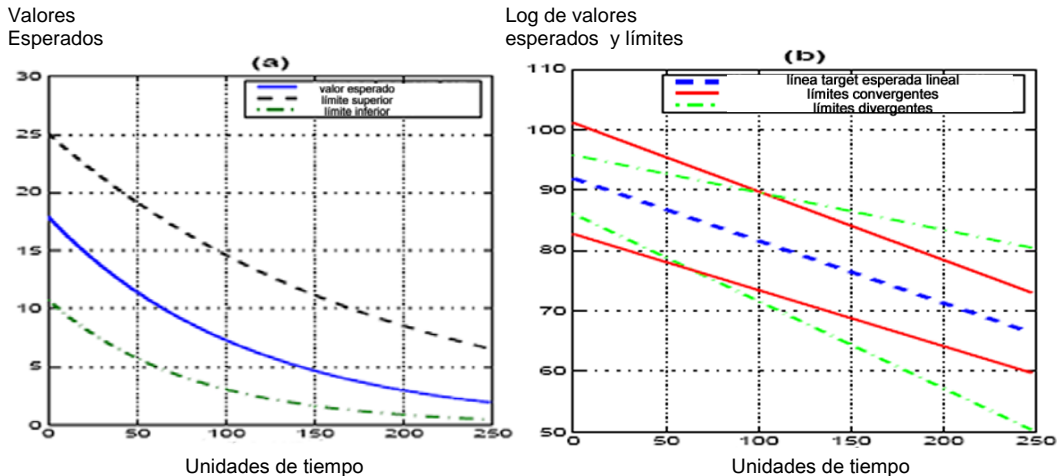


Figura A1.7: Ejemplo del comportamiento esperado, límites superior e inferior y la transformación logarítmica correspondiente.

Una manera alternativa de definir los límites de la especificación es aplicar la variación Δ a log del decaimiento esperado en vez de aplicarlo al decaimiento exponencial. Las ecuaciones 11 y 12 se utilizan para alcanzar esta meta. Esto resulta en límites convergentes más que divergentes que parecen ser más apropiados para el cálculo del DNS y del ST. La figura A1.7 dibuja los dos límites convergentes.

$$e_v^{up}(t) = (1 + \Delta) * \log(e_v) \quad (11)$$

[Escribir texto]

$$e_v^{low}(t) = (1 - \Delta) + \log(e_v) \quad (12)$$

Calculando $\log(e_v)$ produce un decaimiento lineal esperado (e_{td}) que se puede comparar con el decaimiento lineal medio observado derivado del ajuste de mínimos cuadrados. Es decir, estamos interesados en calcular que tan distantes están los valores observados medios (determinados por el ajuste de mínimos cuadrados) de los valores esperados de la estimación del encargado del proceso. Esta distancia representa la línea central (TCL). La línea central \bar{x} en el histograma de capacidad se calcula usando la distancia promedio de $\sqrt{(\log(v) - LSF)^2}$. Para comprobar que tan lejos está la línea observada de la línea esperada, necesitamos calcular el promedio basado en el mismo conjunto de puntos. Así utilizamos la distancia euclidiana entre el e_{td} y el $\log(v)$ y no entre e_{td} y LSF que conducen al cálculo del TCL como en la ecuación 13.

$$TCL = \bar{\delta}, \text{ para } \delta_i = \sqrt{(e_{td}(i) - \log(v_i))^2} \quad (13)$$

El cómputo de e_v^{up} y de e_v^{low} produce los límites lineales superior e inferior para el proceso como se observa en la figura A1.7.

La distancia euclidiana promedio entre estos límites y el decaimiento esperado (e_{td}) se utiliza para calcular el USL y el LSL en las ecuaciones 14 y 15, respectivamente.

$$USL = TCL + \bar{\delta}^{up}, \text{ para } \delta_i^{up} = \sqrt{(e_v^{up}(i) - e_{td}(i))^2} \quad (14)$$

$$LSL = TCL - \bar{\delta}^{low}, \text{ para } \delta_i^{low} = \sqrt{(e_{td}(i) - e_v^{low}(i))^2} \quad (15)$$

El USL y el LSL se utilizan para calcular el ST en la ecuación 1 y el DNS en la ecuación 2. Finalmente, la capacidad del proceso se puede comprobar cómo se muestra en la figura A1.4. Un resumen de los pasos para comprobar capacidad de proceso se presenta en la tabla A1.1.

Tabla A1.1: Pasos en la aplicación del enfoque SPC_{log}

Paso	Estabilidad	Capacidad
1	$ld_i = \log(v_i), i = 1, \dots, n$	Calcular el decaimiento esperado $e_v(t) = v_0 e^{-\lambda t}$
2	Calcula el ajuste de mínimos	Calcula el log de el valor esperado ($\log(e_v)$) y los límites

[Escribir texto]

	cuadrados $(lsf_i - ld_i)^2$	de especificación superior ($e_v^{up} = (1 + \Delta) * \log(e_v)$) e inferior ($e_v^{low} = (1 - \Delta) * \log(e_v)$) para una Δ dada.
3	Calcular $x_i = \sqrt{(lsf_i - ld_i)^2}$	Calcular TCL, el promedio de la distancia euclidiana entre los promedios esperado y observado ($TLC = \sqrt{(e_{ld}(t) - \log(v_i))^2}$)
4	Usando $x_i, i = 1, \dots, n$, calcular los límites como en SPC	Calcula $USL = TLC + \sqrt{(e_v^{up}(t) - e_{ld}(t))^2}$ y $LSL = TLC - \sqrt{(e_{ld}(t) - e_v^{low}(t))^2}$.
5	Checar estabilidad usando pruebas SPC	Checar capacidad usando pruebas SPC

A1.4. EVALUACIÓN DEL SPC_{log}

La técnica de SPC_{log} presentada aquí se evalúa usando la simulación y los datos colectados de un proyecto comercial.

El ejercicio de simulación corre SPC_{log} de diversas maneras dando por resultado tres diferentes escenarios, dos de los cuales se describen en la sección A1.4.1. La segunda parte del proceso de evaluación ha demostrado la aplicabilidad de SPC_{log} a un proyecto real pero su análisis está más allá del alcance de este artículo.

A1.4.1 Escenarios de simulación.

Tres escenarios son de interés (i) un proceso inestable, (ii) un proceso estable y capaz, y (iii) un proceso estable pero no capaz. Debido a las limitaciones del espacio solo los primeros dos de los tres escenarios se presentan aquí. Estos dos escenarios se basan en perturbaciones aleatorias de un proceso con decaimiento exponencial. Teniendo un decaimiento esperado determinado por $v_0(t) = v_0 e^{-\lambda * t}$, las perturbaciones son insertadas mediante el cálculo de $v_i = \alpha_i * v_e$, donde $0 < \alpha_i < 2$. Es decir, el proceso se hace lento cuando $\alpha_i < 1$ y se acelera cuando $1 < \alpha_i < 2$. Las probabilidades de estos dos casos se controlan para acelerar o desacelerar el proceso y producir los escenarios deseados.

El tamaño del subgrupo se define como 5 para calcular el promedio semanalmente. Sin embargo, la técnica presentada aquí no impone ninguna restricción ante tamaños del subgrupo y diversos valores pueden ser utilizados.

Escenario I

El primer escenario representa un proceso de prueba inestable del software. En este caso, la capacidad no se analiza como si el proceso estuviera en condiciones estables antes de que se considere la capacidad. Las figuras A1.8 y A1.9 muestran los

[Escribir texto]

resultados de este escenario que se analizan en la sección A1.5.1. El gráfico en la figura A1.8a fue generado de acuerdo a la perturbación aleatoria de un proceso con decaimiento exponencial descrito arriba mientras que la figura A1.8b representa los valores logarítmicos correspondientes junto con un ajuste de mínimos cuadrados.

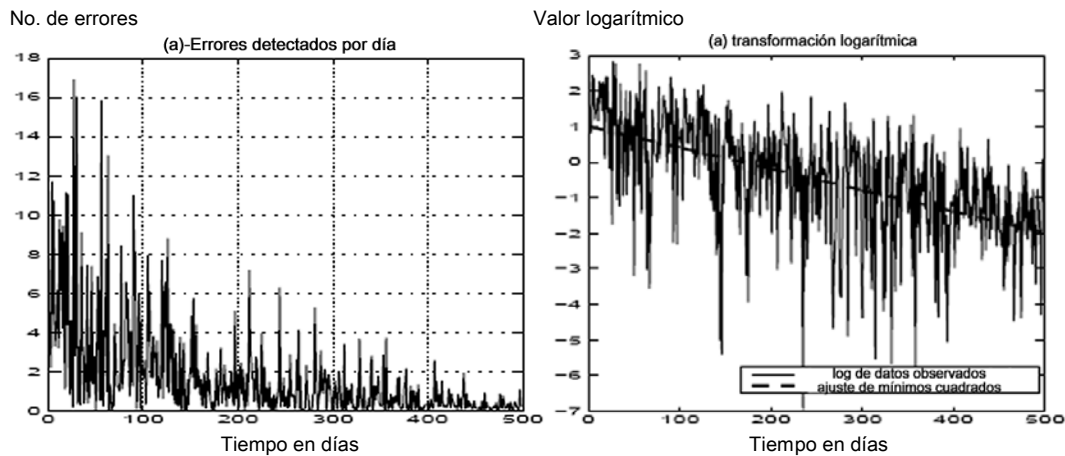


Figura A1.8: Resultados de la simulación Escenario I: Figura (a) representa los datos generados y la Figura (b) muestra la transformación logarítmica correspondiente junto con el ajuste de mínimos cuadrados.

Escenario II

El escenario II de la simulación ilustra un STP estable y capaz. La capacidad para este escenario fue calculada usando una distribución normal para generar probabilidades iguales de $0 < \alpha_1 \leq 1$ y de $0 < \alpha_2 < 2$, con igual probabilidad de aceleración o desaceleración del proceso.

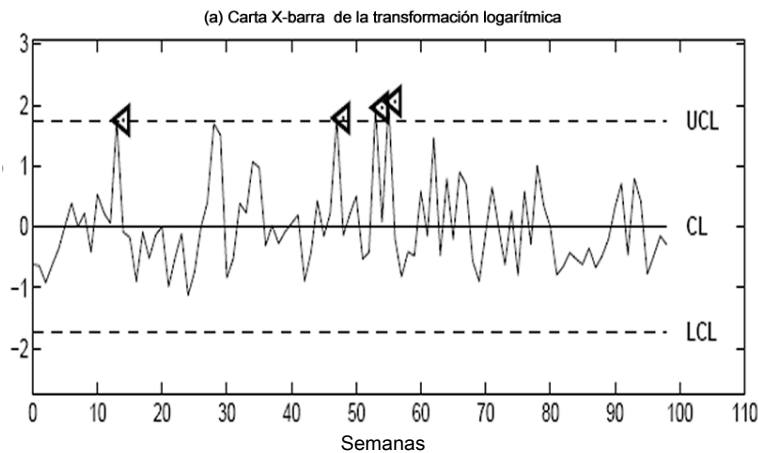
A5. ANALISIS DE LOS RESULTADOS DEL SPC_{log}

A5.1 Escenario I.

[Escribir texto]

Un resultado de la simulación para el escenario I se representa en las figuras A1.8 y A1.9. Como observamos en la figura A1.9a, el proceso no es estable porque cuatro puntos caen fuera del límite 3σ , es decir fuera del límite de control superior (UCL). Los puntos están fuera de los límites debido a una secuencia de los valores pequeños o grandes bajos del número de errores detectados v_i, \dots, v_{i+4} dentro de la misma semana (subgrupo) generando puntos en la transformación logarítmica (figura A1.8b) que están lejos del ajuste de mínimos cuadrados. Sea lsf_i, \dots, lsf_{i+4} el ajuste de mínimos cuadrados de la transformación logarítmica para la semana i . Cuando el cálculo de $\sqrt{(lsf_j - \log(v_j))^2}, j = i, \dots, i + 4$ conduce a una secuencia de valores pequeños o grandes (los puntos son distancias del ajuste de mínimos cuadrados) aumenta el promedio calculado para la carta X-barra para SPC_{log} generando puntos sobre los límites 3σ . La relación del número real de errores con los puntos fuera de los límites puede ser observada comparando los "triángulos" de las figuras A1.9a y A1.9b. Una secuencia de los valores bajos para la detección de error aumenta el promedio $\sqrt{(lsf - \log(v))^2}$ y genera un punto sobre el límite de control superior (UCL) identificando un proceso inestable.

Promedios semanales



Promedios semanales

[Escribir texto]

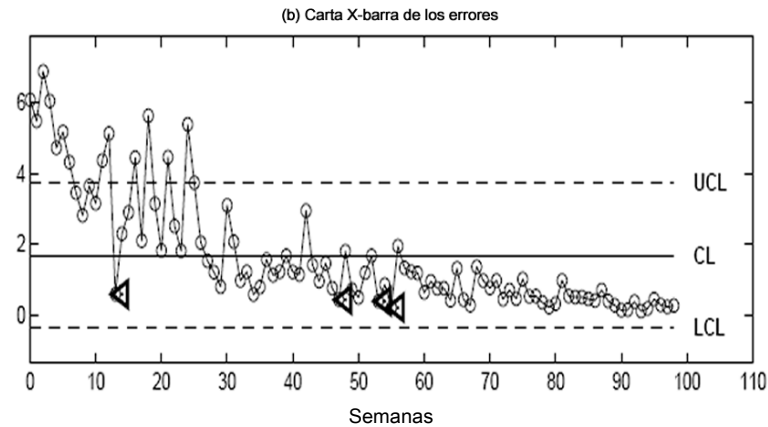


Figura A1.9: Resultados de la simulación escenario I. (a) la carta X-barra para la distancia euclidiana entre los log(errores) y el ajuste de mínimos cuadrados. (b) la carta de la X-barra para el número actual de errores.

También se nota de la figura A1.9a que una carta X-barra para el número real de errores producirá muy probablemente puntos fuera de los límites aunque el proceso puede estar bajo control. Los promedios semanales iniciales serán en general más grandes que el UCL caracterizando un proceso inestable y los promedios de las últimas semanas serán más pequeños que el UCL que según la prueba 4 en la sección A1.2 también caracteriza un proceso inestable. Aunque el proceso en la figura A1.9a y A1.9b se detecta como inestable por ambas técnicas (*SPC* y *SPC_{log}*) como se apuntó en la sección A1.3.2 puede no ser el caso para un proceso estable.

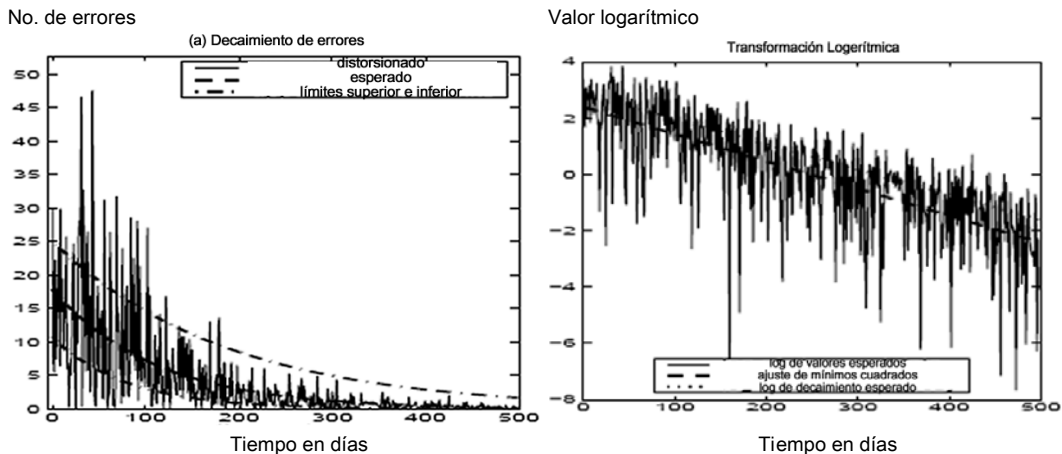


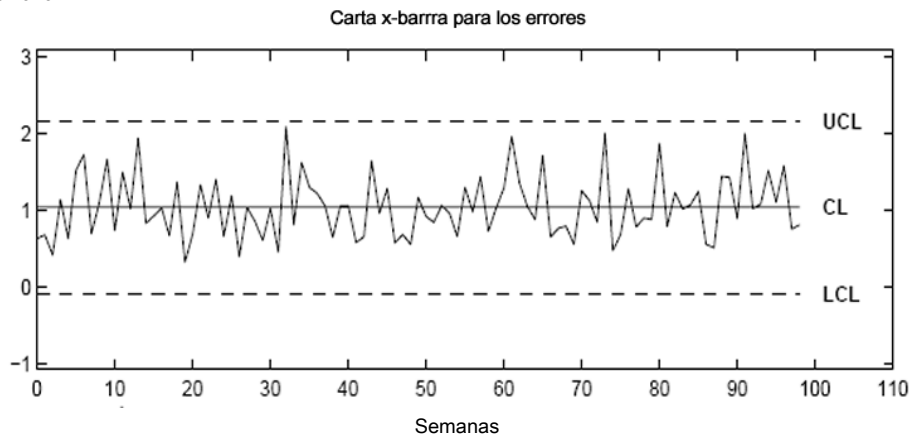
Figura A1.10: Simulación del Escenario II: (a) los datos generados con las expectativas del encargado representadas por el decaimiento esperado con límites superior e inferior, (b) la transformación logarítmica correspondiente con el ajuste de mínimos cuadrados y el decaimiento lineal esperado.

A5.2 Escenario II

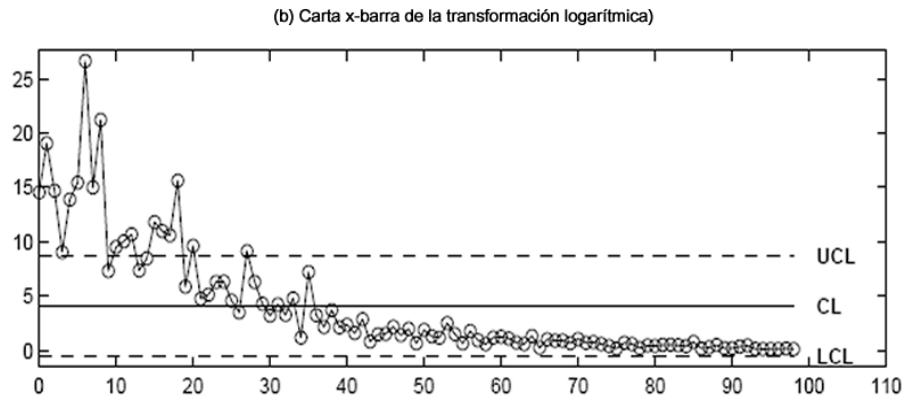
[Escribir texto]

Para la simulación del escenario II, el comportamiento esperado es generado por $r(t) = r_0 e^{-\lambda t}$ y los límites de especificación superior e inferior (USL y LSL) son definidos respectivamente por $e_v^{usp}(t) = (1 + \Delta) * \log(r(t))$ y $e_v^{lsw}(t) = (1 - \Delta) * \log(r(t))$ donde $\delta = 0.1$. Estos resultados se observan en la figura A1.10a mientras que la figura A1.10b demuestra el logaritmo de los puntos de los datos, el ajuste de mínimos cuadrados, y el logaritmo del decaimiento esperado.

