

UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE INGENIERÍA INDUSTRIAL Y DE SISTEMAS



**ASEGURAMIENTO DE LA CALIDAD DE
SOFTWARE EN LA SUNAT BASADO EN LA
METODOLOGIA MDSI**

INFORME DE SUFICIENCIA

**PARA OPTAR EL TÍTULO PROFESIONAL DE
INGENIERO DE SISTEMAS**

MIGUEL ANGEL TAMAYO TORRES

LIMA - PERU
2006

A mis queridos Padres Cesar e Inés, mi esposa Pilar y mis queridos Hijos Miguel y Gabriel que son mi orgullo y fuente de inspiración.

A mis queridos Abuelitos y a esa querida tierra del Distrito de Soraya en Apurímac por influir en mí el cariño y respeto a las tradiciones y costumbres de nuestro hermoso País.

Agradezco a mi Universidad y a todos aquellos que contribuyeron en mi formación profesional.

ÍNDICE

DESCRIPTORES TEMÁTICOS	5
RESUMEN EJECUTIVO	6
INTRODUCCIÓN	8
CAPITULO I	10
ANTECEDENTES	10
1.1 Información de la Empresa	10
1.1.1 Misión de la SUNAT.....	10
1.1.2 Visión de la SUNAT.....	10
1.1.3 Misión de la INSI.....	11
1.1.4 Visión de la INSI	11
1.1.5 Breve reseña de la SUNAT.....	11
1.1.6 Finalidad de la SUNAT	12
1.1.7 Evolución de la función de Calidad en la INSI.....	13
1.2 Diagnóstico Estratégico	14
1.2.1 Análisis FODA.....	15
1.2.1.1 Fortalezas y Debilidades	15
1.2.1.2 Oportunidades y Riesgos	16
1.2.2 Cuadro de Mando Integral de Kaplan y Norton	17
1.2.2 Cuadro de Mando Integral de Kaplan y Norton	18
1.3 Diagnóstico Funcional.....	22
1.3.1 Productos.....	22
1.3.2 Clientes	22
1.3.3 Proveedores.....	22
1.3.4 Procesos	23
1.3.5 Organización de la SUNAT	26
1.3.6 Organización de la INSI	27
1.3.6.1 Gerencia de Servicios a Usuarios	27

1.3.6.2 Gerencia de Desarrollo de Sistemas Tributarios y de Gestión ..	27
1.3.6.3 Gerencia de Desarrollo de Sistemas Aduaneros.....	28
1.3.6.4 Gerencia de Producción.....	29
1.3.6.5 Oficina de Seguridad Informática	29
1.3.6.6 Oficina de Investigación Tecnológica	30
1.3.6.7 Oficina de Planeamiento y Estrategias de Servicios Electrónicos	31
1.3.6.8 Oficina de Ingeniería de Procesos y Software.....	31
 CAPÍTULO II	 34
 MARCO TEORICO.....	 34
2.1 Definiciones Generales de Calidad.....	34
2.2 Definición de Aseguramiento de la Calidad	36
2.3 Normas ISO	36
2.3.1 Normas ISO.....	36
2.3.2 Normas ISO 9126.....	37
2.3.2.1 Funcionalidad	38
2.3.2.2 Fiabilidad	38
2.3.2.3 Usabilidad	39
2.3.2.4 Eficiencia.....	39
2.3.2.5 Mantenibilidad	39
2.3.2.6 Portabilidad	40
2.4 Metodología de Desarrollo de Sistemas MDSI	41
2.4.1 Modelamiento del Negocio (MPN).....	42
2.4.2 Modelamiento de Requerimientos (ASI).....	42
2.4.3 Modelamiento de Tecnología (DSI).....	43
2.4.4 Construcción y Pruebas (CPS).....	44
2.5 Herramientas de Productividad para el apoyo a las Pruebas	47
2.5.1 Pruebas Funcionales.....	47
2.5.2 Pruebas de Performance	48

2.5.3 Aseguramiento en la Calidad en el ciclo de vida de las aplicaciones	49
2.5.4 Pruebas de Caja Blanca.....	51
CAPITULO III	53
PROCESO DE TOMA DE DECISIONES	53
3.1 Planteamiento del Problema	53
3.2 Alternativas de Solución.....	61
3.3 Metodología de Solución.....	61
3.3.1 Evaluación por Criterios de Selección.....	61
3.3.2 Evaluación Financiera	62
3.3.2.1 Alternativa 1	65
3.3.2.2 Alternativa 2	67
3.4 Toma de Decisiones	69
3.5 Estrategias Adoptadas	70
3.5.1 Aplicación de una Metodología de Desarrollo	71
Procedimiento en la fase de Modelamiento de Requerimientos	72
3.5.1.1 ASI 9: Análisis de Consistencia y Especificación de Requerimientos	72
3.5.1.2 ASI 10: Definición del alcance de las pruebas	77
Procedimiento en la fase de Modelamiento de Tecnología.....	81
3.5.1.3 DSI 9: Diseño de la Migración y Carga Inicial de Datos	81
3.5.1.4 DSI 11: Especificación Técnica del Plan de Pruebas.....	82
3.5.1.4.1 DSI 11.1: Especificación del Entorno de Pruebas	82
3.5.1.4.2 DSI 11.2: Especificación Técnica de las Pruebas	85
Procedimiento en la construcción y Pruebas del Sistema.....	89
3.5.1.5 CPS 5 – CPS 6: Ejecución de las Pruebas Unitarias y las Pruebas de Integración.	89
3.5.1.6 CPS 7: Revisión de estándares de construcción.....	91
3.5.1.7 CPS 8: Evaluación de Procedimientos Normativos.....	93
3.5.1.8 CPS 10: Pruebas del Sistema.....	94