Media semanal



Media semanal



[Escribir texto]

Semanas

Figura A1.11: Resultados para simulación del escenario II. (a) La carta X-barra para la distancia euclidiana entre los Logaritmos de los errores y el ajuste de mínimos cuadrados. (b) la carta X-barra para el número real de errores.

De la carta de la X-barra en la figura A1.11a podemos ver que el proceso es estable. Note que el uso de las técnicas estándares del SPC habría identificado el proceso tan inestable como puede ser observado de la figura A1.11b. Como el proceso está bajo control, la capacidad debe ser analizada. Usando el método en la sección A1.3.3, calculábamos $TCL = 1.049$, $USL = 2.268$, y $LSL = -0.357$ según lo demostrado en el histograma de capacidad en la figura A1.12a. El valor de σ para este escenario fue 0.337 . Ya que los dos límites especificados son válidos, el primer paso para verificar la capacidad es verificar si el $ST > 6$, que se observa que es verdad en este caso: $ST = \frac{USL - LSL}{\sigma} = 7.79$.

El paso siguiente es verificar si la distancia a la especificación más cercana (DNS) es mayor de tres. Para el escenario II, $Z_u = \frac{USL - \bar{X}}{\sigma} = 3.62$ y $Z_l = \frac{\bar{X} - LSL}{\sigma} = 4.17$. Por lo tanto, $DNS = \min\{Z_u, Z_l\} = 3.62 > 3$ dando por resultado un proceso capaz según lo observado de la figura A1.12b.

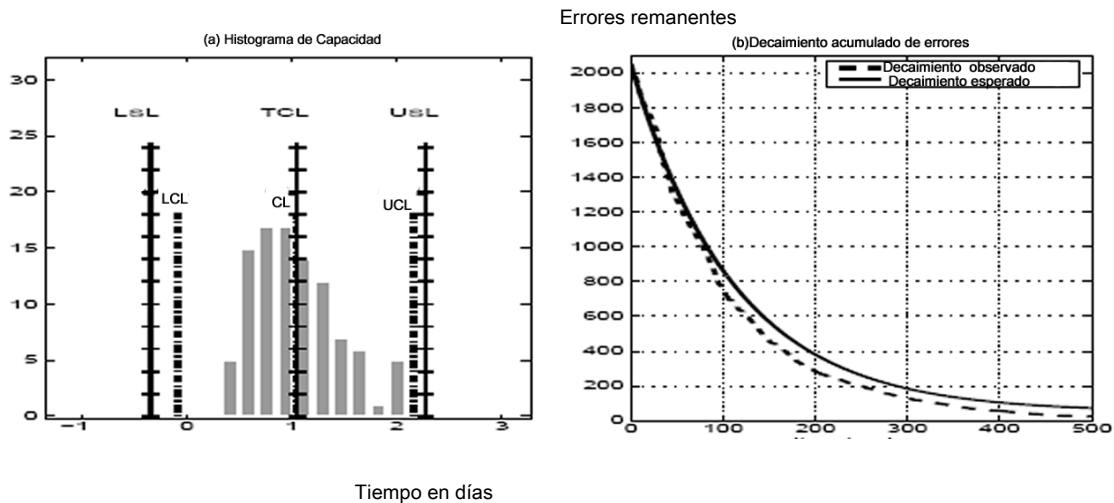


Figura A1.12: Simulación escenario II: (a) Histograma de Capacidad para los datos de la Figura A1.11 y (b) el decaimiento acumulado de errores confirma un proceso capaz

A1.6. RESUMEN Y DISCUSIÓN

Los factores que caracterizan un proceso y presentan una forma exponencial permiten la aplicación del SPC para monitorear y para controlar tales procesos. El STP es caracterizado por varios factores de calidad que exhiben un comportamiento exponencial en el tiempo. Los ejemplos de tales factores incluyen cobertura del código, el número de errores restantes, y la confiabilidad del producto bajo prueba.

[Escribir texto]

Una transformación logarítmica nombrada SPC_{log} se presenta en este artículo. Esta transformación permite el control de procesos con tales factores y mejora la aplicabilidad del SPC al control estadístico del STP.

El análisis de la aplicabilidad de SPC_{log} presentada aquí se basa en las corridas de simulación para dos escenarios. Los resultados obtenidos usando nuestro análisis demuestran la aplicabilidad del enfoque propuesto, y su corrección, cuando se aplican al STP. Los resultados de un tercer escenario y de un estudio de caso que usaban datos de un proyecto industrial también condujeron a la misma conclusión pero una descripción del mismo está más allá del alcance de este artículo. El análisis del enfoque fue basado, sin embargo, solamente en el comportamiento del número de los errores encontrados por unidad del tiempo, la misma transformación se aplica también a los otros factores de proceso mencionados arriba. Además, creemos que no hay restricción en el uso de SPC_{log} a ningún proceso que presenta un comportamiento exponencial.

[Escribir texto]

**ANEXO 2 GUÍA METODOLÓGICA DEL PROCESO DE
PRUEBAS PARA LA REVISIÓN DE SISTEMAS DE
SOFTWARE. Versión 0.2, 04/06/2008.**

[Escribir texto]

Control de Versiones

Nombre del Archivo	Versión	Fecha	Autor	Comentarios
GM_10_Pruebas_0.1	0.1	22/04/2005	AR	Versión Inicial del Documento
GM_10_Pruebas_0.2	0.2	04/06/2008	AR	Ultima Versión

CONTENIDO DEL DOCUMENTO

A2.1	ANTECEDENTES	97
A2.1.1	HISTORIA DEL DEPARTAMENTO DE PRUEBAS MODULARES A LOS SISTEMAS DE INFORMACIÓN.....	97
A2.1.2	ESTADO ACTUAL DE LAS ACTIVIDADES QUE SE REALIZAN EN EL DEPARTAMENTO DE PRUEBAS MODULARES A SISTEMAS DE INFORMACIÓN DE LA DIRECCIÓN DE DESARROLLO DE SISTEMAS DE INFORMACIÓN DE LA DGAI (INEGI).....	98
A2.2	INTRODUCCIÓN.	99
A2.2.1	OBJETIVOS	99
A2.3.	CONCEPTOS BÁSICOS Y DEFINICIONES.	100
A2.3.1	OBJETIVO DE LAS PRUEBAS.....	104
A2.3.2	REGLAS IMPORTANTES EN EL PROCESO DE PRUEBAS.	105
A2.3.3	¿POR QUÉ ES IMPORTANTE PROBAR?.....	106
A2.3.4	¿QUIÉN DEBE DOCUMENTAR Y EJECUTAR LAS PRUEBAS?.....	106
A2.3.5	CONSIDERACIONES DEL PROCESO DE PRUEBAS.	107
A2.3.5.1	PRUEBAS UNITARIAS (PU).....	107
A2.3.5.2	PRUEBAS MODULARES (PM).....	108
A2.3.5.3	PRUEBAS DE INTEGRACIÓN (PI)	108
A2.3.5.4	PRUEBAS DE SISTEMA (PS)	108
A2.3.5.5	PRUEBAS DE ACEPTACIÓN (PA).....	108
A2.4.	LA FASE DE PRUEBAS EN EL SOFTWARE: ESTADO DEL ARTE.	109
A2.4.1	IMPACTO DE LOS DEFECTOS DEL SOFTWARE.	110
A2.4.2	EL PROCESO DE INSPECCIÓN.....	111
A2.4.3	MÉTRICAS DE INSPECCIÓN.....	116
A2.4.4	RECOMENDACIONES CON RESPECTO AL EQUIPO DE INSPECCIÓN.....	116

[Escribir texto]

A2.4.5 TIPOS DE INSPECCIÓN DE SOFTWARE.....	116
A2.4.5.1 INSPECCIÓN INFORMAL.....	116
A2.4.5.2 INSPECCIÓN FORMAL.....	117
A2.4.5.3 MÉTODOS COMUNES DE REVISIÓN.....	119
A2.4.5.3.1 INSPECCIÓN.....	119
A2.4.5.3.2 REVISIÓN EN EQUIPO.....	120
A2.4.5.3.3 WALKTHROUGH.....	121
A2.4.5.3.4 REVISIÓN EN PAR.....	122
A2.4.5.3.5 VERIFICACIÓN DETALLADA DE ESCRITORIO.....	123
A2.4.5.3.6 PASSAROUND.....	123
A2.4.5.3.7 REVISIÓN PERSONAL.....	124
A2.5. ¿CÓMO SELECCIONAR EL MÉTODO MÁS APROPIADO?: CASO INEGI.....	124
A2.5.1 ESTADO ACTUAL: EJEMPLOS DE ESCENARIOS.....	125
A2.6. ¿QUIÉN DEBE Y QUÉ DEBE REVISAR?.....	130
A2.7. AUTOMATIZACIÓN DE PRUEBAS.....	131
A2.8. CONCLUSIONES.....	131
A2.9 GLOSARIO.....	¡ERROR! MARCADOR NO DEFINIDO.
A2.10 BIBLIOGRAFÍA.....	132
A2.11 ANEXOS DE LA GUÍA METODOLÓGICA.....	133

A2.1 Antecedentes.

A2.1.1 Historia del Departamento de Pruebas Modulares a los Sistemas de Información.

Desde Julio de 1986 se creó un grupo de pruebas para atender actividades relacionadas con la revisión de los sistemas, desde entonces se ha participado en diversos proyectos censales y algunos de encuestas en el INEGI, realizando estas actividades, entre las que se encontraban:

- Recibir el Requerimiento del sistema a desarrollar.
- Analizar el requerimiento del sistema.
- Asistir a las reuniones con el usuario para resolver dudas conceptuales del sistema solicitado.
- Servir de enlace entre el usuario y los desarrolladores para aclaración de conceptos.
- Generar, aplicar y revisar pruebas exhaustivas a cada uno de los Módulos o componentes del sistema, para garantizar su correcto funcionamiento.
- Generar, aplicar y revisar pruebas de Integración para verificar el correcto funcionamiento de los enlaces entre los módulos que conforman el sistema, así como el correcto funcionamiento del medio ambiente de operación del sistema.
- Aplicar y revisar pruebas al sistema con archivos que contienen información Real, realizándose éstas en un ambiente de producción para garantizar el correcto funcionamiento del sistema y, al mismo tiempo, verificar el comportamiento de la información durante su paso por el sistema.
- Aplicar nuevamente las pruebas necesarias al sistema cada vez que fuera modificada alguna parte de su programación, para garantizar su correcto funcionamiento.
- Llevar un registro y control de los cambios y acuerdos, durante el desarrollo y producción del sistema, para así entregar un requerimiento actualizado al finalizar la producción.

Los resultados de estas actividades se veían reflejados en Reportes de revisión.

[Escribir texto]

A2.1.2 Estado actual de las actividades que se realizan en el Departamento de Pruebas Modulares a Sistemas de Información de la Dirección de Desarrollo de Sistemas de Información de la DGAI (INEGI).

Actualmente se reciben sistemas desarrollados para su revisión en diferentes circunstancias, a las que llamaremos escenarios:

- Escenario 1: Sólo con manual de operación o requerimiento,
- Escenario 1A: Con manual de operación o requerimiento y servidor de pruebas actualizado.

- Escenario 2: Con funcionalidad y/o casos de uso
- Escenario 3: Con funcionalidad y/o casos de uso y casos prueba acompañados con checklist propuesto por el desarrollador del sistema
- Escenario 4 (deseado): Con manual de operación o requerimiento, funcionalidad y/o casos de uso y casos prueba acompañados con checklist, servidor de pruebas actualizado.

Asimismo los resultados de dichas revisiones y pruebas se plasman en un reporte de revisión con imágenes tomadas como evidencias y/o en un checklist de casos prueba(ver anexo).

A2.2 Introducción.

La ejecución de la fase de pruebas es uno de los procesos finales del desarrollo de software[27], sin embargo los probadores deben estar involucrados en todas las Fases del Desarrollo, desde el Análisis pasando por el Diseño y hasta su Liberación, adicionalmente en ocasiones cuando los proyectos están contra el tiempo, los procesos de pruebas se llevan a cabo de manera deficiente, esto es un error muy grande porque esta fase es esencial para entregar un producto de software de calidad, que cumpla los requerimientos de los clientes o usuarios.

Cuando no se llevan a cabo las pruebas de manera correcta, los defectos que se han pasado por alto, las fallas ocurren, originadas por estos defectos, durante la operación del producto, y por consiguiente se tiene un alto costo de mantenimiento y la insatisfacción del cliente o usuario.

Con una revisión formal, una persona diferente al autor del producto de software, puede identificar los defectos que se han por alto antes de ser liberado.

Existen distintos métodos de revisión los cuales se irán mencionando en este documento para darle formalidad y estructura a este **Proceso de Pruebas**.

A2.2.1 Objetivos

Esta Guía Metodológica del Proceso de Pruebas tiene por objetivos:

- Ayudar al probador de productos de desarrollo a identificar qué método de revisión es el adecuado para atender la revisión y pruebas de los sistemas que se reciben en el Departamento de Pruebas modulares a Sistemas de Información, promoviendo que se generen productos de software con calidad.
- Establecer las bases metodológicas para realizar un **Proceso de Pruebas** que permita estandarizar las actividades de Pruebas en la Dirección de Desarrollo de Sistemas de Información de la DGAI, y como resultado en un futuro tener conteos de defectos de sistemas desarrollados y poder aplicar técnicas estadísticas para obtener indicadores de calidad.

[Escribir texto]

A2.2.2 Alcance

Esta **Guía Metodológica del Proceso de Pruebas** se aplica para todos los Sistemas (productos de software desarrollados) que se reciban en el Departamento de Pruebas Modulares en la Dirección de Desarrollo de Sistemas de Información (DDSI) de la Dirección General de Innovación y Tecnologías de Información (DGAI). Las pruebas que se realizan se enfocan únicamente a la funcionalidad visible de los productos del sistema de software desarrollado (**Pruebas Caja Negra**), no a la revisión de líneas de código (**Pruebas Caja Blanca**). Sin embargo ésta última técnica de revisión también se comenta en la siguiente sección.

A2.3. Conceptos Básicos y definiciones.

Existen conceptos básicos que deben ser entendidos en el Proceso de Pruebas, es esencial que encargados de proyectos, desarrolladores, probadores y usuarios comprendan la diferencia de éstos:

Fase.- Conjunto de actividades relacionadas con un objetivo en el desarrollo del proyecto.

Inspección ó Revisión.- Es el proceso mediante el cual una persona diferente al autor de un producto de software, lo examina para identificar defectos y poderlo mejorar en su siguiente ciclo o iteración.

Iteración.- Relacionada con el ciclo que se cuenta entre una nueva versión del sistema y los defectos encontrados durante la revisión de la misma.

Proceso.- Marco de trabajo de las tareas que se requieren para construir software de alta calidad y mediante el proceso de pruebas garantizar su calidad.

Producto de Software.- Es el resultado de un proceso de desarrollo de software.

Especificaciones del Requerimiento.- Reflejan el fin general del Sistema así como de cada una de sus componentes y son definidos por el cliente o usuario del Sistema.

Cliente o usuario del Sistema.- Es la persona que define conceptual y teóricamente, mediante las especificaciones del requerimiento, el Sistema a desarrollar, además de revisar y liberar todas y cada una de sus componentes así como el Sistema completo. Usualmente es quien se encargará de usarlo o supervisarlo durante su producción.

[Escribir texto]

Verificar.- Es el proceso de evaluación del software para determinar si el producto cumple satisfactoriamente cada una de las fases de desarrollo. En el marco de este **Proceso** esta actividad deberá realizarse por el desarrollador.

Validar.- Es el proceso de evaluación del software durante o al final del proceso de desarrollo para determinar si este satisface los requerimientos especificados por el cliente. La validación se realiza con algunas revisiones pero siempre es completada con las pruebas. En el marco de este **Proceso** esta actividad se realizará por el grupo de probadores.

Especificación del Procedimiento de la prueba. Es un documento el cual describe la secuencia de acciones para la ejecución de las pruebas, también se puede definir la métrica de evaluación de las pruebas.

Producto de la Prueba.- Es un objeto de software, ya sea un programa, un módulo o una aplicación completa, el cual es causa de una prueba (o varias).

Caso de Prueba.- Es la especificación (usualmente formal) de un conjunto de entradas de pruebas, condiciones de ejecución y resultados esperados, identificados con el propósito de hacer una evaluación de algún aspecto en particular del Producto de Prueba.

Prueba.- Es un conjunto de dos o más casos prueba y/o procedimientos de pruebas.

Testing.- Es el proceso de encontrar defectos en relación a un conjunto de especificaciones o criterios predeterminados, usualmente las especificaciones del requerimiento. El propósito de hacer pruebas es demostrar que un sistema, software o configuración de software no está haciendo lo que debería hacer.

Tester.- Probador de sistemas.

Plan de Prueba.- Es un documento el cual describe el alcance, la estimación de recursos y la planeación en tiempo de las pruebas y sus diversas actividades. En este documento se identifican los objetos a probar, las partes del sistema a ser probados, las tareas sobre las pruebas, quién debe realizar las diversas pruebas y actividades, describe cualquier riesgo y el plan de contingencia a ejecutarse en caso de ser necesario. (Describe estas actividades y su duración). Actualmente no se ha podido hacer este Plan de Prueba debido a que ya no se participa en las actividades de Planeación y Programación como en el pasado.

Especificación del diseño de una prueba.- Es un documento que especifica los detalles que se deben incluir para las pruebas de software, de manera general y específica para los módulos, incluyendo otros objetos de prueba asociados.

[Escribir texto]

Especificación de un caso prueba.- Es un documento que especifica las entradas, resultados esperados y un conjunto de condiciones de ejecución para cada objeto de la prueba.

Objeto de la prueba.- Puede ser un componente, pantalla, botón, etc., dependiendo del sistema que se esté revisando.

Reporte de Revisión del Sistema (Reporte de Problemas). Es un documento donde se registra cualquier evento que ocurre durante las pruebas, el cual requiere investigación (incidencias de las pruebas). Sólo usado bajo solicitud expresa del desarrollo de sistema.

Checklist.- Lista de revisiones propuesta al tester por el autor o encargado del sistema bajo pruebas.

Nivel de Prueba.- Se refiere a las pruebas que se solicite por el usuario o el Líder de Proyecto de desarrollado: Unitarias, Modular, de Integración, de Sistema y de Aceptación.

Auditoría.- Hacer énfasis en la revisión, evaluación y elaboración de un informe para el Líder de Proyecto.

Gestión de Proyectos.- Es la disciplina de organizar y administrar recursos de manera tal que se pueda culminar todo el trabajo requerido en el proyecto dentro del alcance, el tiempo, y coste definidos.