3.5.1.9 CPS 11: Pruebas de Aceptación del sistema	98
3.5.2 Implementación de Métricas de Calidad de Software	100
3.5.2.1 Indicadores de Funcionalidad.....	104
3.5.2.2 Indicadores de Eficiencia	105
3.5.2.3 Indicadores de Mantenimiento	106
3.5.2.4 Indicadores de Usabilidad	107
3.5.2.5 Indicadores de Fiabilidad	109
3.5.2.6 Indicador de Portabilidad.....	110
3.5.3 Utilización de Herramientas BTO	114
3.5.3.1 Plan de implantación de herramientas BTO.....	115
 CAPÍTULO IV	 119
 EVALUACION DE RESULTADOS	 119
4.1 Beneficios Tangibles	119
4.2 Estadísticas.....	119
 CAPÍTULO V	 123
 CONCLUSIONES Y RECOMENDACIONES	 123
 CONCLUSIONES.....	 123
 RECOMENDACIONES	 126
 GLOSARIO DE TÉRMINOS.....	 127
 BIBLIOGRAFÍA	 128
 ANEXOS	 129

DESCRIPTORES TEMÁTICOS

Superintendencia Nacional de Administración Tributaria

Aseguramiento de la calidad de software

Pruebas de software

Metodología de Desarrollo de Sistemas de Información MDSI

Métricas de calidad

Herramientas para el aseguramiento de la calidad de software

RESUMEN EJECUTIVO

La Intendencia Nacional de Sistemas de Información INSI no dispone de un proceso de aseguramiento de la calidad de Software, hoy en día la División de Control de Calidad DCC está encargada únicamente de las pruebas de los sistemas; un gran avance se produce cuando en el mes de Abril del 2005 se aprueba la Metodología de Desarrollo de Sistemas de Información MDSI la cual fue desarrollada por la Oficina de Ingeniería de Procesos de Sistemas OIPS.

El presente informe se inicia con una descripción del marco general de la Intendencia Nacional de Sistemas de Información, poniendo mayor énfasis en la participación de la División de Control de Calidad DCC en su función preponderante en cada una de las fases de la Metodología de Desarrollo de Sistemas de Información MDSI, en el marco teórico se describen primero los conceptos generales de la Calidad, en segundo lugar se rescatan los principales conceptos de la ISO 9126 que define una serie de criterios para medir y mejorar la calidad interna y externa del Software, en tercer lugar se describen cada una de las fases que comprenden la Metodología MDSI y por último se detallan algunas características de una Solución BTO (Business Technology Optimize) para el proceso de aseguramiento de la calidad del software.

En el capítulo de toma de decisiones se propone como solución a los problemas de la Intendencia Nacional de Informática un proceso de Aseguramiento de la Calidad de Software basado en la implementación de

procedimientos alineados a la metodología MDSI en conjunción con la implementación de métricas propuestas para medir y mejorar la calidad de software, complementados con el uso de herramientas BTO para mejorar la productividad en las pruebas de software.

La solución en el presente trabajo monográfico debe considerarse como una propuesta ya que complementa al proyecto de implantación del proceso de aseguramiento de la calidad de software que a la fecha se está poniendo en práctica en la INSI mediante la adquisición de herramientas BTO, por lo que se recomienda tomarse en cuenta para otros estudios relacionados con el tema. Cabe precisar que el conjunto de procedimientos y métricas propuestos en este estudio buscan aplicarse a desarrollos de sistemas orientados a Objetos.

La estrategia propuesta para el aseguramiento de la calidad de software se basa en realizar las siguientes acciones :

- Aplicación de una Metodología de desarrollo durante el ciclo de vida de las aplicaciones.
- Implementación de métricas de calidad de software.
- Utilización de herramientas BTO para el soporte al proceso de aseguramiento de calidad.

La problemática descrita de la Intendencia Nacional de Sistemas de Información no es ajena a las que enfrentan otras grandes organizaciones, por lo que se recomienda que la solución propuesta sea tomada como base para ser adaptada a su realidad.

Para el análisis de alternativas y la evaluación de resultados se ha realizado una evaluación en la SUNAT a las Herramientas BTO proporcionados por la organización Compuware Corporation, así como se ha recogido algunos datos estadísticos brindados por ésta.

INTRODUCCIÓN

La calidad del producto de software es una preocupación cada vez mayor en el ámbito informático y cuyos resultados inmediatos se aprecian en todas las actividades en donde se utilicen computadoras.

Lograr el éxito en la producción de *software* es hacerlo con calidad y demostrar su buena calidad. Esto sólo es posible con la implantación de un Sistema para el Aseguramiento de la Calidad del *Software* directamente relacionado con la política establecida para su elaboración.

El aseguramiento de la calidad pretende lograr que el producto reúna las características necesarias para satisfacer todos los requisitos del sistema de información. El equipo de aseguramiento de la calidad debe realizar un conjunto de actividades con la finalidad de asegurar que :

- Se reduzcan, eliminen y lo más importante, prevengan las deficiencias de los productos a obtener. Poner énfasis en las áreas de alto riesgo.
- Se utilice una metodología de desarrollo de software apropiada.
- Los proyectos utilicen estándares y procedimientos.
- Exista documentación para soportar mantenimiento y mejoras.
- La documentación se produzca durante y no después del desarrollo de software.
- Cada fase o tarea de software se complete a satisfacción, antes de continuar con la siguiente.
- Las desviaciones con respecto a estándares y procedimientos se detecten lo más pronto posible.