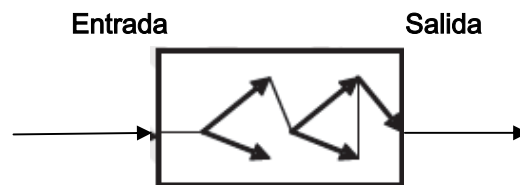
Proyecto.- es un esfuerzo temporal, único y progresivo, emprendido para crear un producto o un servicio también único.

Funciones de información de la Gestión.- La gestión de proyectos, muchas veces, es responsabilidad de un solo individuo. Este individuo raramente participa de manera directa en las actividades que producen el resultado final. En vez de eso se esfuerza por mantener el progreso y la interacción mutua productiva de las varias partes de manera que el riesgo general de fracasar se disminuya. La experiencia del Departamento de Pruebas ha versado en este sentido en algunos proyectos, con la finalidad de obtener insumos suficientes y oportunos para realizar adecuadamente las revisiones.

Existen 2 tipos de prueba relevantes al marco del presente trabajo de investigación que son:

[Escribir texto]

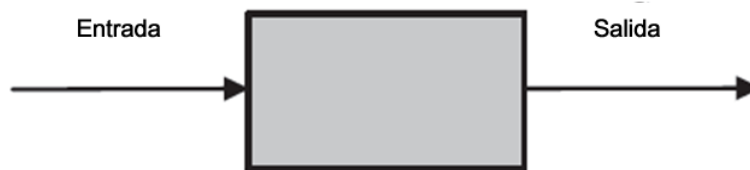
1. **Pruebas de Caja Blanca.** Es una técnica de prueba que se basa en el conocimiento de la estructura interna del código y las rutas ó caminos lógicos de un programa ó componente. Estas pruebas consisten en verificar que el diseño del software es válido y que fue construido de acuerdo a un diseño específico. Estas pruebas rastrean la administración de la configuración, los controles y las especificaciones de interfaces internas. Todas estas especificaciones han sido identificadas como una parte integral del proceso de control de configuración. En la Figura 2-1 se puede apreciar el funcionamiento de una prueba de caja blanca.



Fuente: http://info.pue.udlap.mx/~pgomez/ing_sw/

Figura A2.3.1 Prueba de Caja Blanca

2. **Pruebas de Caja Negra.** Es una técnica que se enfoca en pruebas de funcionalidad de programas, componentes ó aplicaciones, se basa en especificaciones externas sin conocer cómo el sistema está construido. Con estas pruebas se valida si el software está de acuerdo a los requerimientos sin recordar qué rutas de ejecución se toman para conocer cada requerimiento.



Fuente: http://Info.pue.udlap.mx~pgomez/ling_sw/

Figura A2.3.2 Prueba de Caja Negra

Como se comentó en los objetivos, en el Departamento de Pruebas Modulares a los Sistemas de Información solo se realizan **Pruebas de caja Negra.**

La fase de pruebas se forma de una serie de actividades continuas de validación de un producto, desde que éste es definido y verificado por el autor del desarrollo hasta su

[Escribir texto]

validación final en un sistema de configuración instalado. Esta fase no necesariamente se lleva a cabo en una computadora.

Pruebas de Integración (PI). Las pruebas de integración consisten en probar la integridad de los módulos que han sido exitosos en las pruebas modulares, son realizadas por el grupo de pruebas apoyados por el desarrollador. Inician una vez que los módulos han pasado exitosamente las pruebas unitarias. Además con el esquema del diseño del sistema, los módulos probados se integran para comprobar sus interfaces.

Pruebas de Sistema (PS). Una prueba de sistema se lleva a cabo al probar el sistema completo y las interfaces con otras aplicaciones para validar los requerimientos funcionales y no funcionales, son realizadas por el grupo de pruebas junto con el Líder de Proyecto. Inician cuando las pruebas de integración han sido completadas exitosamente.

Pruebas de Aceptación (PA).- Una prueba de aceptación se hace efectiva al momento que el probador y después el usuario comprueban que el sistema es adecuado, apoyándose en las especificaciones del Requerimiento, además de evaluar la manera en que el sistema se integra con el manual de proceso de negocios (usualmente derivado del requerimiento de usuario), estas pruebas son realizadas por el usuario y dirigidas por el Líder de Proyecto y el grupo de pruebas. Inician cuando las pruebas de sistema han sido completadas exitosamente.

A2.3.1 Objetivo de las Pruebas.

El objetivo principal es descubrir o encontrar defectos que pueden producir fallas en el producto que se está probando. La actitud que se debe tomar ante las pruebas debe ser la de “Destruir el sistema”, la finalidad es encontrar defectos y comprobar que el sistema no trabaja correctamente bajo cierto tipo de condiciones. Durante el proceso de pruebas se descubren fallas y debilidades en un producto de software. La postura que se debe tener durante el proceso de las pruebas debe ser contemplar la perspectiva del cliente o usuario del sistema, con el objeto de no perder algún detalle significativo para el proceso de pruebas.

A2.3.2 Reglas Importantes en el Proceso de Pruebas.

1. Las pruebas deben ser planeadas, priorizadas, programadas y controladas si se desea tener un proceso de pruebas efectivo.
2. La ejecución de pruebas comienza con las **Pruebas de Caja Blanca** a unidades de código (actualmente realizadas por el desarrollador) y va progresando en base a una serie de **Pruebas de Caja Negra** hasta llegar a las **pruebas de Integración** y las **pruebas de aceptación del sistema** (de acuerdo al plan previamente definido).
3. Todos los componentes de una prueba deben estar definidos, revisados, actualizados y controlados en el plan de pruebas previamente definido, asimismo si hay cambios del cliente o usuario se deben actualizar las especificaciones del Requerimiento.
4. Las pruebas deben diseñarse para probar que un sistema NO FUNCIONA, no lo contrario.
5. Todo Sistema de desarrollo de software debe ser probado, usando las especificaciones de del requerimiento controladas y actualizadas.
6. Cuando el Software es desarrollado por diferentes empresas o personas, solo una debe ser responsable de la integración total del sistema.
7. Las herramientas para probar, de requerirse alguna, solo deben ser adquiridas o construidas, si proveen una habilidad que no se posee en el equipo o institución. Estas herramientas deben apoyar en ofrecer más control, reproducir los casos de prueba o errores y brindar más eficiencia en los resultados de las pruebas (Tiempo).
8. Las Pruebas de Software no terminan hasta que el sistema se haya "Probado" en base a su capacidad de respuesta, y otros factores críticos de funcionamiento óptimo (seguridad, factores operacionales, etc.).
9. Un programa completo de pruebas reduce el plan de riesgos del proyecto.
10. Todas las versiones y liberaciones que se realicen después de la primera versión del Sistema, deben documentarse al final en Plan de Pruebas.
- 11.- No deben incluirse en el equipo de probadores personas que desconozcan el Proceso de Pruebas y/o que no cuenten con el perfil de probadores y experiencia, sobre todo en épocas críticas de revisión.

[Escribir texto]

A2.3.3 ¿Por qué es Importante Probar?

- La calidad que se obtenga del proceso de pruebas determina el éxito de todo el esfuerzo de la etapa de pruebas y del producto de software final.
- Se prevén fallas, errores o defectos, cuando se utiliza oportunamente y en cada fase dentro del ciclo de vida, utilizando técnicas de pruebas.
- Para encontrar posibles errores ó debilidades en el producto y solucionarlas a tiempo.
- Debe existir un responsable de la planeación, diseño, implementación y documentación de todo el proceso de pruebas.
- El **Proceso de Pruebas** es una disciplina profesional que requiere entrenamiento y ciertas habilidades y aptitudes.
- Para cultivar en el equipo una actitud positiva hacia las pruebas. Así como saber gestionar los insumos para realizar una revisión efectiva y oportuna.

A2.3.4 ¿Quién debe Documentar y Ejecutar las Pruebas?

El desarrollador.

Ventajas:

- Conoce la estructura interna del código (programa o conjunto de estos).
- Ya tiene una idea de cuáles son los puntos necesarios a probar de manera unitaria.

Desventajas:

- Pero, por lo anterior, se puede caer en suposiciones, que pueden evitar encontrar a tiempo algún error.
- De igual manera se pueden subestimar o menospreciar ciertos puntos del programa o especificaciones.
- Al ser familiar con el código ya que es parte de su creación, es difícil tener en la mente la idea de criticarlo.

Un especialista en Pruebas.

Ventajas:

[Escribir texto]

- Su actitud es de encontrar fallas, errores o defectos.
- Aunque no se posee el conocimiento de qué partes del código son más complejas y cuáles no, puede gestionarlo adecuadamente con el desarrollador autor del producto de software.
- Se tienen menos suposiciones.

Recomendaciones:

El desarrollador: Documentar sus objetos de prueba (principalmente su funcionalidad y /o casos de uso).

El especialista en Pruebas: Enfocarse en la generación y ejecución de casos prueba. Así como verificar la corrección de fallas y dar el seguimiento mediante reportes dirigidos al líder del proyecto.

A2.3.5 Consideraciones del Proceso de Pruebas.

- Adicionar valor al producto final ya sea por medio de garantizar su calidad ó incrementar su reutilización.
- Determinar todas las entradas y salidas para los diferentes niveles de prueba (criterios a probar en los casos de prueba).
- Tener un plan para cada nivel de prueba, adecuándose a la realidad de los insumos recibidos para su revisión y contingencias (tiempos).
- Cubrir las tres capas de las pruebas:

1. Del Código (Pruebas Unitarias, realizadas por el desarrollador).
2. Modular (Pruebas de Integración)
3. Funcional (Pruebas del Sistema)

- Cubrir y pasar satisfactoriamente las tres capas son indispensables para seguir con el plan de pruebas.
- Cubrir la etapa del diseño con las entradas para la generación de los casos de pruebas.

A2.3.5.1 Pruebas Unitarias (PU)

[Escribir texto]

Una prueba de unidad es probar programas individuales o componentes para validar que su lógica trabaja correctamente de acuerdo a las especificaciones, son realizadas por cada desarrollador quien ha creado el componente en su entorno*.

Las pruebas unitarias inician una vez que el componente ha sido desarrollado completamente.

A2.3.5.2 Pruebas Modulares (PM)

Los módulos que componen el Sistema, pueden integrar varios componentes y hay que probar su correcto funcionamiento.

A2.3.5.3 Pruebas de Integración (PI)

Las pruebas de integración consisten en probar la integridad de los módulos que han sido exitosos en las pruebas modulares, son realizadas por el grupo de pruebas apoyados por el desarrollador. Inician una vez que los módulos han pasado exitosamente las pruebas unitarias. Además con el esquema del diseño de los sistemas, los módulos probados se integran para comprobar sus interfaces.

A2.3.5.4 Pruebas de Sistema (PS)

Una prueba de sistema se lleva a cabo al probar el sistema completo y las interfaces con otras aplicaciones para validar los requerimientos funcionales y no funcionales, son realizadas por el grupo de pruebas junto con el Líder de Proyecto. Inician cuando las pruebas de integración han sido completadas exitosamente.

A2.3.5.5 Pruebas de Aceptación (PA)

Una prueba de aceptación se hace efectiva al momento que el probador y después el usuario comprueban que el sistema es adecuado, apoyándose en las especificaciones del Requerimiento, además de evaluar la manera en que el sistema se integra con el manual de proceso de negocios (usualmente derivado del requerimiento de usuario), estas pruebas son realizadas por el usuario y dirigidas por el Líder de Proyecto y el grupo de pruebas. Inician cuando las pruebas de sistema han sido completadas exitosamente.

A2.4. La Fase de Pruebas en el Software: Estado del arte.

Las inspecciones de software surgen a partir de la necesidad de producir software de alta calidad. Algunos grupos de desarrollo creen que la calidad del software es algo en lo que deben preocuparse una vez que se ha generado el código. Esto es un error.

La garantía de la calidad del software es una actividad de protección que se aplica a lo largo de todo el proceso de ingeniería de software. La SQA (Software Quality Assurance) engloba:

- Un enfoque de gestión de calidad.
- Tecnología de Ingeniería de Software efectiva (métodos y herramientas).
- Revisiones y técnicas formales que se aplican durante el proceso del software.
- Una estrategia de prueba multiescalada.
- Un control de la documentación del software y de los cambios realizados.
- Un procedimiento que asegure un ajuste a los estándares de desarrollo de software.
- Mecanismos de medición y de generación de informes.

El control de la calidad es una serie de revisiones y pruebas utilizadas a lo largo del ciclo de desarrollo para asegurar que cada producto cumple con los requisitos que le han sido asignados.

La garantía de calidad o aseguramiento de la calidad consiste en la auditoría y las funciones de información de la gestión. El objetivo de la garantía de la 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 producto está cumpliendo sus objetivos. Es de esperar, que si los datos proporcionados mediante la garantía de la calidad identifican problemas, la gestión afronte los problemas y aplique los recursos necesarios para resolverlos.

En este marco podemos ver a las inspecciones como una implementación de las revisiones formales de software las cuales representan un filtro para el proceso de ingeniería de software, éstas se aplican en varios momentos del desarrollo y sirven para detectar defectos que pueden así ser eliminados. Freeman y Weinberg argumentan de la siguiente forma la necesidad de revisiones:

[Escribir texto]

El trabajo técnico necesita ser revisado por la misma razón que los lápices necesitan gomas: errar es humano. La segunda razón por la que necesitamos revisiones técnicas es que, aunque la gente es buena descubriendo algunos de sus propios errores, algunas clases de errores se le pasan más fácilmente al que los origina que a otras personas.

Una revisión es una forma de aprovechar la diversidad de un grupo de personas para:

1. Señalar la necesidad de mejoras en el producto de una sola persona o de un equipo.
2. Confirmar las partes del producto en las que no es necesaria o no es deseable una mejora.
3. Conseguir un trabajo de mayor calidad maximizando los criterios de cómo hacer las cosas correctamente (Correctitud) y de hacerlas de manera completa (Complejidad) principalmente.

Existen diferentes tipos de revisiones que se pueden llevar adelante como parte de la ingeniería del software. Cada una tiene su lugar. Una reunión informal durante el almuerzo o en un café es una forma de revisión, si se discuten problemas técnicos. Una presentación formal de un diseño de software a una audiencia de clientes, ejecutivos y personal técnico es una forma de revisión. Sin embargo en este trabajo nos concentraremos en una revisión técnica formal, que llamaremos Inspección formal de Software.

La garantía de calidad del software comprende una gran variedad de tareas, asociadas con dos elementos diferentes: los ingenieros de software, que realizan trabajo técnico, y un grupo SQA, que tiene la responsabilidad de la planificación de garantía de calidad.

En este marco podemos ver a las inspecciones como una implementación de las revisiones formales del software las cuales representan un filtro para el proceso de ingeniería de software, éstas se aplican en varios momentos del desarrollo y sirven para detectar defectos que pueden así ser eliminados.

A2.4.1 Impacto de los Defectos del Software.

Dentro del contexto de desarrollo de software, los términos "defecto" y "fallo" son sinónimos. Ambos implican un problema de calidad descubierto después de entregar el software a los usuarios finales.

[Escribir texto]

El objetivo primario de las revisiones técnicas formales (inspección) es encontrar errores durante el proceso para evitar que se conviertan en defectos después de la entrega del software. El beneficio obvio de estas inspecciones es el descubrimiento de errores al principio para que no se propaguen al paso siguiente del proceso de desarrollo del software (catarata de errores de Mizuno).

Una serie de estudios (TRW, Nippon Electric y Mitre Corp., entre otros) indican que las actividades del diseño introducen entre el 50% y 65% de todos los errores (y en último lugar, todos los defectos) durante el proceso de software. Sin embargo se ha demostrado que las inspecciones de software son efectivas en un 75% a la hora de detectar errores [JON86].

Con la detección y la eliminación de un gran porcentaje de errores, el proceso de inspección reduce substancialmente el costo de los pasos siguientes en las fases de desarrollo y mantenimiento.

A2.4.2 El Proceso de Inspección.

Podemos ver a las inspecciones de software como un repaso detallado y formal del trabajo en proceso. Pequeños grupos de trabajadores (4 ó 5) estudian el “producto de trabajo” independientemente y luego se reúnen para examinar el trabajo en detalle. El “producto de trabajo” representa 200 a 250 líneas de código, requerimientos, diseño y otros productos de trabajo son inspeccionados en un tamaño similar. Los productos de trabajo son considerados en progreso hasta que la inspección y las correcciones necesarias estén completas.

El tiempo de la inspección varía según cuan familiarizado esté el inspector con el material.

La inspección consiste en seis pasos [Fagan 1986]:

1. Planificación: Cuando el desarrollador completa un “producto de trabajo”, se forma un grupo de inspección y se designa un moderador. El moderador asegura que el “producto de trabajo” satisfaga el criterio de inspección. Se le asignan diferentes roles a las personas que integran el grupo de inspección, así como la planificación de tiempos y recursos necesarios.
2. Overview: Si los inspectores no están familiarizados con el desarrollo del proyecto, una vista general es necesaria en este momento. Este es un paso opcional, pero no

[Escribir texto]

menos importante ya que en esta etapa se dará al grupo de inspección un “background” y el contexto a cubrir por las inspecciones.

3. Preparación: Los inspectores se preparan individualmente para la evaluación en la reunión, estudiando los productos de trabajo y el material relacionado.

Aquí es aconsejable la utilización de listas de chequeos para ayudar a encontrar defectos comunes, y el tiempo que pueda llevar esta etapa va a depender de cuan familiarizado esté el inspector con el trabajo que debe analizar.

4. Examen: En esta etapa los inspectores se reúnen para analizar su trabajo individual en forma conjunta. El moderador deberá asegurarse que todos los inspectores están suficientemente preparados. La persona designada como lector presenta el “producto de trabajo”, interpretando o parafraseando el texto, mientras que cada participante observa en busca de defectos. Es recomendable que este examen no dure más de 2 horas ya que la atención en busca de defectos va disminuyendo con el tiempo. Al terminar con la reunión, el grupo determina si el producto es aceptado o debe ser retrabajado para una posterior inspección.

5. Retrabajo: El autor corrige todos los defectos encontrados por los inspectores.

6. Seguimiento: El moderador chequea las correcciones del autor. Si el moderador está satisfecho, la inspección está formalmente completa, y el “producto de trabajo” es puesto bajo el control de configuración.

A partir de estos seis pasos surge la necesidad de la conformación de objetivos, participantes de la inspección y con qué roles, productos de trabajo a inspeccionar y cuál deberá ser el resultado de las reuniones de inspección.

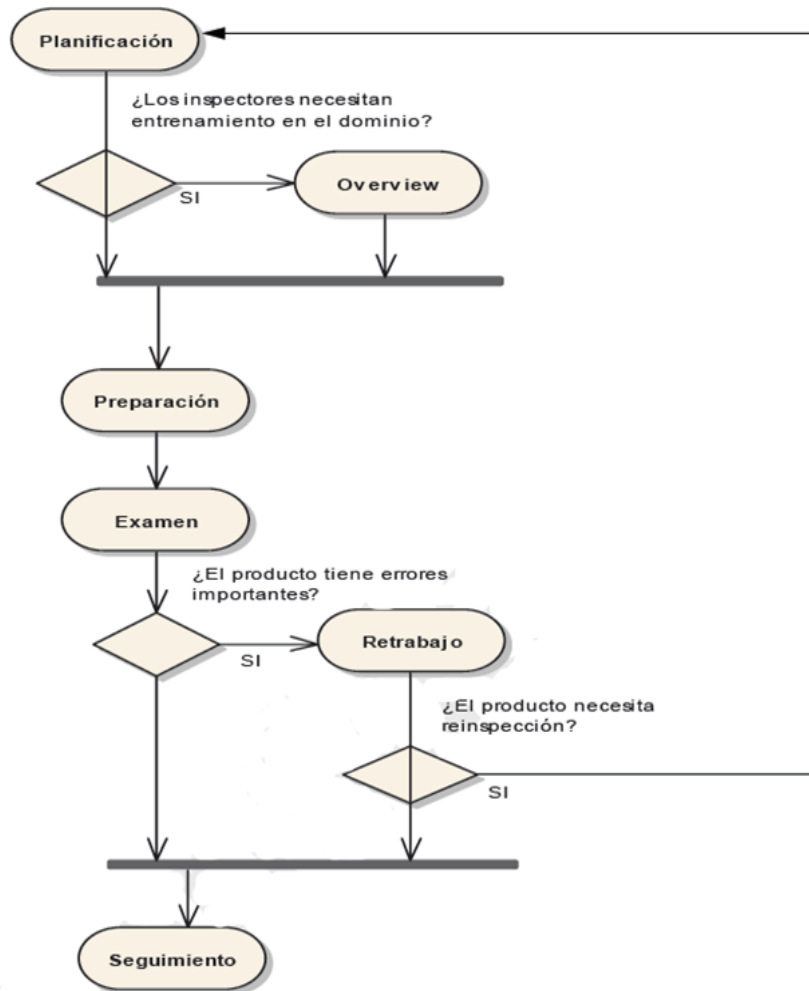
- **Objetivos:** El principal objetivo es encontrar defectos en el “producto de trabajo”, de esta manera debemos definir cuáles serán las metas a alcanzar para el cumplimiento de este objetivo. En nuestra opinión el establecimiento de estas metas están en relación directa con el tipo de proyecto, metodologías y herramientas utilizados.
- **Grupos de inspección:** Se recomienda formar grupos entre 3 y 6 personas [IEEE STD 1028-1988]. Dentro de este grupo debe incluirse al autor de los productos de trabajo.
- **Roles:** Además del autor se deberá tener en cuenta al moderador quien dirige la inspección y deberá contar con mayor experiencia que el resto, el lector quien presenta los productos de trabajo en las reuniones y el escriba quien documenta la descripción y ubicación de los defectos encontrados.

[Escribir texto]

- **Productos de trabajo:** El proceso de inspección puede ser aplicado a diferentes tipos de productos de trabajo que pueden encontrarse en un desarrollo de software incluyendo requerimientos, diseño, código, test, guías de usuario y otro tipo de documentación. El estándar de la IEEE no especifica un tamaño, pero los productos de trabajo tienen un tamaño de 10 a 20 hojas para especificación de requerimientos, 200 o 250 líneas de código. En nuestra opinión también se debe tener en cuenta la división natural que pueda tener cada uno de estos documentos, ya que toda especificación podremos dividirla en partes así como el diseño o el código.
- **Resultado de las reuniones de inspección:** Los dos resultados principales debe ser: Una lista de defectos a corregir, y un reporte de inspección que describa qué es lo que se inspeccionó, quién inspeccionó qué y el número de defectos encontrados.

Utilizando una notación UML (Lenguaje unificado de modelado, de Booch-Rumbaugh-Jacobson), se describirá gráficamente con un diagrama de actividades como se puede apreciar en la figura A2.4.1, y casos de uso del proceso de inspección como se puede apreciar en la figura A2.4.2.

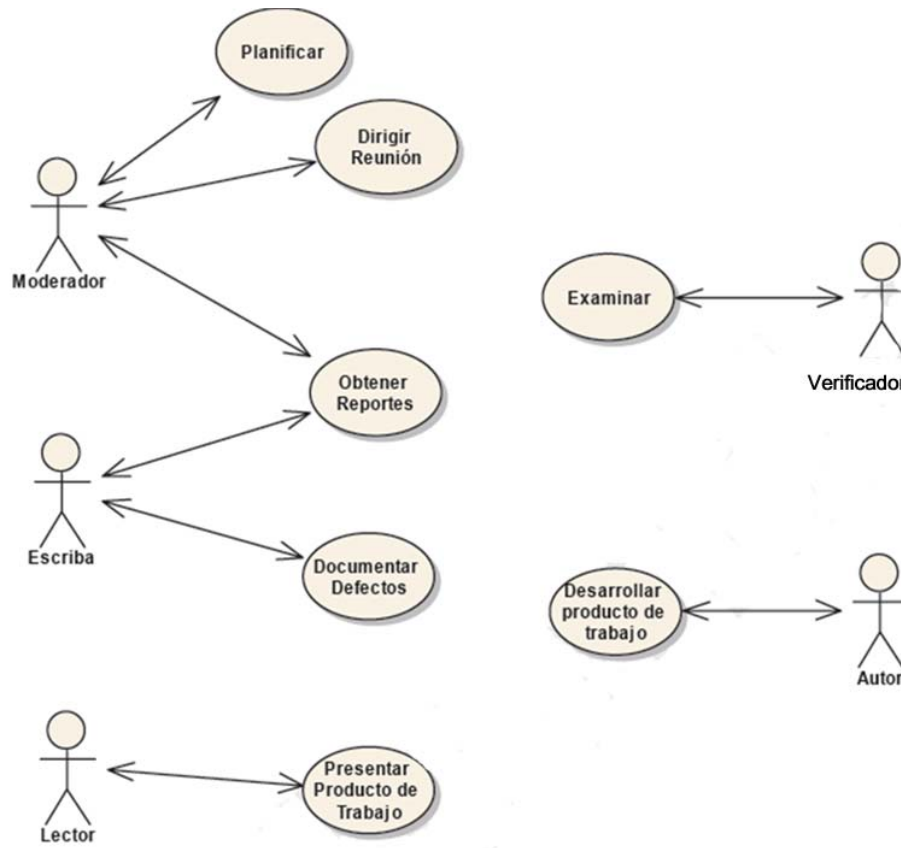
[Escribir texto]



Fuente: www.ilustrados.com/publicaciones

Figura A2.4.1 Diagrama de Actividades de una Inspección

[Escribir texto]



Fuente: www.ilustrados.com/publicaciones

Figura A2.4.2 Modelo de Caso de Uso de una Inspección

Infraestructura de soporte

Además de los actores que identificamos en el diagrama anterior, debemos tener en cuenta un nuevo actor ya que las inspecciones no ocurren espontáneamente.

Deben ser planeadas y soportadas por alguien, que tenga la responsabilidad de llevarlas adelante. Este nuevo actor es el llamado “coordinador de inspecciones de software” [Ackerman-Buchwald-Lewski]. Cuyas tareas incluyen:

- Aprender sobre inspecciones y convencer al proyecto de utilizarlas.
- Determinar en qué parte del proyecto deben ser utilizadas.
- Crear y documentar específicamente para cada proyecto los procedimientos de inspección así como los manuales de inspección.
- Organizar entrenamientos en el proceso de inspección manteniendo la documentación y actualización de dicho entrenamiento.
- Recolectar datos de inspecciones en los proyectos para una base de datos de inspecciones.

[Escribir texto]

- Analizar datos de inspecciones de distintos proyectos para hacer recomendaciones de mejoras en los procesos.

A2.4.3 Métricas de Inspección.

El proceso de inspección puede ser medido para analizar distintos aspectos del proceso (planificación, monitoreo, control, mejora, etc.) y poder maximizar su eficacia así como corregir posibles desvíos que puedan producirse durante la inspección.

En nuestra opinión las mediciones deben llevarse a cabo para poder formar una base de datos con los distintos proyectos con el fin de utilizarla a la hora de planificar nuevas inspecciones.

A2.4.4 Recomendaciones con Respecto al Equipo de Inspección.

No debemos descuidar la comunicación que debe existir entre los inspectores y el equipo de desarrollo. Debemos tener en cuenta aspectos como la forma en que se comunican los defectos que existan en el software, ya que por una reacción normal el autor del "producto de trabajo" intentará justificarlo y muchas veces esa justificación se desvía de su objetivo principal si el autor se siente "atacado" por el inspector.

Deberemos seleccionar cuidadosamente al grupo de inspección, éste deberá ser "respetado" por el equipo de desarrollo en cuanto a sus conocimientos profesionales y del proyecto ya que de no ser así esto será sin dudas una fuente de conflicto permanente.

A2.4.5 Tipos de Inspección de Software.

Personas que participan en la generación de un producto de software, liberan sus tareas con cierto número de defectos y aunque ellos mismos lo han revisado en repetidas ocasiones, pueden no darse cuenta de que han inyectado errores que pueden resultar graves en el proceso de liberación del producto. Por esta razón es conveniente buscar otro punto de vista, es decir, otra persona que a través de un proceso de inspección ayude a identificar los errores que se han introducido.

Existen dos tipos de inspección para la revisión detallada de un producto:

Inspección Informal e Inspección Formal.

A2.4.5.1 Inspección Informal.

Una inspección informal es rápida y barata, no requiere una planeación, ni tiene una infraestructura organizacional.

[Escribir texto]

En una inspección informal:

- No se cuenta con un proceso definido ni la participación de roles.
- Es usualmente más una revisión personal que planeada.
- Ejemplos de inspecciones informales: Walkthrough, Peer Deskchek y

Passaround

- No se describe a detalle cómo se lleva a cabo la inspección.

A2.4.5.2 Inspección Formal.

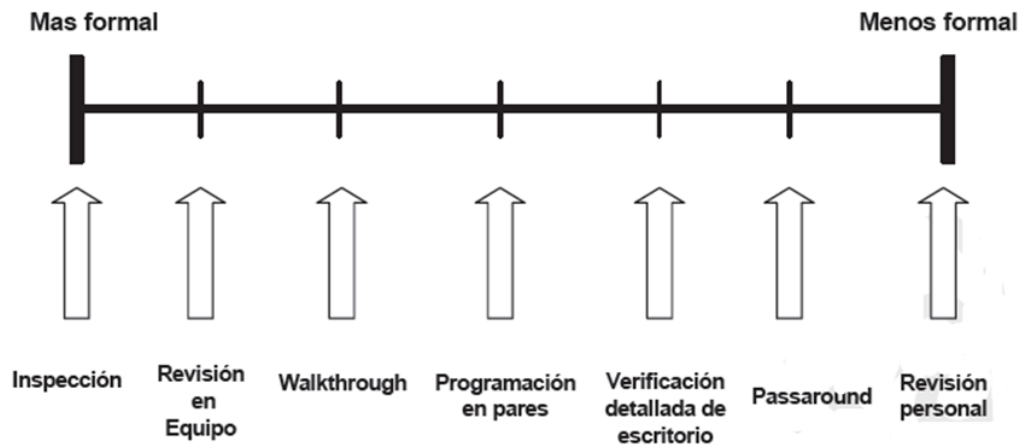
Las inspecciones formales se identifican por tener características específicas como:

- Objetivos de revisión definidos.
- Un procedimiento documentado de revisión.
- La participación de un equipo experimentado con roles definidos.
- La dirección de un moderador experimentado.
- Uso de checklist, reglas, o métodos de análisis usados para encontrar defectos.
- Reporte de resultados de parte de la gerencia sobre el estatus del producto.
- Dar seguimiento a los defectos encontrados
- Colección de datos para un control de calidad y dirección de procesos

En una revisión formal, una persona diferente al autor que está trabajando en el producto, lo examina para identificar defectos y éstos deben ser corregidos para poderlo mejorar en su siguiente ciclo.

Existen varias y distintas técnicas de revisión. En la Figura 5-1, se pueden apreciar los métodos más comunes.

[Escribir texto]



Fuente: Karl E. Wiegers, 2001

Figura A2.4.5.1 Visión de una Revisión Formal

Todas las revisiones formales, involucran alguna combinación de planeación, estudio del producto en el que se está trabajando, una junta de revisión, corrección de errores y verificación de correcciones. En la Tabla A2.4.5.1 se puede apreciar cuál de estas actividades son típicamente incluidas en cada tipo de revisión mostrado en la figura A2.4.5.1.

Tipo de Revisión	Planeación	Preparación	Reunión	Corrección	Verificación
Inspección	Sí	Sí	Sí	Sí	Sí
Equipo de Revisión	Sí	Sí	Sí	Sí	Sí
Walkthrough	Sí	No	Sí	Sí	No
Programación en par	Sí	No	Interrupción	Sí	Sí
Verificación de escritorio detallada	No	Sí	Posiblemente	Sí	No

[Escribir texto]

Revisión Personal	No	No	Sí	Sí	No
-------------------	----	----	----	----	----

Tabla A2.4.5.1 Actividad de Revisión

A2.4.5.3 Métodos Comunes de Revisión.

A2.4.5.3.1 Inspección.

Una inspección es el tipo de revisión más sistemático y riguroso que existe. La inspección sigue un proceso gradual bien definido con roles específicos que son asignados a cada integrante del equipo de trabajo. El proceso de inspección más común incluye siete etapas que son: planeación, visión general, preparación, reunión, reanudación de trabajo, dar seguimiento y análisis. Para mayor efectividad, los inspectores deben ser entrenados en el proceso de inspección y también deben tener la habilidad de poder desempeñar cualquiera de los diferentes roles. Las inspecciones cuentan con checklist de defectos comúnmente encontrados en diversos tipos de trabajo de software, reglas para construir estos productos y varias técnicas de análisis para buscar errores.

Un aspecto fundamental de la inspección es que los participantes a excepción del autor del producto, conducen la reunión, presentan el material al equipo de inspección y registran los puntos que se hayan mencionarlo en la reunión.

En la Tabla 5-2, se pueden apreciar las tareas de los responsables en dirigir una reunión de inspección.

Responsable	Actividad
Moderador	Conducir la reunión

[Escribir texto]

Lector	Presentar el material al equipo de inspección
Writer	Tomar nota de los errores encontrados

Tabla A2.4.5.2 Responsabilidades en una inspección

Los participantes se preparan para la reunión de la inspección, examinando el material por ellos mismos para entenderlo y para encontrarle detalles que pudieran estar dando problemas. Durante la reunión, el lector presenta el material a los demás inspectores, quienes hacen más comentarios e identifican posibles defectos. El lector ayuda al equipo a darle la misma interpretación a cada parte del producto para que los inspectores puedan comparar lo que ellos han comprendido, con lo que el lector ha expresado. Al final de la reunión, el equipo lleva a cabo una valoración del producto de trabajo y decide cómo verificar los cambios que el autor realizará durante la reanudación del trabajo.

Las inspecciones son más efectivas para encontrar defectos que las técnicas de revisión informal.

A2.4.5.3.2 Revisión en Equipo.

Las revisiones en equipo son un tipo de inspección más ligera, que aunque están planeados y estructurados, son menos formales y menos comprendidos. El equipo de revisión tiene varios nombres como el de “walkthrough estructurado” o simplemente “revisión”. Estas revisiones siguen varios de los pasos encontrados en una inspección. Los participantes reciben el material a revisar, con días de anticipación a la reunión para que lo puedan estudiar y analizar por ellos mismos.

El equipo reúne datos sobre el esfuerzo dedicado a la revisión y el número de defectos encontrados. Sin embargo, se simplifican o se omiten la descripción y las etapas de inspección, y algunos roles de los participantes pueden ser combinados.

El autor puede conducir al equipo de revisión, mientras que en una inspección, el autor no puede funcionar como asesor. A diferencia de la mayoría de las inspecciones, se omite el papel del lector.

En lugar de tener un lector que vaya describiendo por partes el producto, el líder de la revisión, sólo pregunta a los participantes si tienen comentarios sobre una sección o página específica del trabajo.

[Escribir texto]

Es más costoso tener un equipo de revisión que tener a un colega realizando la revisión con ayuda de un checklist detallado, la diferencia es que los participantes del equipo encontrarán otros o más errores.

Como en cualquier reunión, pueden presentarse discusiones u opiniones diferentes o bien, acuerdos que puedan ser olvidados más adelante, por lo tanto la persona que dirige la revisión, deberá registrar todo lo que se trató durante la junta. La persona que registra o escribe, captura todo lo que sucede durante la reunión pero usando formatos estándar que la organización ha definido o adoptado.

A2.4.5.3 Walkthrough.

Un Walkthrough, es una revisión informal en el cual, el autor de un producto lo describe a un grupo de compañeros y le solicita sus comentarios. En una revisión Walkthrough el autor toma absolutamente el control, pues no hay otros papeles específicos de revisión. Mientras que en una inspección se piensa cómo resolver los objetivos de la calidad del equipo, la revisión Walkthrough, sólo responde a las necesidades del autor. En la Tabla A2.4.5.3.3 se pueden apreciar algunas diferencias entre la forma de cómo se lleva a cabo una inspección, entre un Equipo de Revisión y un Walkthrough.

Característica	Inspección	Equipo de Revisión	Walkthrough
Líder	Moderador	Moderador	Autor
Presentar material	Lector	Moderador	Autor
Revisar en	Pequeñas partes	Páginas o secciones	A selección del autor
Uso de registros	Sí	Sí	Probablemente
Seguir el Procedimiento de un	Sí	Probablemente	Probablemente

[Escribir texto]

documento			
Roles definidos a los participantes	Sí	Sí	No
Uso de un checklist de defectos	Sí	Sí	No
Colección y análisis de datos	Sí	Probablemente	No
Evaluación del Producto determinado	Sí	Sí	No

Tabla A2.4.5.3.3 Diferencias entre Walkthrough y Equipo de Revisión

Una revisión típica Walkthrough, no sigue un procedimiento definido, no requiere reportar a alguna gerencia y tampoco genera métricas. Corre el riesgo de pasar por alto correcciones que podrían haber sido cambiadas pero que no lo fueron, esto puede ser consecuencia de que los expedientes fueron guardados incorrectamente o existen pocos datos sobre cómo los Walkthroughs han detectado errores en el producto.

Cuando no siguen un procedimiento bien definido, la gente que realiza este tipo de inspección lo hará de diferentes maneras. En un Walkthrough típico, el autor presenta un módulo del código o un componente del diseño a los participantes, describiendo lo que hace, cómo está estructurado, cómo realiza sus tareas y cómo es el flujo de la lógica de sus entradas y sus salidas.

A2.4.5.3.4 Revisión en Par.