Para alcanzar estos objetivos, es necesario desarrollar un plan de aseguramiento de calidad específico en la SUNAT durante la planificación del proyecto de acuerdo a la estrategia de desarrollo adoptada según la metodología MDSI.

CAPITULO I

ANTECEDENTES

1.1 Información de la Empresa

Es necesario comenzar describiendo la misión y visión de la SUNAT y de la Intendencia Nacional de Sistemas de Información INSI en torno al cual se realizará el diagnóstico estratégico, además de precisar la finalidad de la empresa y los antecedentes de la INSI y la Gerencia de Sistemas a Usuarios como una unidad dependiente de INSI y cuya función principal es la de encargarse directamente de la calidad de Software entre otros. A continuación se describen brevemente los puntos antes mencionados :

1.1.1 Misión de la SUNAT

"Incrementar sostenidamente la recaudación tributaria y aduanera combatiendo, de forma honesta y justa, la evasión y el contrabando, mediante la generación efectiva de riesgo y servicios de calidad a los contribuyentes y usuarios del comercio exterior, así como, promoviendo el cumplimiento de sus obligaciones tributarias para contribuir con el bienestar económico y social de los peruanos".

1.1.2 Visión de la SUNAT

"Ser un emblema de distinción del Perú por su excelencia recaudadora y facilitadora del cumplimiento tributario y comercio exterior, comprometiendo al contribuyente y al usuario de comercio exterior como aliados en el desarrollo y bienestar del país".

1.1.3 Misión de la INSI

“Proveer a nuestros usuarios servicios y sistemas de información de calidad así como contribuir en la innovación de procesos, a través del uso automatizado de la información y el empleo estratégico de las tecnologías de información y comunicaciones”.

1.1.4 Visión de la INSI

“Ser impulsores del Gobierno electrónico en la SUNAT y en el Estado peruano en lo referido al ámbito tributario y el comercio exterior”.

1.1.5 Breve reseña de la SUNAT

La Superintendencia Nacional de Administración Tributaria es, de acuerdo a su Ley de creación, Ley N° 24829 y a su Ley General aprobada por Decreto Legislativo N° 501, una Institución Pública descentralizada del Sector Economía y Finanzas, dotada de personería jurídica de Derecho Público, patrimonio propio y autonomía económica, administrativa, funcional, técnica y financiera que, en virtud a lo dispuesto por el Decreto Supremo N° 061-2002-PCM, expedido al amparo de lo establecido en el numeral 13.1 del artículo 13° de la Ley N° 27658, ha absorbido a la Superintendencia Nacional de Aduanas, asumiendo las funciones, facultades y atribuciones que por ley, correspondían a esta entidad.

De acuerdo a la Ley General de la Superintendencia Nacional de Administración Tributaria, ésta tiene domicilio legal y sede principal en la ciudad de Lima y puede establecer dependencias en cualquier lugar del territorio nacional.

1.1.6 Finalidad de la SUNAT

La Superintendencia Nacional de Administración Tributaria (SUNAT) tiene por finalidad administrar, aplicar, fiscalizar y recaudar los tributos internos, con excepción de los municipales, así como proponer participar en la reglamentación de las normas tributarias.

Así mismo, la SUNAT proveerá servicios al contribuyente a fin de promover y facilitar el cumplimiento de sus obligaciones tributarias.

Las funciones de la SUNAT son las siguientes:

- a) Administrar todos los tributos internos con excepción de los municipales.
- b) Recaudar todos los tributos internos con excepción de los municipales.
- c) Fiscalizar el cumplimiento de las obligaciones tributarias a efecto de combatir la evasión fiscal.
- d) Otorgar el aplazamiento y/o fraccionamiento para el pago de la deuda tributaria de acuerdo con la Ley.
- e) Ejercer los actos de coerción para el cobro de la deuda tributaria de acuerdo con la Ley.
- f) Administrar los mecanismos de control tributario preventivo.
- g) Solicitar la adopción de medidas precautorias para cautelar la percepción de los tributos que administra y disponer la suspensión de las mismas cuando corresponda.
- h) Proponer al Ministerio de Economía y Finanzas la reglamentación de las normas tributarias.
- i) Resolver en primera instancia administrativa los recursos interpuestos por los contribuyentes o responsables; concediendo los recursos de apelación y dando cumplimiento a las resoluciones de Tribunal Fiscal, y en su caso a las del Poder Judicial.
- j) Sistematizar y ordenar la legislación vinculada con los tributos que administra.
- k) Proponer al Poder Ejecutivo los lineamientos tributarios para la celebración de acuerdos y convenios internacionales, así como de cooperación técnica y administrativa.

- l) Formar y mantener actualizado el Banco de Datos de Información Tributaria.
- m) Desarrollar programas de información y divulgación en materia tributaria.
- n) Desarrollar programas de capacitación.
- o) Establecer la Procuradurías de Administración Tributaria y proponer al Poder Ejecutivo el nombramiento
- p) Efectuar el remate de bienes comisados y/o embargados en el ejercicio de sus funciones.
- q) Emitir opinión respecto a los proyectos de acuerdos y convenios internacionales que celebre el Poder Ejecutivo sobre tributos cuya administración esté a su cargo.
- r) Establecer obligaciones de los contribuyentes en el ámbito de competencia de SUNAT, así como regular aspectos administrativos relacionados a la gestión institucional.
- s) Suscribir convenios interinstitucionales con entidades del sector público y privado relacionados con los fines de la SUNAT.
- t) Las demás funciones que sean compatibles con la finalidad de la Superintendencia Nacional de Administración Tributaria.