En la programación en par, dos desarrolladores trabajan en el mismo producto de manera simultánea y en un mismo sitio de trabajo. Esto facilita la comunicación y da oportunidad para una revisión continua, incremental e informal de las ideas de cada persona que está participando. El hecho de que cada línea de código sea escrita por

[Escribir texto]

dos personas que conducen un solo sistema, puede ser resultado de aplicar el adagio de que “dos cabezas piensan mejor que una”. La programación en par se puede utilizar para crear cualquier producto de trabajo

De software, promueve la colaboración y una actitud positiva de cada participante para hacer rápidamente correcciones en curso, qué es y cómo si se llevara a cabo una revisión en tiempo real. La programación en par es un tipo de revisión informal porque no es estructurada y no implica ningún proceso, preparación o documentación. Carece de una perspectiva exterior para alguien que no se ha involucrado personalmente en el código o en su revisión, no incluye al autor del producto como perspectiva separada. En este caso el autor del producto es la compañía que solicitó el trabajo.

La programación en par no es específicamente una técnica de revisión, pero sí una estrategia de desarrollo de software que confía en la sinergia del enfoque de dos mentes para crear productos superiores en diseño, ejecución y calidad. Sin embargo, la programación en par es un cambio importante de cultura en la manera de cómo funciona un equipo de desarrollo.

A2.4.5.3.5 Verificación Detallada de Escritorio.

En la verificación detallada de escritorio, solamente una persona además del autor, examina el producto. En ocasiones, el autor no tiene forma de saber cómo se llevó a cabo la revisión. Una verificación detallada, depende del conocimiento, habilidad y disciplina de la persona que revisa, pero puede ser mucho más formal si esta persona lleva a cabo la revisión empleando listas de comprobación de defectos, métodos específicos de análisis y formas estandarizadas para registrar los expedientes.

Cuando se termina la verificación, la persona que revisa puede entregar una lista de defectos encontrados al autor o simplemente darle el producto marcado.

Este tipo de verificación es el acercamiento de revisión más barato que se pueda encontrar, porque implica solamente una revisión. Este método es conveniente para productos de trabajo poco arriesgados. Las personas que realizan estas verificaciones detalladas, promueven el comenzar a desarrollar una nueva cultura de revisión.

A2.4.5.3.6 Passaround.

El Passaround es una revisión detallada múltiple. En vez de pedir a un solo colega que realice la revisión, se entrega una copia del producto a varias personas y ellos llevan a cabo la tarea. Este método disminuye dos riesgos importantes que ocurren en la

[Escribir texto]

verificación detallada de escritorio: uno es que la persona que hace la revisión, no proporcione un trabajo eficiente, pues evita que el revisor haga un trabajo pobre.

Este método permite que cada revisor vea los comentarios que otros han escrito, minimiza la redundancia y revela diferencias que se dan en la interpretación del producto. Resulta una buena opción cuando no se pueden dar reuniones para realizar las revisiones, debido a limitaciones geográficas o de tiempo.

A2.4.5.3.7 Revisión Personal.

Las revisiones personales son el tipo de revisión más informal que existe. Puede ser que solucionen el problema en el momento inmediato, pero tienen poco impacto más adelante.

A2.5. ¿Cómo Seleccionar el Método más Apropriado?: CASO INEGI.

Una forma de seleccionar el método más apropiado de revisión es considerar el riesgo; que es la probabilidad de que un producto tenga defectos y los daños potenciales que ocasione. Para productos con un alto riesgo, es recomendable hacer uso de las inspecciones formales y para productos con un riesgo más bajo, se puede hacer uso de técnicas simples.

Algunos factores que incrementan el riesgo de un producto son:

- El uso de nuevas tecnologías, técnicas o herramientas.
- Una lógica compleja o algoritmos que son difíciles de entender pero que tienen que ser precisos y optimizados.
- Presión excesiva en horarios de desarrollo.
- Inadecuado entrenamiento o experiencia de desarrollo.
- Partes críticas en la seguridad del producto con modos de fallo severos.
- Componentes arquitectónicos dominantes que proporcionan una base para la evolución del producto.
- Un gran número de fallas que tienen que ser revisadas de forma particular y son difíciles de estimular durante la fase de pruebas.
- Componentes que se piensan reutilizar.
- Componentes que servirán como modelo o plantillas para otros componentes.
- Componentes con interfaces múltiples que afectan varias partes del producto

A2.5.1 Estado actual: Ejemplos de Escenarios.

Con lo anterior podemos direccionar las revisiones y pruebas hechas en el Departamento de Pruebas Modulares a Sistemas de Información, de acuerdo a la documentación que acompañe a los Sistemas y aplicaciones direccionados para su revisión:

- Escenario 1: Sólo con manual de operación o requerimiento: *Caso Sistema Integral de Administración (SIA): Módulo Contabilidad.*

Se deberá formar un **equipo de inspección o revisión** (de 3 a 6 personas dependiendo de disponibilidad y competencia con otros proyectos) y dado que seguramente el sistema ya ha sido desarrollado completamente, se podrán solamente realizar las Pruebas del Sistema (funcionalidad) (PS) de las diferentes versiones o preliminares del SIA, posteriormente la Prueba de Aceptación (PA).

Se deberá utilizar el Método de Revisión en Equipo, menos riguroso que el Método de Inspección ó Revisión. Las revisiones en equipo son un tipo de inspección más ligera, que aunque están planeados y estructurados, son menos formales y menos comprendidos. El equipo de revisión tiene varios nombres como el de “walkthrough estructurado” o simplemente “revisión”. Estas revisiones siguen varios de los pasos encontrados en una inspección.

Se optó por aplicar el Método de Revisión en Equipo (menos riguroso que el Método de Inspección ó Revisión, consultar Anexo 2 **Guía Metodológica del Proceso de Pruebas**). Las revisiones en equipo son un tipo de inspección más ligera, que aunque están planeados y estructurados, son menos formales y menos comprendidos. El equipo de revisión tiene varios nombres como el de

“walkthrough estructurado” o simplemente “revisión”. Estas revisiones siguen varios de los pasos encontrados en una inspección. Los pasos fueron:

1.- Los participantes reciben el material a revisar, con días de anticipación a la reunión para que lo puedan estudiar y analizar por ellos mismos.

2.- El equipo reúne datos sobre el esfuerzo dedicado a la revisión y el número de defectos encontrados (se usará el formato de revisión del manual o requerimiento, de la sección A2.11 del Anexo 2 **Guía Metodológica para el Proceso de Pruebas**). Sin embargo, se simplifican o se omiten la descripción y las etapas de inspección, y algunos roles de los participantes pueden ser combinados.

3.- El autor puede conducir al equipo de revisión, de no ser así, el **Administrador de Pruebas** lo hará, mientras que en una inspección el autor no puede funcionar como asesor. A diferencia de la mayoría de las inspecciones, se omite el papel del lector.

4.- En lugar de tener un lector que vaya describiendo por partes el producto, el líder de la revisión (**Administrador de Pruebas**), sólo pregunta a los participantes (**testers**) si tienen comentarios sobre una sección o página específica del trabajo.

5.- Es más costoso tener un equipo de revisión que tener a un colega realizando la revisión con ayuda de un checklist detallado, la diferencia es que los participantes del equipo (**testers**) encontrarán otros o más errores.

6.- Como en cualquier reunión, pueden presentarse discusiones u opiniones diferentes o bien, acuerdos que puedan ser olvidados más adelante, por lo tanto la persona que dirige la revisión (**Administrador de Pruebas**)), deberá registrar todo lo que se trató durante la junta. La persona que registra o escribe, captura todo lo que sucede durante la reunión pero usando formatos estándar que la organización ha definido o adoptado (en este caso no están definidos, pero se anotan en un cuaderno exclusivo para el proyecto).

7.- Al final de la reunión, el equipo lleva a cabo una valoración del producto de trabajo al autor o Líder de Proyecto y decide cómo verificar los cambios que el autor realizará durante la reanudación del trabajo.

8.- Se utilizarán los reportes de revisión de la sección A2.11 llamados Formatoprueb.doc e imágenes.doc del Anexo 2 **Guía Metodológica para el Proceso de Pruebas**, los reportes de cada iteración (versión) de las inspecciones con sus respectivas pruebas se van depurando quitando/agregando los defectos corregidos/agregados para las nuevas versiones de Sistema o Producto de Software, Se cuenta el número de defectos encontrados en el reporte de revisión correspondiente en cada versión del sistema o aplicación puesto a revisión, obteniéndose el promedio de defectos encontrados en el sistema desarrollado por ciclo de prueba y se colocan en una tabla, mismos que se grafican mediante una Gráfica de Control X-barras, para observar si el proceso es estable y tenía la capacidad de cumplir los requerimientos en el tiempo esperado por el encargado del sistema, ésta grafica la realiza el **Administrador de Pruebas** (mediante una función de R llamada **qcc** véase sección A2.12 Programas en R), la interpreta y entrega este material al encargado del sistema acompañado de los reportes de observaciones de la sección A2.11, (actualmente solo se entregan los reportes correspondientes). Asimismo, en estos reportes se encuentran las calificaciones de las cualidades de: Severidad, Ocurrencia y Detección de cada una de las observaciones o defectos encontrados, de acuerdo a las tablas del AMEF incluidas en el Capítulo 3 de este trabajo, obteniéndose el número de prioridad de atención: npr, sugerida para las observaciones o errores.

9.- El ciclo continúa hasta liberar el Producto de Trabajo, el ciclo se cierra al entregar un reporte compendio con todas las observaciones atendidas y corregidas por fecha.

- Escenario 1A: Con manual de operación o requerimiento y servidor de pruebas actualizado. No se han presentado actualmente ejemplos de este escenario.
- Escenario 2: Con funcionalidad y/o casos de uso. No se han presentado actualmente ejemplos de este escenario.
- Escenario 3: Con funcionalidad y/o casos de uso y casos de prueba acompañados con un checklist propuesto por el desarrollador del sistema.

Caso ConProVe.

[Escribir texto]

- Escenario 4 (deseado): Con manual de operación o requerimiento, funcionalidad y/o casos de uso y casos prueba acompañados con checklist, servidor de pruebas actualizado. No se han presentado actualmente ejemplos de este escenario.

Para el Escenario 3

Se forma un **equipo de inspección o revisión** (de 3 a 6 personas dependiendo de disponibilidad y competencia con otros proyectos) y se realizan pruebas de integración ya que el sistema se estaba desarrollando aún, una vez que los desarrolladores terminaron sus pruebas unitarias. Se prueba funcionalidad Pruebas del Sistema (PS) y la Prueba de Aceptación (PA). Dado que presentan además casos de uso, se podrán generar algunos casos prueba que no estén incluidos en la batería de casos de pruebas entregados en el checklist correspondiente.

Desgraciadamente no se contó con un servidor para realizar las pruebas exclusivamente del departamento, se probó en el mismo servidor junto con las pruebas de los desarrolladores y de los usuarios.

Se utilizó el Método de Revisión Formal más riguroso (de acuerdo con la Guía **Metodológica del Proceso de Pruebas**, véase Anexo 2): La Inspección se realizó de la siguiente manera:

- 1.- se asignaron roles específicos a cada integrante del equipo de trabajo.
- 2.- El proceso de inspección más común incluyó siete etapas que son: planeación, visión general, preparación, reunión, reanudación de trabajo, dar seguimiento y análisis.
- 3.- Contamos con checklist de defectos comúnmente encontrados en diversos tipos de trabajo de software, reglas para construir estos productos.
- 4.- Los participantes a excepción del autor del producto, condujeron la reunión (puede ser el **Administrador de Pruebas** o uno de los desarrolladores), presentan el material al equipo de inspección (demás **testers**) y registran los puntos que se hayan mencionarlo en la reunión (el registro lo puede hacer el **Administrador de Pruebas**).

5.- Los participantes se prepararon para la reunión de la inspección, examinando el material por ellos mismos para entenderlo y para encontrarle detalles que pudieran estar dando problemas. El equipo reúne datos sobre el esfuerzo dedicado a la revisión y el número de defectos encontrados (se usará el formato de revisión del manual o requerimiento, de la sección A2.11 del Anexo 2 **Guía Metodológica para el Proceso de Pruebas**. Durante la reunión, el lector (**Administrador de Pruebas** o uno de los **testers**) presenta el material a los demás inspectores (**testers**), quienes hacen más comentarios e identifican posibles defectos. El lector ayudo al equipo a darle la misma interpretación a cada parte del producto para que los inspectores comparen lo que ellos han comprendido, con lo que el lector había expresado.

6.- Al final de la reunión, el equipo llevó a cabo una valoración del producto de trabajo al autor o Líder de Proyecto.

7.- Y se decidió cómo verificar los cambios (en el servidor de prueba) que el autor iba realizando durante la reanudación del trabajo.

8.- Los reportes de cada iteración de las inspecciones con sus respectivas pruebas se iban depurando quitando/agregando los defectos corregidos/agregados para las nuevas versiones del Sistema o Producto de Software. Se cuenta el número de defectos encontrados en el reporte de revisión correspondiente en cada versión del sistema o aplicación puesto a revisión, obteniéndose el promedio de defectos encontrados en el sistema desarrollado por ciclo de prueba y se colocan en una tabla, mismos que se grafican mediante una Gráfica de Control X-barra, para observar si el proceso es estable y tenía la capacidad de cumplir los requerimientos en el tiempo esperado por el encargado del sistema, , ésta grafica la realiza el **Administrador de Pruebas** (mediante una función de R llamada **qcc** véase sección A2.12 Programas en R), la interpreta y entrega este material al encargado del sistema acompañado de los reportes de observaciones de la sección A2.11, (actualmente solo se entregan los reportes correspondientes). Asimismo se calificaron las cualidades de: Severidad, Ocurrencia y Detección de cada una de las observaciones o defectos encontrados, de acuerdo a las tablas del AMEF incluidas en el Capítulo 3 de este trabajo, obteniéndose el

[Escribir texto]

número de prioridad de atención: npr, sugerida para las observaciones o errores.

9.- El ciclo continuó hasta liberar el Producto de Trabajo, el ciclo se cierra al entregar un reporte compendio con todas las observaciones atendidas y corregidas por fecha.

A2.6. ¿Quién Debe y qué Debe Revisar?

En la Tabla A2.6.1 se puede apreciar quienes deben estar involucrados en una revisión y que es lo que deben revisar.

Etapas	Persona involucrada
Especificación de Requerimientos	Analista, líder de proyecto, usuario, diseñador, quien prueba el sistema, un experto en pruebas, soporte técnico
Arquitectura	Arquitecto, diseñador, analista de requerimientos, líder de proyecto, pruebas de integración
Detalle de Diseño	Diseñador, arquitecto, programador, quien prueba la integración
Codificación	Programador, diseñador, mantenimiento
Documentación de Pruebas	Quien realiza las pruebas, programador o arquitecto o analista
Documentación del Sistema	Autor, líder de proyecto, mantenimiento, programador
Interfaces de Usuario	Diseñador UI, analista, quien realiza las pruebas, usuario, un experto en el tema, experto en usabilidad y probador de sistema

Tabla A2.6.1 Personas involucradas en una revisión

A2.7. Automatización de Pruebas.

La automatización de pruebas significa hacer la prueba manualmente una vez, registrarla en un script tipo macro de excel con lenguaje programático, y poder ejecutarla múltiples veces, tantas como sea necesario para verificar que un cambio hecho a la aplicación, ya sea por aumentar o cambiar funcionalidad, como por supuesto, por corregir un defecto, no haya afectado las funciones previamente probadas y certificadas. La automatización es útil durante un proyecto, pero resulta una gran ventaja o prácticamente indispensable para un sistema de producción.

La automatización de pruebas con herramientas no sustituye al proceso de Planeación, Análisis, Diseño e Implementación, más bien son un complemento para este proceso.

Las herramientas para la automatización de pruebas sirven y son útiles, pero no como sustituto del proceso de pruebas, que implica una labor intelectual, analítica, de razonamiento, deducción, inducción, conclusión y priorización.

No se deben de automatizar el 100% de los casos de pruebas , más bien hacerlo solo donde resulte más efectivo.

Hablando de costos y beneficios, el costo está en crear la prueba manualmente la primera vez, en hacer los cambios necesarios a los scripts y en darles mantenimiento cada vez que se requiera por modificaciones a la aplicación. El beneficio está en la ganancia en tiempo de ejecutar una prueba de regresión mucho más rápido cuando ocurren cambios que hay que liberar en breve, y en las posibilidades de ejecutar múltiples combinaciones de acciones de usuario y de datos de entrada a la prueba.

A2.8. Conclusiones.

Definimos el marco en el que se aplican las inspecciones de software partiendo de la base de un desarrollo profesional del mismo en el cual lo principal será la calidad de éste, estableciendo como criterios de calidad: Correctitud y Completitud [Fre90] como los principales e imprescindibles.

De éste modo podemos afirmar que un software en el que se controle la calidad no puede dejar de lado un proceso de revisión formal del mismo, como podemos observarlo en las normas ISO o CMMI del SEI, quizás con otro nombre pero contemplando los mismos objetivos.

[Escribir texto]

El proceso de inspección debe ser llevado a cabo por personas que conozcan tanto del dominio específico, comprendiendo el documento de Especificación de Requerimientos de Software (SRS por sus siglas en Inglés) [DAV93], así como la tecnología aplicada a las soluciones que serán objeto de la inspección. A partir de éste background en el equipo de inspección, deberán respetarse las etapas planteadas precedentemente, creando las condiciones necesarias para maximizar la sinergia que se produzca sobre todo en la etapa de “examen”.

Por último si se decide incorporar inspecciones como parte del ciclo de vida del software a construir, no debe dejar de medirse el proceso para controlarlo e incorporarlo como feedback de los sucesivos proyectos.

A2.10 Bibliografía.

Guía Metodológica para la Fase de Pruebas en el desarrollo de software versión 1.0 (GM_10_Pruebas, Iteración_07_Pruebas) del Proyecto TRITON (CIMAT).

Pruebas del software, Profesor: Juan Antonio López Quesada. Facultado de Informática. [Http://dis.um.es/~lopezquesada](http://dis.um.es/~lopezquesada)

A2.11 ANEXOS de la Guía Metodológica

DEPARTAMENTO DE PRUEBAS MODULARES A LOS SISTEMAS DE INFORMACIÓN

REPORTE DE OBSERVACIONES AL SISTEMA

Nombre del Sistema Revisado:

Módulos:

Periodo de Revisión:

Fecha de elaboración del reporte:

No.	OBSERVACIÓN	TESTER	E	S	O	D	NPR	SOLUCIÓN/ACLARACIÓN	ATENDIÓ	FECHA ACLARADO
1	OB1		p	10	10	10	1000			
2	OB2									
3	OB3									
4	OB4									
5	OB5									
6	OB6									
7	OB7									
8	OB8									
9	OB9									
10	OB10									
11	OB11									
12	OB12									
13	OB13									
14	OB14									
15	OB15									
16	OB16									
17	OB17									
18	OB18									
19	OB19									
20	OB20									
21	OB21									
22	OB22									
23	OB23									

ESTATUS (E)

NA = No aplica

PV = Pendiente siguiente versión

PD = Pendiente de recibir aclaración y por lo tanto detenido

PC = Pendiente de recibir aclaración, pero continua su revisión

C = Corregido

F = Revisión finalizada

Uso exclusivo del Depto. Pruebas

SEVERIDAD = S

OCURRENCIA = O

DETECTABILIDAD = D

NPR = Número de prioridad

(Atención de la observación de 1 a 1000)



Archivo de imágenes (IMÁGENES.DOC) que testifican la detección de los defectos de un sistema, cada error se encuentra identificado por un vínculo con el archivo anterior (FORMATOPRUEB.DOC) facilitando el reconocimiento de los mismos.