1.1.7 Evolución de la función de Calidad en la INSI

En sus inicios la Intendencia Nacional de Informática (INI) quien era el órgano encargado de proveer de aplicaciones y herramientas informáticas necesarias para apoyar la labor de las unidades organizacionales de la institución, no contaba con un equipo de trabajo responsable exclusivamente de las pruebas de los Sistemas; éstos por el contrario eran implantados en el ambiente de producción únicamente con las pruebas realizadas por los mismos programadores responsables del Sistema. Se hacen los esfuerzos para formar una coordinación encargada de resolver las Atenciones a Usuarios (ATUS) producto de los problemas originados por los sistemas que tenían errores, años mas tarde esta coordinación se convierte en la División de Atención a Usuarios con funciones que abarcaba además las pruebas de

los sistemas antes de su implantación, luego del cual se convertiría en la Gerencia de Homologación.

Pero es recién que a partir del 28 de Octubre del 2002 se realiza una reestructuración de toda la organización producto de la fusión SUNAT - Aduanas, mediante el cual se crea la actual Gerencia de Servicios a Usuarios bajo la Intendencia Nacional de Sistemas de Información (INSI). Responden a esta gerencia la División de Atención a Usuarios (DATUS) y la División de Control de Calidad (DCC).

La nueva Gerencia tendría el objetivo de garantizar la calidad de los servicios, mejorar el índice de errores detectados a los Sistemas y a la vez mejorar el servicio de atención a los Usuarios.

Sin embargo, esta tarea se torna muy complicada si no se cuenta con un esquema de solución integral, hoy en día la DCC está encargada principalmente de las pruebas de los sistemas; un gran avance se dio cuando en Mayo del 2005 se aprueba la Metodología de Desarrollo de Sistemas de Información MDSI y la Metodología de Gestión de Proyectos MGP ambas desarrolladas por la Oficina de Ingeniería de Procesos de Sistemas (OIPS)

El MDSI está en un proceso de mejoramiento continuo, actualmente se viene implementando los procedimientos específicos y suficientes que permitan gestionar y ejecutar con eficiencia el proceso de aseguramiento de la calidad de software en todas la etapas del desarrollo de los Sistemas.

1.2 Diagnóstico Estratégico

El ámbito de estudio concerniente al aseguramiento de la calidad de software hace que nuestro diagnóstico estratégico se concentre en la Intendencia Nacional de Sistemas de Información. Para ello el presente estudio se ha basado en dos técnicas : el análisis FODA y el Cuadro de Mando Integral de Kaplan y Norton.

1.2.1 Análisis FODA

El análisis Foda hace referencia a las fortalezas, debilidades, oportunidades y riesgos de la INSI . Estos 4 elementos suministran una lista básica de situaciones tanto internas como externas que darán soporte a la toma de decisiones.

1.2.1.1 Fortalezas y Debilidades

Fortalezas

- Personal con conocimiento del negocio tributario y aduanero.
- Personal en la DCC con experiencia y conocimiento en el desarrollo de software.
- Se dispone de recursos económicos suficientes para adquirir las mejores soluciones tecnológicas del mercado.
- Personal de INSI con conocimiento técnico altamente calificado.

Debilidades

- Sobrecarga de trabajo, expectativa por resultado a corto plazo y proyectos urgentes redundan en errores del software.
- Configuración del entorno para las pruebas diferente a los existentes en el entorno de producción. Cambios en la arquitectura y el entorno de producción no son comunicados oportunamente al área de pruebas.
- Demora en el trabajo de preparación del entorno antes de iniciar las pruebas. En la DCC existe un conocimiento limitado en la administración y soporte de servidores, gestión de base de datos, herramientas ETL y en general sobre las nuevas tendencias tecnológicas.
- No se dispone de herramientas adecuadas que permitan incrementar la productividad de las pruebas de software (Disminuir los tiempos en las pruebas y mejorar la identificación de errores).
- Falta de cultura organizacional sobre la calidad del SW. (Se generan conflictos en todos los niveles, programas sin pruebas unitarias, etc.)

- Incumplimiento del uso de la metodología MDSI.
- Errores detectados tardíamente. Los errores son detectados al final del desarrollo de software o son reportados por los usuarios una vez producidos, lo que redundará en mayores costos de corrección y mantenimiento.
- Lenguajes de programación sin estándares de programación.
- No existe en la DCC procedimientos formales para las pruebas.
- No se dispone de métricas adecuadas para medir y mejorar la calidad de software.
- El conocimiento en la INSI no está bien difundida en todas sus áreas que la conforman. (El conocimiento está muy disperso o no se dispone de herramientas para facilitar su búsqueda)

1.2.1.2 Oportunidades y Riesgos

Oportunidades

- Disponibilidad en el mercado de herramientas BTO para el proceso de aseguramiento de la calidad del software.
- Crecimiento sostenido de la recaudación y por ende mayor presupuesto asignado a la SUNAT para seguir disponiendo de recursos.

Amenazas

- Posibilidad latente de reducción de personal ante la situación de la fusión de SUNAT y Aduanas.
- Tendencia al reemplazo periódico de los programadores a plazo determinado, lo que aumenta el riesgo de errores por la inexperiencia y desconocimiento del nivel técnico y de negocio.
- Sueldos por debajo del mercado para los programadores a plazo determinado que promueven su permanencia por un corto plazo.
- Posibilidad de encontrar en el mercado herramientas BTO que no satisfagan las expectativas deseadas.