[OB1](#)

[OB2](#)

[OB3](#)

[OB4](#)

[OB5](#)

[OB6](#)

[OB7](#)

[OB8](#)

[OB9](#)

[OB10](#)

[OB11](#)

[OB12](#)

[OB13](#)

[OB14](#)



SISTEMA DE CONTROL DE PRODUCCIÓN Y VENTAS (CONPROVE)

Módulo de Ventas

Casos de Prueba para la Pantalla de Facturación Parte 1

Descripción del Caso de Prueba	Entradas del caso de Prueba	Salida
Datos Generales de la Factura		
1.- Prueba de datos requeridos		
Fuente	Requerido	
Tipo de Venta	Requerido	
Clave Vendedor	Requerido	
Tipo de Cliente	Requerido	
Subtipo de Cliente	Requerido	
2.- Campo Fuente		
Ninguna	No hace nada	
Cotización	Aparece un cuadro para capturar el no. cotización	
Nota de Venta	Aparece un cuadro para capturar el no. Nota de Venta	
3.- No. Cotización o Nota de Venta		
	Requerido si Fuente igual a cotización o nota de venta	
Validar tipo y longitud	Numerico(9)	
4.- Si Fuente igual a cotización o nota de venta y oprime el botón de cargar		
	Verificar que se cargue correctamente toda la información de la cotización o de la Nota de Venta	
En ambos casos	Verificar que se puedan cambiar los datos de cliente	

DGITI	DD SI
DEPARTAMENTO DE PRUEBAS MODULARES A LOS SISTEMAS DE INFORMACIÓN	
REPORTE DE OBSERVACIONES AL REQUERIMIENTO O MANUAL	
Nombre del Sistema Revisado: <u>Sistema de Enumeración en Palm para el Censos Agropecuario 2007</u>	
Módulos: <u>Procedimiento 1</u>	
Periodo de Revisión: <u>Del 05 al 09 de marzo del 2007</u>	
Fecha de elaboración del reporte: <u>09 de marzo del 2007</u>	

No.	OBSERVACIÓN AL REQUERIMIENTO O MANUAL	ESTATUS	SOLUCIÓN/ACLARACIÓN	ATENDIÓ	FECHA ACLARADO
1	Es mejor enumerar las pantallas, para fines de referencia. En la <u>página 20 del requerimiento.</u>	PC			
2	Pantalla 3. Faltan los iconos de fecha y hora, en el requerimiento se tienen que especificar antes de iniciar la enumeración. En la <u>página 67 del requerimiento.</u>	PC			
3	Pantalla 4. Al requerimiento le falta el icono de negativos. En la <u>página 73 del requerimiento.</u>	PC			
4	Falta indicar en la pantalla descrita en el requerimiento la versión que se esté utilizando del sistema.	PC			
5	Sugerencia para el requerimiento: P13. Esta pantalla permite realizar el respaldo de la información del registro que se esté enumerando, si así lo considera el enumerador, es mejor que al oprimir el botón de navegación "siguiente", el sistema emita un mensaje "Existen más registros a enumerar?" si la respuesta es si, envía a la P5, si la respuesta es No, que envíe a la P14, ya que es más ilustrativo para quien captura y evita errores de captura al querer abandonar la aplicación. En la <u>página 78.</u>	PC			
6	Por cuestiones de estética es mejor estandarizar el uso de acentos pues en algunas preguntas lo omiten por ejemplo: P5 en la <u>página 103 del requerimiento.</u>	PC			
7	En la pantalla P18 del Requerimiento falta el icono del reloj para ingresar el fin de la jornada laboral. <u>Página 103.</u>	PC			

ESTATUS: NA = No aplica
 PV = Pendiente siguiente versión del Requerimiento o Manual
 PD = Pendiente de recibir aclaración y por lo tanto detenido

PC = Pendiente de recibir aclaración, pero continua su revisión
 C = Corregido

ANEXO 3.- DATOS ESTADÍSTICOS Y SCRIPTS DE R.

Tabla A3.1 Tabla con la información de defectos encontrados en el CONPROVE

Ciclo	Severidad	Ocurrencia	Detección	NPR=S*O*D	Defectos por ciclo
22/01/2008	7	10	1	70	10
	5	10	2	100	
	9	10	1	90	
	5	10	2	100	
	5	10	2	100	
	9	10	3	270	
	5	10	2	100	
	10	10	1	100	
	9	10	2	180	
	9	10	2	180	
29/01/2008	7	10	1	70	6
	5	10	2	100	
	9	10	1	90	
	5	10	2	100	
	9	10	2	180	
	9	10	2	180	
07-Feb-08	7	10	1	70	6
	5	10	2	100	
	9	10	1	90	
	5	10	2	100	
	9	10	2	180	
	9	10	2	180	
14/02/2008	9	10	2	180	10
	9	10	2	180	
	9	10	1	90	
	7	10	1	70	
	7	10	3	210	
	9	10	2	180	
	9	10	2	180	
	9	10	2	180	
	9	10	2	180	
	10	10	1	100	
18/02/2008	9	10	1	90	3
	7	10	1	70	
	7	10	3	210	
28/02/2008	1	1	1	1	0

[Escribir texto]

Tabla A3-2 Tabla con la información de defectos ciclos de prueba del CONPROVE

Ciclo	Número de ciclo	Defectos por ciclo
22/01/2008	1	10
29/01/2008	2	6
07-Feb-08	3	6
14/02/2008	4	10
18/02/2008	5	3
28/02/2008	6	0

[Escribir texto]

Tabla A3.5 Tabla con la información de defectos encontrados en el proyecto SIA-Contabilidad

Ciclo	Severidad	Ocurrencia	Detección	NPR=S*O*D	Defectos por día	Defectos por ciclo	Media de defectos por ciclo
17/01/2008	4	10	1	40	11		
	1	10	1	10			
	5	10	3	150			
	5	10	1	50			
	2	10	2	20			
	1	10	1	10			
	3	10	1	30			
	2	10	1	20			
	1	10	1	10			
	1	10	1	10			
	1	10	1	10			
18/01/2008	9	10	1	90	15	26	13
	10	10	1	100			
	9	10	1	90			
	9	10	1	90			
	5	10	1	50			
	4	10	1	40			
	6	10	2	120			
	5	10	1	50			
	6	10	2	120			
	3	10	1	30			
	9	10	1	90			
	10	10	2	200			
	5	10	1	50			
	1	10	1	10			
	3	10	1	30			
21/01/2008	10	10	1	100	8		
	2	10	1	20			
	10	10	1	100			
	10	10	1	100			
	5	10	1	50			
	10	10	1	100			
	5	10	1	50			
	1	10	1	10			
22/01/2008	9	10	1	90			
	4	10	1	40			
	10	10	1	100			
	10	10	1	100			

[Escribir texto]

	9	10	1	90			
	9	10	1	90			
	9	10	1	90			
	7	10	1	70			
	5	10	1	50			
	10	10	1	100			
	2	10	1	20			
	2	20	2	80			
	1	10	1	10			
	4	10	1	40			
	10	10	1	100			
	5	10	1	50			
	5	10	1	50			
	8	10	1	80	18		
23/01/2008	2	10	1	20			
	8	10	1	80			
	10	10	3	300			
	7	10	3	210			
	9	10	2	180			
	6	10	2	120			
	9	10	2	180	7		
24/01/2008	6	10	1	60			
	10	10	1	100			
	6	10	1	60			
	2	10	1	20			
	5	10	1	50			
	9	10	1	90			
	5	10	1	50			
	1	10	1	10			
	10	10	1	100			
	9	10	1	90			
	9	10	1	90			
	1	10	1	10			
	10	10	1	100	13	46	11.5
28/01/2008	1	10	1	10			
	9	10	2	180			
	3	10	1	30			
	8	10	2	160	4		
29/01/2008	1	10	1	10			
	1	10	1	10			
	10	10	1	100			
	9	10	1	90			
	1	10	1	10			
	9	10	2	180			

[Escribir texto]

	1	10	1	10			
	4	10	1	40			
	6	10	1	60			
	5	10	1	50	10		
30/01/2008	9	10	1	90			
	10	10	1	100			
	5	10	1	50			
	10	10	2	200			
	5	10	2	100			
	8	10	1	80	6	20	6.66666667
01/02/2008	7	10	2	140			
	9	10	1	90			
	7	10	2	140	3		
05/02/2008	7	10	2	140			
	10	10	2	200			
	6	10	2	120			
	7	10	2	140			
	4	10	2	80			
	9	10	2	180			
	5	10	2	100			
	9	10	2	180	8		
06/02/2008	8	10	1	80			
	4	10	2	80			
	2	10	1	20			
	9	10	2	180			
	9	10	2	180	5		
07/02/2008	1	10	1	100			
	2	10	1	20			
	8	10	1	80			
	9	10	1	90	4		
08/02/2008	9	10	1	90			
	9	10	1	90			
	2	10	3	60			
	9	10	3	270			
	5	10	3	150			
	4	10	3	120			
	8	10	2	160	7	27	5.4
11/02/2008	8	10	2	160			
	5	10	2	100			
	2	10	1	20			
	2	20	2	80			
	4	10	1	40			
	9	10	1	90			
	2	10	1	20			

[Escribir texto]

	5	10	1	50			
	1	10	2	20			
	2	10	1	20			
	5	10	1	50			
	10	10	1	100			
	10	10	1	100			
	10	10	2	200			
	10	10	1	100			
	9	10	1	90			
	10	10	1	100	17		
12/02/2008	10	10	1	100			
	10	10	1	100			
	8	10	1	80			
	8	10	1	80			
	9	10	1	90			
	2	10	1	20	6		
13/02/2008	9	10	1	90			
	9	10	1	90	2		
14/02/2008	1	10	1	10			
	3	10	1	30			
	1	10	1	10			
	1	10	1	10	4		
15/02/2008	9	10	1	90			
	1	10	1	10			
	2	10	1	20			
	9	10	1	90			
	2	10	1	20			
	9	10	1	90	6	35	7
18/02/2008	10	10	1	100			
	10	10	1	100			
	1	10	1	10			
	2	10	1	20			
	8	10	1	80			
	9	10	1	90			
	2	10	1	20			
	2	10	1	20			
	1	10	1	10			
	1	10	2	20			
	2	10	1	20			
	2	10	1	20			
	2	10	1	20	13		
19/02/2008	6	10	1	60			
	10	10	1	100			
	7	10	1	70			

[Escribir texto]

	9	10	1	90			
	7	10	1	70			
	5	10	1	50			
	2	10	1	20			
	9	10	1	90	8		
20/02/2008	10	10	1	100			
	9	10	1	90			
	10	10	1	100			
	9	10	1	90	4		
21/02/2008	8	10	1	80			
	8	10	1	80			
	10	10	1	100			
	6	10	1	60			
	10	10	1	100			
	4	10	1	40			
	5	10	1	50			
	9	10	1	90			
	10	10	1	100	9	34	8.5
25/02/2008	10	10	1	100			
	2	10	1	20			
	10	10	1	100			
	8	10	1	80			
	9	10	1	90			
	2	10	1	20			
	5	10	1	50			
	5	10	1	50	8		
27/02/2008	10	10	1	100			
	1	10	2	20			
	10	10	1	100			
	1	10	2	20			
	2	10	1	20			
	9	10	1	90			
	9	10	1	90			
	10	10	1	100			
	1	10	1	10			
	6	10	1	60			
	10	10	1	100	11		
28/02/2008	2	10	1	20			
	5	10	1	50			
	10	10	1	100			
	10	10	1	100			
	2	10	1	20			
	8	10	1	80			
	2	10	2	20			

[Escribir texto]

	9	9	1	81			
	5	10	1	50			
	9	10	1	90			
	7	10	1	70			
	9	10	1	90	12		
29/02/2008	1	9	2	18			
	1	10	2	20			
	1	10	1	10			
	9	9	2	162			
	5	9	1	45			
	1	10	2	20			
	1	9	1	9			
	1	9	1	9			
	9	9	1	81			
	9	9	1	81			
	2	10	2	40	11	42	10.5
03/03/2008	5	10	1	50			
	9	9	1	81			
	9	10	1	90			
	5	9	1	45			
	9	9	1	81			
	10	9	1	90			
	1	10	1	10			
	9	10	1	90			
	5	10	1	50			
	5	10	1	50			
	2	10	1	20			
	8	10	1	80			
	1	10	1	10	13		
04/03/2008	1	10	1	10			
	1	10	1	10			
	5	9	1	45			
	9	9	1	81			
	9	9	1	81			
	9	9	1	81			
	10	10	1	100	7		
06/03/2008	5	10	1	50			
	10	10	1	100			
	10	10	1	10			
	9	10		90			
	10	10	1	100			
	1	10	1	10			
	10	9	1	90			
	9	9	1	81			

[Escribir texto]

	9	9	1	81			
	2	10	1	20			
	10	10	1	100			
	1	10	1	10			
	9	9	1	81			
	9	9	1	81			
	4	9	1	36			
	8	9	1	72	16		
07/03/2008	9	9	1	81			
	1	10	1	10			
	1	9	2	18			
	2	10	1	20			
	5	9	1	45	5	41	10.25
10/03/2008	5	9	1	45			
	5	9	1	45			
	1	10	1	10			
	9	9	1	81	4		
11/03/2008	1	10	1	10			
	9	9	1	81			
	10	9	1	90			
	5	9	1	45			
	1	10	1	10			
	9	9	1	81			
	3	9	1	27			
	10	9	1	90			
	2	10	1	20			
	5	9	1	45			
	9	9	1	81	11		
12/03/2008	9	9	1	81			
	5	9	1	45			
	5	9	1	45			
	5	9	1	45			
	10	10	1	100			
	5	9	1	45			
	2	10	1	20			
	10	9	1	90			
	9	9	1	81			
	1	10	1	10			
	10	9	1	90			
	9	9	1	81			
	2	10	1	20	13		
13/03/2008	2	10	1	20			
	10	9	1	90			
	2	9	2	36			

[Escribir texto]

	1	10	1	10			
	5	9	1	45			
	1	10	1	10	6	34	8.5
18/03/2008	1	10	1	10			
	9	9	1	81			
	9	9	1	81			
	8	9	1	72			
	1	10	1	10			
	1	10	1	10	6		
19/03/2008	5	9	1	45			
	1	10	1	10			
	9	9	1	81			
	1	10	1	10			
	5	9	1	45	5	11	5.5
24/03/2008	1	10	1	10			
	2	10	1	20			
	5	9	1	45			
	10	9	1	90			
	1	10	1	10	5		
26/03/2008	10	9	1	90			
	10	9	1	90			
	10	9	1	90			
	2	10	1	20			
	1	10	1	10	5		
28/03/2008	9	10	1	90			
	1	10	1	10			
	8	9	1	72			
	10	10	1	100			
	5	10	1	50			
	10	10	1	100			
	10	10	1	100	7	17	5.66666667
31/03/2008	5	10	1	50			
	9	10	1	90			
	10	9	1	90			
	4	10	1	40			
	9	9	1	81			
	6	10	1	60			
	1	9	1	9			
	10	9	1	90			
	5	10	1	50			
	10	9	1	90			
	6	10	1	60	11		
01/04/2008	1	9	1	9			
	9	10	1	90			

[Escribir texto]

	1	10	1	10			
	5	10	1	50	4		
03/04/2008	9	9	1	81			
	10	9	1	90	2		
04/04/2008	6	10	1	60			
	5	10	1	50			
	1	10	1	10			
	2	9	1	18			
	8	9	1	72			
	5	9	1	45			
	10	9	1	90	7	24	6
07/04/2008	3	9	1	27			
	9	9	1	81	2		
09/04/2008	5	9	1	45			
	5	9	1	45			
	10	9	1	90			
	10	9	1	90	4		
10/04/2008	8	9	1	72			
	5	9	1	45			
	8	9	1	72			
	7	9	1	63	4	29	3.333333333
14/04/2008	2	9	2	36			
	3	9	1	27			
	5	9	1	45			
	5	9	1	45	4	4	4
22/04/2008	5	9	1	45			
	9	10	1	90			
	5	9	1	45	3		
24/04/2008	5	9	1	45			
	9	9	1	81			
	10	9	1	90	3	6	3
28/04/2008	4	9	1	36			
	10	9	1	90	2	2	2
06/05/2008	5	9	1	45			
	4	10	1	40	2		
07/05/2008	10	9	1	90			
	10	9	1	90	2	4	2
14/05/2008	10	9	1	90			
	10	9	1	90	2	2	2
12/06/2008	4	9	1	36			
	10	9	1	90			
	10	9	1	90	3	3	3
17/06/2008	10	9	1	90			
	10	9	1	90	2	2	2

[Escribir texto]

Tabla A3.5 Tabla con la información de defectos del SIA-Contabilidad

Ciclo	Defectos por ciclo	Media de defectos por ciclo
1	26	13
2	46	11.5
3	20	6.66666667
4	27	5.4
5	35	7
6	38	8,5
7	42	10.5
8	41	10.25
9	34	8.5
10	11	5.5
11	17	5.66666667
12	24	6
13	29	3.33333333
14	4	4
15	6	3
16	2	2
17	4	2
18	2	2
19	3	3
20	2	2

SCRIPTS EN R.

```
# 1.er paso instalar qcc. del menú packages elegir opción instalar paquete
```

```
library (qcc)
```

```
# Recupera el archivo de datos del SIA.
```

```
datos <- read.csv ("d://qcc//Defectos-A5-SIA.csv",header=TRUE)
```

```
obj <- qcc(datos, type="xbar")
```

```
media.def <- mean (datos[,"Defectos"])
```

```
sd.def <- sd (datos[,"Defectos"])
```

```
# Límites de control (superior e inferior)
```

```
UCL <- media.def + 3 * sd.def
```

```
LCL <- media.def - 3 * sd.def
```

```
process.capability(obj, spec.limits=c(LCL,UCL))
```

```
## Process Capability análisis.
```

```
process.capability.sixpack(obj,spec.limits=c(LCL,UCL),target= media.def)
```

[Escribir texto]

```
# Recupera el archivo de datos del CONPROVE.
datos <- read.csv ("d://qcc//Defectos-A4-CONPROVE.csv",header=TRUE)
obj <- qcc(datos, type="xbar")
```

```
media.def <- mean (datos["Defectos"])
sd.def <- sd (datos["Defectos"])
# Límites de control (superior e inferior)
UCL <- media.def + 3 * sd.def
LCL <- media.def - 3 * sd.def
process.capability(obj, spec.limits=c(LCL,UCL))
```

```
## Process Capability análisis.
process.capability.sixpack(obj,spec.limits=c(LCL,UCL),target= media.def)
```

```
#####grafica del articulo de cangussu
```

```
library (qcc)
# Recupera el archivo de datos del artículo
datos <- read.csv ("d://qcc//Defectos-Articulo.csv",header=TRUE)
obj <- qcc(datos, type="xbar")
```

```
media.def <- mean (datos["Defectos"])
media.def
sd.def <- sd (datos["Defectos"])
# Límites de control (superior e inferior)
UCL <- media.def + 3 * sd.def
LCL <- media.def - 3 * sd.def
process.capability(obj, spec.limits=c(LCL,UCL))
```

```
## Process Capability análisis.
process.capability.sixpack(obj,spec.limits=c(LCL,UCL),target= media.def)
```

```
##### Ensayos.....
```

```
plot (datos,type="b",ylim=c(max(c(0,min(LCL))),max(UCL)))
plot (datos,type="b",ylim=c(min(LCL),max(UCL)))
abline ( h = media.def )
abline ( h = UCL, col = "red", lty = 2)
abline ( h = LCL, col = "blue", lty = 2)
```

```
#####
```

```
pchShow <-
function(extras = c("x",".", "o","O","0","+","-","|","%","##"),
  cex = 3, ## good for both .Device=="postscript" and "x11"
  col = "red3", bg = "gold", coltext = "brown", cextext = 1.2,
  main = paste("plot symbols : points (... pch = *, cex =",
  cex,")"))
```

[Escribir texto]

```
{
nex <- length(extras)
np <- 26 + nex
ipch <- 0:(np-1)
k <- floor(sqrt(np))
dd <- c(-1,1)/2
rx <- dd + range(ix <- ipch %% k)
ry <- dd + range(iy <- 3 + (k-1) - ipch %% k)
pch <- as.list(ipch) # list with integers & strings
if(nex > 0) pch[26+ 1:nex] <- as.list(extras)
plot(rx, ry, type="n", axes = FALSE, xlab = "", ylab = "",
      main = main)
abline(v = ix, h = iy, col = "lightgray", lty = "dotted")
for(i in 1:np) {
  pc <- pch[[i]]
  ## 'col' symbols with a 'bg'-colored interior (where available) :
  points(ix[i], iy[i], pch = pc, col = col, bg = bg, cex = cex)
  if(cextext > 0)
    text(ix[i] - 0.3, iy[i], pc, col = coltext, cex = cextext)
}
}
```

[Escribir texto]

ANEXO 4 JUSTIFICACIÓN DEL CAMBIO DE RUMBO EN LA INVESTIGACIÓN DEL TEMA DE TESIS.

----- Original Message -----

From: [Sergio nava](#)

To: clemola@cimat.mx

Sent: Monday, April 14, 2008 1:46 PM

Subject: como ves?

Hola Cuau,

del articulo de Cangussu "Monitoring the Software Test Process Using Statistical Process Control: A Logarithmic Approach"

de la seccion 3.1. textualmente dice:

3.1 Does the STP Actually Present an Exponential Behavior?

SPClog as described in Section 3.3 can be applied to any process presenting an exponential behavior. Here, we use the STP to exemplify the applicability of this technique. It is widely experienced that defects are relatively easier to find during the early portions of the system test phase. Perhaps this is because there are many of them and not difficult to trigger [18]. Finding defects becomes increasing difficult with the progress of time perhaps because there are fewer of them and often require a specific combination of events to be triggered. This behavior implies an exponential shape for the decay in the number of accumulated defects and, as defects are removed, failure intensity and reliability also present an exponential behavior [12]. However, in some situations the STP presents a S rather than an exponential shape. This behavior is perhaps due to factors such as learning and adaptation present at the beginning of the STP [19]. In both cases, the assumption here is that the STP is a well defined, not ad-hoc, process. Our concern is with processes where the learning and adaptation effects are minimized and the dominant behavior can be characterized as exponential.

y segun lo que veo es justamente eso, que no hay bien implantada una metodologia para hacer las pruebas, sino que en cada desarrollo como que es a la medida. lo que no se es bien como ponerlo. Pues este es un supuesto importante, y que seria el equivalente a tener en control el proceso pues estaria estandarizado y bien establecido.

p.d. no me ha escrito andres.

saludos

sergio

[Escribir texto]

ANEXO 5. DATOS OBTENIDOS DE LAS PRUEBAS DEL VIII CENSO AGROPECUARIO 2007.

Departamento de Pruebas Modulares a los Sistemas de Información

Sistema: Captura en palm del VIII Censo agrícola, ganadero y forestal._Procedimiento: Subrutina de visita dirigida,

Coordinación Estatal: _____

Periodo de Revisión: _____

Pantalla	Caso Prueba	No. Prueba	Escenario	Variables de captura						Resultado esperado	Comportamiento real del sistema	Evaluación del Flujo
				SI	NO							
26	1	2	En caso de que la actividad principal sea agricultura, verificar despliegue del mensaje "¿en este terreno cultiva caña de azúcar?"	SI	NO					Guardar registro		OK
				X	X							
26	1	3	Verificar que el sistema despliegue el mensaje: "¿ Deseas agregar otro registro?"	SI	NO					Despliega pantalla 26, limpia, para agregar otro registro		OK
				X								
26	1	4	Verificar que el sistema despliegue el mensaje: "¿Deseas agregar otro registro?"	SI	NO					Pasa a la pantalla 27	SI ES SOLO UN MUNICIPIO PASA A LA PANTALLA CUESTIONARIOS Que indica cuantos cuestionarios se levantaran para ese	x
					X							

[Escribir texto]

												productor, así como el número de terrenos y los municipios en los que se encuentran.	
27	1	1	Probar que las opciones sean editables y excluyentes. Que haya avance y regreso a la siguiente pantalla y que no permita avanzar sin selección de alguna opción.	SI	NO							Pasa a la pantalla 29	NA
					X								

ANEXO 6. ANALISIS DE MODOS Y EFECTOS DE FALLAS POTENCIALES (AMEF) [26].

Introducción

Tradicionalmente, en los procesos de comercialización de bienes y servicios, y con el objetivo de satisfacer al cliente, las empresas se han visto en la obligación de ofrecer garantías, es decir, de comprometerse con el cliente por un período determinado a reparar o sustituir de manera total o parcial los productos que presenten defectos operacionales o de construcción.

Aun cuando este compromiso representa tranquilidad para el consumidor, el hecho de no poder disponer del producto durante un período de reparación o sustitución, o que éste se averíe con mucha frecuencia; representa un motivo de insatisfacción, el cual se traduce como una pérdida de prestigio para el proveedor.

De igual manera, en aquellos casos en que el producto o servicio es utilizado en lugares remotos o en condiciones muy críticas, la garantía pasa a un segundo plano y el interés principal del cliente recae en que el producto no falle.

Por estos motivos, es deseable colocar en el mercado un producto o servicio que no presente defectos, y para tal fin en el presente trabajo se expone el Análisis de modos y efectos de fallas potenciales (AMEF) como un procedimiento de gran utilidad para aumentar la confiabilidad y buscar soluciones a los problemas que puedan presentar los productos de software y los procesos de desarrollo antes de que estos ocurran(cultura de prevención de fallas).

Reseña Histórica

La disciplina del AMEF fue desarrollada en el ejercito de la Estados Unidos por los ingenieros de la National Agency of Space and Aeronautical (NASA), y era conocido como el procedimiento militar MIL-P-1629, titulado "Procedimiento para la Ejecución de un Modo de Falla, Efectos y Análisis de criticabilidad" y elaborado el 9 de noviembre de 1949; este era empleado como una técnica para evaluar la confiabilidad y para determinar los efectos de las fallas de los equipos y sistemas, en el éxito de la misión y la seguridad del personal o de los equipos.

En 1988 la Organización Internacional para la Estandarización (ISO), publicó la serie de normas ISO 9000 para la gestión y el aseguramiento de la calidad; los requerimientos de esta serie llevaron a muchas organizaciones a desarrollar sistemas

[Escribir texto]

de gestión de calidad enfocados hacia las necesidades, requerimientos y expectativas del cliente, entre estos surgió en el área automotriz el QS 9000, éste fue desarrollado por la Chrysler Corporation, la Ford Motor Company y la General Motors Corporation en un esfuerzo para estandarizar los sistemas de calidad de los proveedores; de acuerdo con las normas del QS 9000 los proveedores automotrices deben emplear Planeación de la Calidad del Producto Avanzada (APQP), la cual necesariamente debe incluir AMEF de diseño y de proceso, así como también un plan de control.

Posteriormente, en febrero de 1993 el grupo de acción automotriz industrial (AIAG) y la Sociedad Americana para el Control de Calidad (ASQC) registraron las normas AMEF para su implementación en la industria, estas normas son el equivalente al procedimiento técnico de la Sociedad de Ingenieros Automotrices SAE J - 1739.

Los estándares son presentados en el manual de AMEF aprobado y sustentado por la Chrysler, la Ford y la General Motors; este manual proporciona lineamientos generales para la preparación y ejecución del AMEF.

Actualmente, el AMEF se ha popularizado en todas las empresas automotrices americanas y ha empezado a ser utilizado en diversas áreas de una gran variedad de empresas a nivel mundial.

Requerimientos del AMEF

Para hacer un AMEF se requiere lo siguiente:

- Un equipo de personas con el compromiso de mejorar la capacidad de diseño para satisfacer las necesidades del cliente.
- Diagramas esquemáticos y de bloque de cada nivel del sistema, desde sub-ensambles hasta el sistema completo.
- Especificaciones de los componentes, lista de piezas y datos del diseño.
- Especificaciones funcionales de módulos, sub-ensambles, etc.
- Requerimientos de manufactura y detalles de los procesos que se van a utilizar.

Formas de AMEF (en papel o electrónicas) y una lista de consideraciones especiales que se apliquen al producto.

[Escribir texto]

¿Qué es AMEF?

El Análisis de modos y efectos de fallas potenciales, AMEF, es un proceso sistemático para la identificación de las fallas potenciales del diseño de un producto o de un proceso antes de que éstas ocurran, con el propósito de eliminarlas o de minimizar el riesgo asociado a las mismas.

Por lo tanto, el AMEF puede ser considerado como un método analítico estandarizado para detectar y eliminar problemas de forma sistemática y total, cuyos objetivos principales son:

- Reconocer y evaluar los modos de fallas potenciales y las causas asociadas con el diseño y manufactura de un producto
- Determinar los efectos de las fallas potenciales en el desempeño del sistema
- Identificar las acciones que podrán eliminar o reducir la oportunidad de que ocurra la falla potencial
- Analizar la confiabilidad del sistema
- Documentar el proceso

Aunque el método del AMEF generalmente ha sido utilizado por las industrias automotrices, éste es aplicable para la detección y bloqueo de las causas de fallas potenciales en productos y procesos de cualquier clase de empresa, ya sea que estos se encuentren en operación o en fase de proyecto; así como también es aplicable para sistemas administrativos y de servicios.

Beneficios del AMEF

La eliminación de los modos de fallas potenciales tiene beneficios tanto a corto como a largo plazo. A corto plazo, representa ahorros de los costos de reparaciones, las pruebas repetitivas y el tiempo de paro. El beneficio a largo plazo es mucho más difícil de medir puesto que se relaciona con la satisfacción del cliente con el producto y con sus percepción de la calidad; esta percepción afecta las futuras compras de los productos y es decisiva para crear una buena imagen de los mismos.

Por otro lado, el AMEF apoya y refuerza el proceso de diseño ya que:

- Ayuda en la selección de alternativas durante el diseño.

[Escribir texto]

- Incrementa la probabilidad de que los modos de fallas potenciales y sus efectos sobre la operación del sistema sean considerados durante el diseño.
- Proporciona información adicional para ayudar en la planeación de programas de pruebas concienzudos y eficientes.
- Desarrolla una lista de modos de fallas potenciales, clasificados conforme a su probable efecto sobre el cliente.
- Proporciona un formato documentado abierto para recomendar acciones que reduzcan el riesgo para hacer el seguimiento de ellas.
- Detecta fallas en donde son necesarias características de auto corrección o de leve protección
- Identifica los modos de fallas conocidos y potenciales que de otra manera podrían pasar desapercibidos.
- Detecta fallas primarias, pero a menudo mínimas, que pueden causar ciertas fallas secundarias.
- Proporciona un punto de visto fresco en la comprensión de las funciones de un sistema

Formato y elementos del AMEF

Para facilitar la documentación del análisis de fallas potenciales y sus consecuencias, la empresa Ford estandarizó un formato para la realización del AMEF; sin embargo, dado que cada empresa representa un caso particular es necesario que éste sea preparado por un equipo multidisciplinario integrado por personal con experiencia en diseño, manufactura, ensamblaje, servicio, calidad y confiabilidad. Es muy importante que, aún cuando se realicen modificaciones, se mantengan los siguientes elementos:

Encabezado.

Tipo De AMEF: se debe especificar si el AMEF a realizar es de diseño o de proceso.

Nombre/Número De Parte O Proceso: Se debe registrar el nombre y número de la parte, ensamble o proceso que se está analizando. Utilice sufijos, cambie letras y/o el número de Reporte de Problema/solicitud de cambio (CR/CR), según corresponda.

Responsabilidad de Diseño/Manufactura: Anotar el nombre de la operación y planta de manufactura que tiene responsabilidad primaria de la maquinaria, equipo o proceso de ensamble, así como el nombre del área responsable del diseño del componente, ensamble o sistema involucrado.

[Escribir texto]

Otras Áreas Involucradas: Anotar cualesquier área/departamento u organizaciones afectadas o involucradas en el diseño o función del (los) componente(s), así como otras operaciones manufactureras o plantas involucradas.

Proveedores Y Plantas Afectadas: Enlistare cualquier proveedor o plantas manufactureras involucradas en el diseño o fabricación de los componentes o ensambles que se están analizando.

Vehículo (S)/Año Modelo (depende de donde se está haciendo): Registra todas las líneas de vehículos que utilizarán la parte/proceso que se está analizando y el año modelo.

Fecha De Liberación De Ingeniería: Indica el último nivel de Liberación de Ingeniería y fecha para el componente o ensamble involucrado.

Fecha Clave De Producción: Registrar la fecha de producción apropiada.

Preparado Por: Indicando el nombre, teléfono, dirección y compañía del ingeniero que prepara el AMEF.

Fecha Del AMEF: Anotar la fecha en que se desarrolló el AMEF original y posteriormente, anotar la fecha de la última revisión del AMEF.

Descripción/propósito del proceso.

Anotar una descripción simple del proceso u operación que se está analizando e indicar tan brevemente como sea posible el propósito del proceso u operación que se esté analizando.

Modo de falla potencial.

Se define como la manera en que una parte o ensamble puede potencialmente fallar en cumplir con los requerimientos de liberación de ingeniería o con requerimiento específicos del proceso.

Se hace una lista de cada modo de falla potencial para la operación en particular; para identificar todos los posibles modos de falla, es necesario considerar que estos pueden caer dentro de una de cinco categorías:

Falla Total

Falla Parcial

Falla Intermitente

Falla Gradual

Sobrefuncionamiento

Efectos de falla potencial.

[Escribir texto]

El siguiente paso del proceso de AMEF, luego de definir la función y los modos de falla, es identificar las consecuencias potenciales del modo de falla; ésta actividad debe de realizarse a través de la tormenta de ideas y una vez identificadas estas consecuencias, deben introducirse en el modelo como efectos.

Se debe asumir que los efectos se producen siempre que ocurra el modo de falla. El procedimiento para Consecuencias Potenciales es aplicado para registrar consecuencias remotas o circunstanciales, a través de la identificación de modos de falla adicionales, el procedimiento es el siguiente:

Se comienza con un modelo de falla (MF-1), y una lista de todas sus consecuencias potenciales

Separar aquellas consecuencias que se asumen como resultado siempre que MF-1 ocurra, éstas se identifican como efectos MF-1.

Se escriben modos de falla adicionales para las consecuencias restantes (consecuencias que pudiesen resultar si MF-1 ocurre, dependiendo de las circunstancias bajo las cuales ocurra). Los nuevos modos de falla implican que las consecuencias inusuales ocurrirán al incluir las circunstancias bajo las cuales ocurren, que se asume, resultarán siempre que los modos de falla y sus circunstancias especiales ocurran; éstas se deben identificar como efectos de los modos de fallas adicionales.

Severidad.

El primer paso para el análisis de riesgos es cuantificar la severidad de los efectos, éstos son evaluados en una escala del 1 al 10 donde 10 es lo más severo.

A continuación se presentan las tablas con los criterios de evaluación para diseño y para el proceso:

Tabla 3. 4 Criterios de la evaluación y sistema de graduación sugeridos para la severidad de los efectos para un diseño AMEF.

Efecto	Criterios: Severidad del efecto para AMEF	Fila
Alerta peligrosa	El incidente afecta la operación segura del producto o implica la no conformidad con la regulación del gobierno, sin alarma.	10
- peligroso; con	El incidente afecta la operación segura del producto o implica la no	9

[Escribir texto]

alarma	conformidad con la regulación del gobierno, con alarma.	
Muy Arriba	El producto es inoperable con pérdida de función primaria.	8
Alto	El producto es operable, pero en el nivel reducido del funcionamiento	7
Moderado	El producto es operable, pero el ítem(s) de la comodidad o de la conveniencia es inoperable.	6
Bajo	El producto es operable a un nivel reducido de funcionamiento.	5
Muy Bajo	La mayoría de los clientes notan los defectos.	4
De menor Importancia	Los clientes medios notan los defectos.	3
Muy de menor importancia	El ajuste y el final o el chirrido y el ítem del traqueteo no se conforma. Los clientes exigentes notan los defectos.	2
Ninguno	Ningún efecto	1

Tabla 3. 5 Criterios de la evaluación y sistema de graduación sugeridos para la severidad de los efectos para un proceso AMEF

Efecto	Criterios: Severidad del efecto para AMEF	Fila
- peligroso; sin alarma	Puede poner en peligro al operador del ensamblaje. El incidente afecta la operación o la no conformidad segura del producto con la regulación del gobierno. El incidente ocurrirá sin alarma.	10
- peligroso; con alarma	Puede poner en peligro al operador del ensamblaje. El incidente afecta la operación o la no conformidad segura del producto con la regulación del gobierno. El incidente ocurrirá con alarma.	9
Muy Arriba	Interrupción importante a la cadena de producción. 100% del producto puede ser desechado. El producto es inoperable con pérdida de función primaria.	8
Alto	Interrupción de menor importancia a la cadena de producción. El producto puede ser clasificado y una porción desechada. El producto es operable, pero en un nivel reducido del funcionamiento.	7
Moderado	Interrupción es de menor importancia a la cadena de producción. Una porción del producto puede ser desechado (no se clasifica). El producto es operable, pero un cierto ítem(s) de la comodidad / de la conveniencia es inoperable	6
Bajo	Interrupción es de menor importancia a la cadena de producción. 100% del producto puede ser devuelto a trabajar. El producto es operable, pero algunos ítems de la comodidad / de la conveniencia funcionan en un nivel reducido del	5

[Escribir texto]

	funcionamiento.	
Muy Bajo	Interrupción es de menor importancia a la cadena de producción. El producto puede ser clasificado y una porción puede ser devuelto a trabajar. La mayoría de los clientes notan el defecto.	4
De menor importancia	Interrupción es de menor importancia a la cadena de producción. Una porción del producto puede ser devuelto a trabajar en línea solamente hacia fuera-de-estación. Los clientes medios notan el defecto.	3
Muy De menor importancia	Interrupción es de menor importancia a la cadena de producción. Una porción del producto puede ser devuelto a trabajar en línea solamente en-estación. Los clientes exigentes notan el defecto.	2
Ninguno	El modo de fallo no tiene ningún efecto.	1

Características Especiales

El AIAG define una característica especial del producto como un producto característico para cuál razonablemente anticipó la variación podría afectar perceptiblemente una seguridad o la conformidad del producto con estándares o regulaciones gubernamentales, o es probable afectar perceptiblemente la satisfacción de cliente con un producto. Ford Motor Company divide características especiales en dos categorías: Características críticas y características significativas

Las características críticas son definidas por Ford como producto o requisitos del proceso que afecten conformidad con la regulación del gobierno o la función segura del producto, y que requieren acciones o controles especiales. En un diseño AMEF, se consideran las características críticas del potencial. Una característica crítica potencial existe para cualquier clasificación de la severidad mayor que o el igual a 9. En el proceso AMEF, se refieren como características críticas reales.