A continuación se muestra el cuadro de análisis FODA :

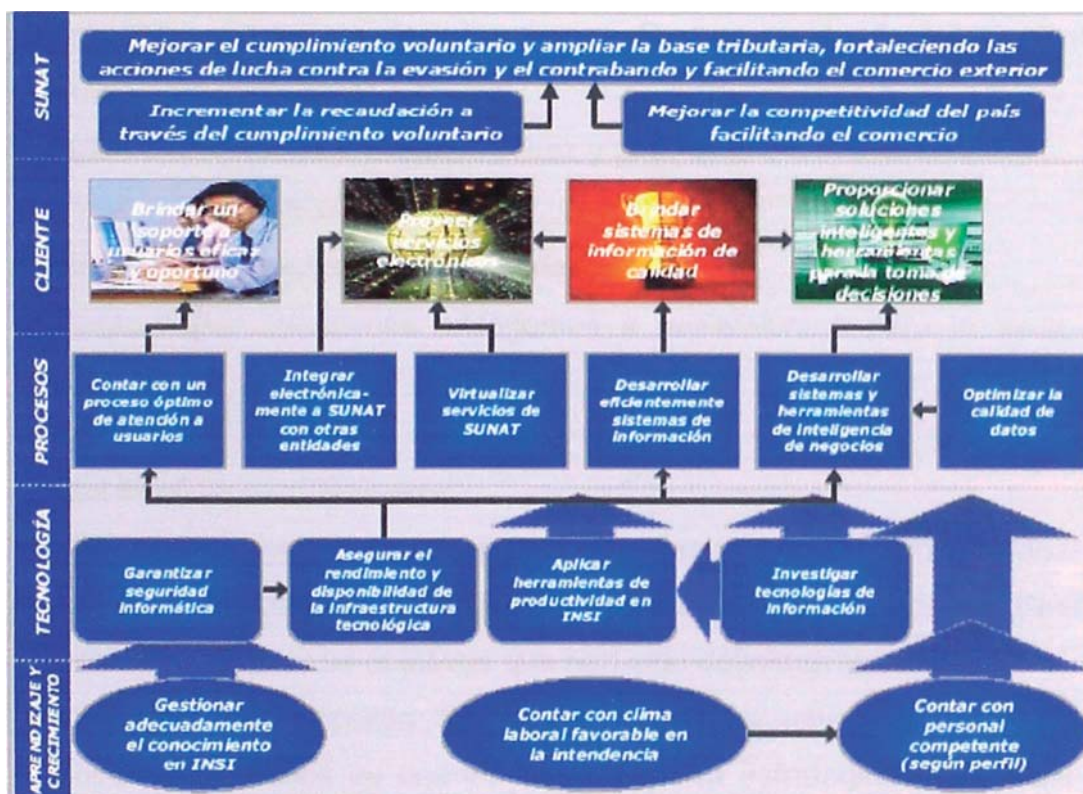
<p style="text-align: center;">FACTORES INTERNOS</p> <p style="text-align: center;">FACTORES EXTERNOS</p>	<p>Fortalezas</p> <p>F1. Personal de la DCC con experiencia en el desarrollo de SW.</p> <p>F2. Personal técnico de INSI altamente calificado.</p> <p>F3. Personal con conocimiento del negocio tributario y aduanero.</p> <p>F4. Se dispone de recursos económicos.</p>	<p>Debilidades</p> <p>D1. Sobrecarga de trabajo y expectativas a corto plazo para producir el SW</p> <p>D2. Configuración del entorno de pruebas diferente al existente en producción.</p> <p>D3. Demora en la preparación del entorno de pruebas</p> <p>D4. No se dispone de herramientas de productividad.</p> <p>D5. Falta de cultura organizacional de la calidad.</p> <p>D6. Incumplimiento del uso del MDSI</p> <p>D7. Errores detectados tardamente.</p> <p>D8. Sin estándares de programación.</p> <p>D9. En la DCC no existen procedimientos formales para las pruebas.</p> <p>D10. No se dispone de métricas para medir la Calidad de SW.</p> <p>D11. Conocimiento no difundido.</p>
<p>Oportunidades</p> <p>O1. Disponibilidad en el mercado de herramientas BTO.</p> <p>O2. Crecimiento sostenido de la recaudación, entonces mayor presupuesto para disponer de recursos.</p>	<p>o Aprovechar el conocimiento técnico y de negocios de la DCC, para minimizar el tiempo de aprendizaje de herramientas BTO (O1,O2,F1,F3)</p>	<p>o Adquirir herramientas de productividad. (O1,O2,D4,D7,D10)</p> <p>o Contratar talleres de inducción sobre calidad del SW para personal INSI y en todos los niveles (O2,D5,D6,D8,D9,D10)</p> <p>o Contratar cursos de capacitación en TIC (O2,D3).</p>
<p>Amenazas</p> <p>A1. Posibilidad latente de reducción de personal ante la situación de la fusión SUNAT y Aduanas.</p> <p>A2. Tendencia al reemplazo periódico del personal a plazo determinado.</p> <p>A3. Sueldos por debajo del mercado para dicho personal.</p> <p>A4. Herramientas BTO que no satisfagan las expectativas.</p>	<p>o Gestionar el conocimiento en la INSI (A1,A2,A3,F2,F3).</p> <p>o Identificar al personal de INSI potencialmente aprovechable en otras áreas (A1,A2,,A3,F3)</p>	<p>o Convocar a los líderes mundiales de herramientas BTO.(A4,D4)</p>

1.2.2 Cuadro de Mando Integral de Kaplan y Norton

Se trata de un conjunto de medidas que proporcionan una visión rápida y completa de ciertos indicadores divididos en 4 perspectivas : Financiera, Clientes, Procesos Internos, y Aprendizaje y Crecimiento. Para el caso específico de la INSI, se ha adaptado este modelo a las necesidades de la Intendencia, definiéndose las siguientes perspectivas

- **SUNAT**, que es una adaptación a la perspectiva financiera tradicional dada la condición de la INSI de área de apoyo dentro de la institución.
- **CLIENTE**, relacionado a la identificación de los clientes de la INSI y de sus necesidades.
- **PROCESOS INTERNOS**, referidos a los procesos en los cuales la INSI debe alcanzar la excelencia;
- **TECNOLÓGICA**, una perspectiva adicional al modelo tradicional recomendada para organizaciones que utilizan intensivamente tecnologías de información.
- **APRENDIZAJE y CRECIMIENTO**, relacionado a la formación del recurso humano para asegurar el crecimiento a largo plazo.

Todas estas estrategias se encuentran graficadas en la siguiente ilustración



A continuación se describe cada una de las perspectivas así como los objetivos que la conforman :

1.2.2.1 Perspectiva SUNAT

Esta perspectiva permite visualizar el fin último de las actividades que se realizan al interior de la Intendencia. Los objetivos agrupados en esta perspectiva permite conectar a los objetivos internos de la INSI con la SUNAT.

Esta perspectiva muestra el objetivo general de la institución establecido en el plan estratégico institucional 2004-2006: “Mejorar el cumplimiento voluntario y ampliar la base tributaria, fortaleciendo las acciones de lucha contra la evasión y el contrabando y facilitando el comercio exterior”; y los siguientes objetivos planteados :

- Incrementar la recaudación a través del cumplimiento voluntario.
- Mejorar la competitividad del país mejorando el comercio.

1.2.2.2 Perspectiva Cliente

La perspectiva cliente tiene un enfoque basado en la eficacia y se traduce en cumplir las expectativas de los usuarios. Se han planteado los siguientes objetivos a nivel cliente

- Brindar un soporte a usuarios eficaz y oportuno.
- Proveer servicios electrónicos.
- Brindar sistemas de información de calidad.
- Proporcionar soluciones inteligentes y herramientas para la toma de decisiones.

1.2.2.3 Perspectiva Procesos

Los objetivos agrupados en la perspectiva procesos permiten gestionar la eficiencia al interior de la INSI para satisfacer la perspectiva SUNAT y perspectiva Cliente. Los objetivos que han sido definidos son :

- Contar con un proceso óptimo de atención a usuarios, dado que el soporte a usuarios es una de las principales estrategias de la INSI, se mejorará el servicio de atención para contar con un sistema uniforme, ordenado y sobre todo acordado con los usuarios.
- Integrar electrónicamente a la SUNAT con otras entidades, ya que muchos de los servicios que se brinda requiere de procesos con otras entidades externas.
- Virtualizar los servicios de SUNAT, comprende que tanto los procesos de negocio como de apoyo sean accesibles en forma virtual.
- Desarrollar eficientemente sistemas de información para alcanzar una óptima calidad en la elaboración de software y su documentación asociada, incluyéndose tanto el desarrollo interno como la tercerización de los sistemas de información.
- Desarrollar sistemas y herramientas de inteligencia de negocios, se incorporarán sistemas inteligentes que tomen decisiones efectivas en forma automática y se pondrán a disposición de los usuarios consultas de fácil acceso por temas.

- Optimizar la calidad de datos para poder tomar decisiones acertadas.

1.2.2.4 Perspectiva Tecnológica

La perspectiva tecnológica agrupa los objetivos relacionados a la estructura informático-tecnológica, la cual debe ser integrada y segura; así mismo, contar con herramientas de productividad e investigar tecnologías de información. Se tienen los siguientes objetivos

- Garantizar la seguridad informática
- Asegurar el rendimiento y disponibilidad de la infraestructura tecnológica.
- Aplicar herramientas de productividad en la INSI considerando que estas permiten incrementar la productividad, así como garantizar productos o servicios de calidad.
- Investigar tecnologías de información que permitirán identificar nuevas herramientas de productividad , incorporar nuevas técnicas de seguridad informática y mejorar la infraestructura tecnológica.

1.2.2.5 Perspectiva Aprendizaje y Crecimiento

Esta perspectiva agrupa los objetivos vinculados al personal, los objetivos planteados son

- Gestionar adecuadamente el conocimiento INSI, a través de este objetivo se implantará un procesos de gestión del conocimiento.
- Contar con un clima laboral favorable en la Intendencia, este objetivo se orienta a fomentar el trabajo en equipo, donde la coordinación y comunicación sean permanentes en todos los niveles dentro y fuera de la INSI.
- Contar con personal competente para alcanzar los objetivos de las perspectivas superiores, se incluyen planes de capacitación en temas técnicos y de negocio, desarrollo de competencias, fortalecer la responsabilidad de principio a fin, implementar una línea de carrera y un sistema de evaluación y reconocimientos.

1.3 Diagnóstico Funcional

1.3.1 Productos

El principal producto de la INSI son los sistemas de información que provee a la institución y a otras entidades externas.

En el Anexo 1 y Anexo 2 se muestra la relación de sistemas clasificados por temas y según correspondan a Sistemas de Tributos Internos (SUNAT) o a los Sistemas de Tributos Aduaneros respectivamente.

1.3.2 Clientes

Los usuarios de la SUNAT pueden clasificarse en usuarios internos y usuarios externos, en donde la INSI cumple la función de proveerles los sistemas y la información necesaria para el cumplimiento de sus obligaciones, sistemas de recaudación para entidades externas y sistemas que brinden información para cruces según convenios específicos.

Los usuarios internos de la SUNAT serán aquellos pertenecientes a la Intendencias Regionales y zonales, la Intendencia Nacional de principales contribuyentes y a los órganos de línea de la Superintendencia Nacional Adjunta de Aduanas.