Cualquiera característica con una severidad de 9 o 10 que requiera un control especial asegurar la detección es una característica crítica.

Los ejemplos del producto o de los requisitos del proceso que podrían ser características críticas incluyen dimensiones, especificaciones, pruebas, secuencias de ensamblaje, los útiles, los empalmes, los esfuerzos de torsión, las autógenas, las conexiones, y los usos componentes. Las acciones o los controles especiales necesarios para resolver estos requisitos pueden implicar la fabricación, ensamblaje, un surtidor, envío, el vigilar, o examen.

Las características significativas requieren controles especiales porque son importantes para la satisfacción de cliente. Los grados de la severidad entre 5 y 8 se juntaron con una ocurrencia que clasificaba mayor de 3 indican características significativas. En un diseño AMEF, son potenciales Características Significativas. En el proceso AMEF, si un control especial se

[Escribir texto]

requiere para asegurar la detección entonces una característica significativa real existe. Las compañías no han estandarizado un método para agrupar y denotar características especiales del producto. La nomenclatura y la notación variarán.

Causas de fallas potenciales.

Luego de que los efectos y la severidad han sido listadas, se deben de identificar las causas de los modos de falla.

En el AMEF de diseño, las causas de falla son las deficiencias del diseño que producen un modo de falla. Para el AMEF de proceso, las causas son errores específicos descritos en términos de algo que puede ser corregido o controlado.

Ocurrencia.

Las causas son evaluadas en términos de ocurrencia, ésta se define como la probabilidad de que una causa en particular ocurra y resulte en un modo de falla durante la vida esperada del producto, es decir, representa la remota probabilidad de que el cliente experimente el efecto del modo de falla.

EL valor de la ocurrencia se determina a través de las siguientes tablas, en caso de obtener valores intermedios se asume el superior inmediato, y si se desconociera totalmente la probabilidad de falla se debe asumir una ocurrencia igual a 10.

Tabla 3. 6 Criterios de la evaluación y sistema de graduación sugeridos para la ocurrencia del incidente en un diseño AMEF.

Probabilidad del incidente	Porcentajes de averías	Fila
Muy Arriba: El incidente es casi inevitable	1 en 2 \geq	10
	1 en 3	9
Alto: Incidentes repetitivos	1 en 8	8
	1 en 20	7
Moderado: Incidentes ocasionales	1 en 80	6
	1 en 400	5
	1 de 2,000	4
Bajo: Relativamente pocos incidentes	1 en 15,000	3
	1 en 150,000	2
Telecontrol: El incidente es inverosímil	1 en 1,500,000 \leq	1

[Escribir texto]

Tabla 3. 7 Criterios de la evaluación y sistema de graduación sugeridos para la ocurrencia del incidente en un proceso AMEF

Probabilidad del incidente	Incidente Tarifas	Pk de C	Fila
Muy Arriba: El incidente es casi inevitable	1 en 2 \geq	< 0.33	10
	1 en 3	0.33 IV	9
Alto: Asociado generalmente a los procesos similares que han fallado anteriormente	1 en 8	0.51 IV	8
	1 en 20	0.67 IV	7
Moderado: Asociado generalmente a los procesos similares previos que han experimentado incidentes ocasionales, pero no en proporción es importantes	1 en 80	0.83 IV	6
	1 en 400	1.00 IV	5
	1 de 2,000	1.17 IV	4
Bajo: Los incidentes aislados se asocian a procesos similares	1 en 15,000	1.33 IV	3
Muy Bajo: Solamente los incidentes aislados se asocian a procesos casi idénticos	1 en 150,000	1.50 IV	2
Telecontrol: El incidente es inverosímil.	1 en 1,500,000 IV	1.67 IV	1

Controles actuales.

Los controles actuales son descripciones de las medidas que previenen que ocurra el modo de falla o detectan el modo de falla en caso de que ocurran. Los controles de diseño y proceso se agrupan de acuerdo a su propósito:

Tipo 1: Estos controles previenen la causa o el modo de falla de que ocurran, o reduce su ocurrencia

Tipo 2: Estos controles detectan la causa del modo de falla y guían hacia una acción correctiva

Tipo 3: Estos controles detectan el modo de falla antes de que el producto llegue al cliente

Detección.

[Escribir texto]

La detección es una evaluación de las probabilidades de que los controles del proceso propuestos (listados en la columna anterior) detecten el modo de falla, antes de que la parte o componente salga de la localidad de manufactura o ensamble.

No es probable que verificaciones de control de calidad al azar detecten la existencia de un defecto aislado y por tanto no resultarán en un cambio notable del grado de detección. Un control de detección válido es el muestreo hecho con bases estadísticas.

Tabla 3. 8 Criterios de la evaluación y sistema de graduación sugeridos para la detección de una causa del incidente o del modo de fallo en un diseño AMEF

Detección	Probabilidad de la detección por control del diseño	Fila
Incertidumbre Absoluta	El control del diseño no detecta una causa potencial del incidente o del modo de fallo subsecuente, o no hay control del diseño.	10
Muy Alejado	La probabilidad muy alejada de que el control del diseño detecte una causa potencial del incidente o del modo de fallo subsecuente	9
Alejado	La probabilidad alejada de que el control del diseño detectará una causa potencial del incidente o del modo de fallo subsecuente	8
Muy Bajo	La probabilidad muy baja el control del diseño detectará un potencial Causa del incidente o del modo de fallo subsecuente	7
Bajo	La probabilidad baja el control del diseño detectará un potencial Causa del incidente o del modo de fallo subsecuente	6
Moderado	La probabilidad moderada de que el control del diseño detectará una causa potencial del incidente o del modo de fallo subsecuente	5
Moderadamente Alto	La probabilidad moderadamente alta de que el control del diseño detectará una causa potencial del incidente o del modo de fallo subsecuente	4
Alto	La alta probabilidad de que el control del diseño detectará una causa potencial del incidente o del modo de fallo subsecuente	3
Muy Alto	La probabilidad muy alta de que el control del diseño detectará una causa potencial del incidente o del modo de fallo subsecuente	2
Casi Seguro	El control del diseño detectará casi ciertamente una causa potencial del incidente o del modo de fallo subsecuente	1

[Escribir texto]

Tabla 3. 9 Criterios de la evaluación y sistema de graduación sugeridos para la detección de una causa del incidente o del modo de fallo en un proceso AMEF.

Detección	Probabilidad de la detección por el control del proceso.	Fila
Casi imposible	Ninguno de los controles disponibles detecta el incidente de Modo o causa.	10
Muy Alejado	Los controles actuales tienen una probabilidad muy alejada de detectar modo o causa de fallo.	9
Alejado	Los controles actuales tienen una probabilidad alejada de detectar modo o causa de fallo.	8
Muy Bajo	Los controles actuales tienen una probabilidad muy baja de detectar modo o causa de fallo.	7
Bajo	Los controles actuales tienen una probabilidad baja de detectar Modo o causa de fallo.	6
Moderado	Los controles actuales tienen una probabilidad moderada de detectar modo o causa de fallo.	5
Moderadamente Alto	Los controles actuales tienen una probabilidad moderadamente alta de detectar modo o causa de fallo.	4
Alto	Los controles actuales tienen una alta probabilidad de detectar modo o causa de fallo.	3
Muy Alto	Los controles actuales tienen una probabilidad muy alta de detectar modo o causa de fallo.	2
Casi Seguro	Los controles actuales detectan casi seguros el modo o la causa de fallo. Los controles confiables de la detección se saben con procesos similares.	1

NPR

El número de prioridad de riesgo (NPR) es el producto matemático de la severidad, la ocurrencia y la detección, es decir:

$$\text{NPR} = \text{S} * \text{O} * \text{D}$$

Este valor se emplea para identificar los riesgos más serios para buscar acciones correctivas.

Acción (es) recomendada (s).

Cuando los modos de falla han sido ordenados por el NPR, las acciones correctivas deberán dirigirse primero a los problemas y puntos de mayor grado e elementos críticos. La intención de

[Escribir texto]

cualquier acción recomendada es reducir los grados de ocurrencia, severidad y/o detección. Si no se recomienda ninguna acción para una causa específica, se debe indicar así.

Un AMEF de proceso tendrá un valor limitado si no cuenta con acciones correctivas y efectivas. Es la responsabilidad de todas las actividades afectadas el implementar programas de seguimiento efectivos para atender todas las recomendaciones.

Área/individuo responsable y fecha de terminación (de la acción recomendada)

Se registra el área y la persona responsable de la acción recomendada, así como la fecha meta de terminación.

Acciones tomadas.

Después de que se haya completado una acción, registre una breve descripción de la acción actual y fecha efectiva o de terminación.

Npr resultante.

Después de haber identificado la acción correctiva, se estima y registra los grados de ocurrencia, severidad y detección finales. Se calcula el NPR resultante, éste es el producto de los valores de severidad, ocurrencia y detección.

El ingeniero en proceso es responsable de asegurar que todas las acciones recomendadas sean implementadas y monitoreadas adecuadamente. El AMEF es un documento viviente y deberá reflejar siempre el último nivel de diseño.

Secuencia de procedimientos para la elaboración del AMEF

Una vez identificados los elementos del AMEF, es necesario conocer cómo se debe llevar a cabo, es decir, el orden lógico que deben de llevar las operaciones; esta secuencia se expresa mejor a través del diagrama de flujo presentado a continuación.

Cabe Destacar que previamente se debe de haber definido al equipo responsable para la ejecución del AMEF, así como también se debe realizar un análisis previo para la recolección de datos.

El papel del AMEF en los Sistemas de Calidad.

Se pueden considerar como los objetivos principales de cualquier sistema de calidad, la prevención y la solución de problemas.

Para la prevención de problemas los sistemas de calidad emplean el Despliegue de la Función Calidad (QFD), el Análisis del Árbol de Falla (FTA), el Análisis de Árbol de Falla Reverso (RFTA), la Planeación de la Calidad del Producto Avanzada (APQP) y el AMEF, éste último es empleado tanto de manera directa como indirecta a través de la APQP y del Diseño de Experimentos (DOE), el cual es un elemento importante para la prevención y la solución de problemas; en cuanto a ésta última los sistemas de calidad utilizan principalmente el Mejoramiento Continuo, el Sistema Operativo de Calidad (QOS), las ocho disciplinas para la solución de problemas (8D) y el Plan de Control, cuya elaboración requiere directamente del AMEF, de herramientas de Control Estadístico de Proceso (SPC) y la consideración de las características especiales establecidas a través del AMEF.

[Escribir texto]

Relación del AMEF con las Normas ISO 9000

Las normas ISO 9000 solo definen directrices y modelos, no indican procedimientos a ser implementados ni las estrategias correspondientes que deberán ser definidas por cada empresa.

La serie ISO 9000 es especialmente aplicable cuando es necesario comprobar al cliente, como requisito contractual, que están siendo considerados un conjunto de parámetros de calidad previamente establecidos. En estos casos, el cliente exige contractualmente la comprobación de la calidad, no sólo del proyecto de desarrollo.

Entre los requerimientos establecidos en la norma 9000:2000 se hace referencia al control de diseño y al control del proceso, en sus cláusulas se establece como requisito la verificación de los mismos incluyendo un análisis de fallas y de sus correspondientes efectos. Esta verificación debe confirmar que los datos resultantes del proyecto cumplen las exigencias establecidas, a través de actividades de control de proyecto, tales como la realización y registro del análisis crítico de proyecto. El AMEF puede ser considerado particularmente como uno de los métodos más útiles y eficientes para tal fin.

Conclusión

Mediante la realización del presente informe, se establece la gran importancia y el alcance de los beneficios que proporciona el Análisis de Modo y Efectos de Falla Potencial como una herramienta para examinar todas las formas en que un producto o proceso pueda fallar; además se hace una revisión de la acción que debe tomar para minimizar la probabilidad de falla o el efecto de la misma.

Dado que para la mayoría de los productos y procesos no es económico llevar a cabo el AMEF para cada componente, se hace necesaria la realización de los elementos críticos que deben ser sometidos al mismo.

Aunque el AMEF es muy valioso como una técnica de advertencia temprana, la prueba definitiva viene dado por el uso del producto por parte del cliente.

Sin embargo la experiencia de campo llega demasiado tarde, y es aquí donde resalta la importancia de que ésta sea precedida por el AMEF para que las empresas puedan simular el uso de sus productos y procesos en el campo de trabajo.

BIBLIOGRAFÍA

Incluir referencias a todas las lecturas realizadas (libros, artículos, presentaciones, etc.)

- [1] D. J. Wheeler and D. S. Chambers, *Understanding Statistical Process Control*. SPC Press, 1992.
- [2] J. F. Manji, "SPC inspires global quality culture for multinational giant," *Managing Automation*, November 1999.
- [3] J. L. Cawley, "Improving yields with statistical process control," *Circuits Assembly*, March 1999.
- [4] J. G. Surak, J. L. Cawley, and S. A. Hussain, "Integrating haccp and spc," *Food Quality*, May 1998.
- [5] M. A. Lantzy, "Application of statistical process control to the software process," in *Proceedings of the 9th Washington Ada symposium on Empowering software users and developers*, (New York, NY, USA), pp. 113–123, ACM Press, 1992.
- [6] E. F. Weller, "Practical applications of statistical process control," *IEEE Software*, vol. 17, pp. 48–55, May-June 2000.
- [7] P. S. Pande, R. P. Neuman, and R. R. Cav anagh, *The Six Sigma Way: How GE, Motorola, and Other Top Companies are Honing Their Performance*. McGraw-Hill, 2000.
- [8] D. N., "Sorting out six sigma and the cmm," *IEEE Software*, vol. 17, pp. 11–13, May-June 2000.
- [9] R. V. Binder, "Can a manufacturing quality model work for software?," *IEEE Software*, September-October 1997.
- [10] A. P. Mathur, F. D. Frate, P. Garg, and A. Pasquini, "On the correlation between code coverage and software reliability," in *Proceedings of the Sixth International Symposium on Software Reliability Engineering*, (Toulouse, France), pp. 124–132, IEEE Press, October 24-27 1995.
- [11] S. Biffl, "Using inspection data for defect estimation," *IEEE Software*, vol. 17, pp. 36–43, November/December 2000.
- [12] J. W. Cangussu, R. A. DeCarlo, and A. P. Mathur, "Feedback control of the software test process through measurements of software reliability," in *Proceeding of 12th International Symposium on Software Reliability Engineering*, (Hong Kong), pp. 232–241, November 2001.

[Escribir texto]

- [13] J. S. Oakland, *Statistical Process Control*. Butterworth Heinemann, fourth ed., 1999.
- [14] W. K. Ehrlich, J. P. Stampfel, and J. R. Wu, "Application of software reliability modeling to product quality and test process," in *Proceedings of ICSE*, pp. 108–116, 1990.
- [15] W. A. Florac and A. D. Carleton, *Measuring the Software Process: Statistical Process Control for Software Process Improvement*. SEI Series in Software Engineering, Addison-Wesley, 1999.
- [16] W. A. Florac, A. D. Carleton, and J. Barnard, "Statistical process control: Analyzing a space shuttle onboard software process," *IEEE Software*, vol. 17, pp. 97–106, July/August 2000.
- [17] P. Jalote and A. Saxena, "Optimum control limits for employing statistical process control in software process," *IEEE Transactions on Software Engineering*, vol. 28, no. 12, pp. 1126–1134, 2002.
- [18] J. W. Cangussu, R. A. DeCarlo, and A. P. Mathur, "A formal model for the software test process," *IEEE Transaction on Software Engineering*, vol. 28, pp. 782–796, August 2002.
- [19] J. W. Cangussu, R. A. DeCarlo, and A. P. Mathur, "Using sensitivity analysis to validate a state variable model of the software test process," *IEEE Transactions on Software Engineering*, vol. 5, May 2003.
- [20] D. E. Knuth, "The errors of TEX," *Software-Practice and Experience*, vol. 19, no. 7, pp. 607–685, 1989.
- [21] S. R. Dalal and C. L. Mallows, "Some graphical aids for deciding when to stop testing software," *IEEE Journal on Selected Areas in Communications*, vol. 8, pp. 169–175, February 1990.
- [22] R. Jacoby and Y. Tohma, "The hyper-geometric distribution software reliability growth model (hgdm): Precise formulation and applicability," in *14th Conference on Computer Software and Applications (COMPSAC'90)*, (Chicago, IL), pp. 13–19, IEEE, 31 October - 2 November 1990.
- [23] W. K. Ehrlich, J. P. Stampfel, and J. R. Wu, "Application of software reliability modeling to product quality and test process," in *12th International Conference on Software Engineering (ICSE'90)*, (Nice, France), pp. 108–116, IEEE, 26-30 March 1990.

[Escribir texto]

[24] J. W. Cangussu, R. A. DeCarlo, and A. P. Mathur, "Monitoring the Software Test Process Using Statistical Process Control: A Logarithmic Approach". September 1–5, 2003, Helsinki, Finland.

[25] "Generación Estadística de Casos Prueba basados en UML", artículo Publicado en 2002 por Matthias Riebisch, ilka Philippow, Marco Götze.

[26] Manual del AMEF, www.monografias.com . **Armando Hidalgo Mascorro** mando_hidalgo@yahoo.com.mx .

[27] Guía Metodológica para la Fase de Pruebas en el desarrollo de software versión 1.0 (GM_10_Pruebas, Iteración_07_Pruebas) del Proyecto TRITON (CIMAT).

[28] Institute for human and machine cognition: <http://cmap/ihmc.us> (CmapTools, version 4.12).

[Escribir texto]