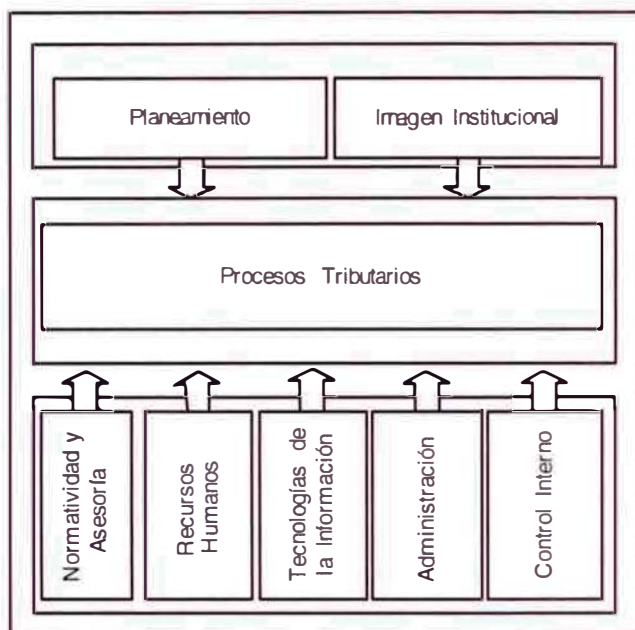
Los usuarios externos la comprenden los contribuyentes que usan los sistemas implementados por la SUNAT para el cumplimiento de sus obligaciones tributarias y las entidades externas tales como ESSALUD, ONP, SENCICO, Gobiernos Regionales y las Municipalidades para quienes la SUNAT recauda algunos tributos para los cuales le ha sido expresamente delegada la función de recaudación.

1.3.3 Proveedores

Los proveedores de la INSI son las empresas que proveen de tecnologías de información, tanto de software como de la arquitectura informática además de los sistemas que se vienen tercerizando para ponerlas a disposición de todas las unidades organizacionales y facilitarles el cumplimiento de sus objetivos.

1.3.4 Procesos

En el siguiente gráfico se pueden apreciar los grandes procesos de la SUNAT, donde se puede apreciar la tecnología de la información como una de los procesos de apoyo que la INSI provee a la institución.



Brevemente se detallan los procesos de la SUNAT que se han clasificado en Procesos Estratégicos, Tributarios y de Apoyo.

Procesos Estratégicos:

Planeamiento.- Elaborar el Plan Institucional e implementar los mecanismos para la formulación y evaluación de proyectos, retroalimentando con los resultados a la Alta Dirección.

Imagen Institucional.- Diseñar y mantener la imagen institucional requerida por la alta dirección, tanto interna como externamente.

Procesos de Apoyo:

Normatividad y Asesoría.- Proveer el marco normativo, es decir, normas, procedimientos y definiciones en el ámbito de los negocios y en los demás procesos.

Recursos Humanos.- Proveer a la institución –en los puestos que lo requieran- de personal idóneo para el cumplimiento de sus funciones.

Impulsar el desarrollo del personal hacia un excelente desempeño y contribuir con su bienestar.

Tecnologías de Información.- Proveer a la institución de sistemas de información, así como de la plataforma informática para el cumplimiento de sus funciones.

Siendo los productos de la INSI los sistemas de información, los principales procesos para su desarrollo están basadas en la metodología MDSI las cuales se representan en la siguiente ilustración



El proceso de desarrollo involucra un conjunto de actividades, métodos y prácticas utilizadas para desarrollar o mantener los sistemas de información, la búsqueda de un producto de calidad sólo es posible si su proceso es de calidad, en el apéndice 2.4 se explican en detalle cada uno de los 5 procesos.

Administración.- Proveer a la institución de los recursos logísticos y financieros para el cumplimiento de sus funciones, asimismo supervisar la ejecución del plan institucional.

Control Interno.- Proveer a la institución de los elementos de control para asegurar la adecuada aplicación de las normas, tanto para los procesos de negocio como para sus demás procesos.

Procesos Tributarios:

Procesos Tributarios.- Son los procesos orientados a recaudar y fomentar el cumplimiento del pago de los tributos administrados por la SUNAT tanto para los Macroprocesos de negocio tributario como para los Macroprocesos de negocio Aduanero. En la siguiente figura se ilustran los Macroprocesos mencionados:

Macroprocesos del Negocio Tributario	Macroprocesos del Negocio Aduanero
Registro	Registro de Operadores
Servicios al Contribuyente	Despacho Aduanero
Declaración y Pago	Ingreso/Salida de Mercancías
Determinación de Obligaciones	Control de Mercancías, Abandono y Comiso
Control de la Deuda	Control de Pago
Aplazamiento y/o Fraccionamiento	Cobranza Administrativa
Cobranza Coactiva	Administración de Garantías
Recursos Impugnatorios	Aplazamiento y/o Fraccionamiento
Programación	Cobranza Coactiva
Acciones de Auditoría	Recursos Impugnatorios
Operativos Masivos	Programación
Devoluciones	Control Posterior Despacho y Operaciones
	Prevención y Control del Contrabando
	Devoluciones

Macroprocesos Tributarios y Aduaneros

1.3.5 Organización de la SUNAT

La estructura organizacional de la SUNAT, considera en la Alta Dirección al Superintendente Nacional de Administración Tributaria, Superintendente Nacional Adjunto de Aduanas y al Superintendente Adjunto de Tributos Internos.

Estructura Orgánica:

ALTA DIRECCIÓN

Superintendencia Nacional de Administración Tributaria
Superintendencia Nacional Adjunta de Tributos Internos
Superintendencia Nacional Adjunta de Aduanas

COMITÉ DE ALTA DIRECCIÓN

ÓRGANO DE CONTROL:

Oficina de Control Interno

ÓRGANOS DE APOYO:

Secretaría General
Instituto de Administración Tributaria y Aduanera

ÓRGANOS DE LÍNEA DEPENDIENTES DE LA SUPERINTENDENCIA NACIONAL ADJUNTA DE TRIBUTOS INTERNOS

Intendencia de Principales Contribuyentes Nacionales
Intendencia Regional Lima
Intendencias Regionales (desconcentradas)
Oficinas Zonales (desconcentradas)

ÓRGANOS DE LÍNEA DEPENDIENTES DE LA SUPERINTENDENCIA NACIONAL ADJUNTA DE ADUANA

Intendencia de Fiscalización y Gestión Recaudación Aduanera
Intendencia de Fiscalización y Gestión Recaudación Aduanera
Intendencia de Aduana Marítima del Callao
Intendencia de Aduana Aérea del Callao
Intendencia de Aduana Postal del Callao
Intendencias de Aduanas (desconcentradas)

ÓRGANOS DE SOPORTE

Intendencia Nacional de Administración
Intendencia Nacional de Cumplimiento Tributario
Intendencia Nacional de Estudios Tributarios y Planeamiento
Intendencia Nacional de Recursos Humanos
Intendencia Nacional de Servicios al Contribuyente
Intendencia Nacional de Sistemas de Información
Intendencia Nacional de Técnica Aduanera
Intendencia Nacional Jurídica

1.3.6 Organización de la INSI

La Intendencia Nacional de Sistemas de información tiene la siguiente organización y funciones

1.3.6.1 Gerencia de Servicios a Usuarios

Funciones

- a) Garantizar la calidad de los servicios a ser brindados por la Intendencia Nacional de Sistemas de Información.
- b) Implementar y capacitar los sistemas desarrollados, autorizados por la Intendencia Nacional de Sistemas de Información.
- c) Evaluar la satisfacción de los usuarios de los servicios brindados por la Intendencia Nacional de Sistemas de Información.
- d) Brindar soporte a los usuarios de los servicios brindados por la Intendencia Nacional de Sistemas de Información.
- e) Proponer modificaciones a los procesos de SUNAT a través del empleo intensivo de tecnologías de información.

1.3.6.2 Gerencia de Desarrollo de Sistemas Tributarios y de Gestión

Funciones

- a) Efectuar el análisis funcional, diseño, construcción y mantenimiento de los sistemas de información que desarrolle la Intendencia Nacional de Sistemas

de Información relacionados a los temas de tributos internos, administración e información de gestión.

b) Liderar el proceso de tercerización para el desarrollo de sistemas de información.

c) Garantizar la calidad de los sistemas de información que desarrolle, a través de la satisfacción de los requerimientos funcionales, optimización de los recursos informáticos y minimización de errores de los productos de su competencia a implementar.

d) Apoyar en las pruebas, implementación y capacitación de los sistemas de información de su competencia.

e) Administrar los requerimientos informáticos de las áreas usuarias de la Institución.

f) Proponer la elaboración de normas legales y modificaciones concernientes a los sistemas de información de su competencia.

g) Elaborar proyectos de resolución, directivas y circulares a fin de garantizar la implementación de los sistemas de información de su competencia.

h) Proponer a la Intendencia Nacional de Estudios Tributarios y Planeamiento los lineamientos técnicos o a ser considerados en los convenios que suscriba SUNAT con otros organismos en el ámbito de su competencia.

i) Proponer modificaciones a los procesos de SUNAT a través del empleo intensivo de tecnologías de información.

j) Promover y apoyar la investigación relacionada a tecnologías de desarrollo de sistemas de información.

1.3.6.3 Gerencia de Desarrollo de Sistemas Aduaneros

Funciones

a) Efectuar el análisis funcional, diseño, construcción y mantenimiento de los sistemas de información que desarrolle la Intendencia Nacional de Sistemas de Información relacionados a los temas aduaneros.

b) Garantizar la calidad de los sistemas de información que desarrolle, a través de la satisfacción de los requerimientos funcionales, optimización de

los recursos informáticos y minimización de errores de los productos de su competencia a implementar.

c) Apoyar en las pruebas, implementación y capacitación de los sistemas de información de su competencia.

d) Proponer la elaboración de normas legales y modificaciones concernientes a los sistemas de información de su competencia.

e) Elaborar proyectos de resolución, directivas y circulares a fin de garantizar la implementación de los sistemas de información de su competencia.

f) Proponer a la Intendencia Nacional de Estudios Tributarios y Planeamiento los lineamientos técnicos a ser considerados en los convenios que suscriba SUNAT con otros organismos, en el ámbito de su competencia.

g) Proponer modificaciones a los procesos de SUNAT a través del empleo intensivo de tecnologías de información.

h) Promover y apoyar la investigación relacionada a tecnologías de desarrollo de sistemas de información.

1.3.6.4 Gerencia de Producción

Funciones

a) Garantizar la disponibilidad de los sistemas y bases de datos implementados, en concordancia con las normas de seguridad establecidas.

b) Garantizar el adecuado funcionamiento de la infraestructura informática y de telecomunicaciones de la SUNAT.

c) Proponer proyectos de mejoramiento de infraestructura informática.

d) Proponer modificaciones a los procesos de SUNAT a través del empleo intensivo de tecnologías de información.

e) Promover y apoyar la investigación relacionada a tecnologías de infraestructura informática y de telecomunicaciones.

1.3.6.5 Oficina de Seguridad Informática

Funciones

a) Establecer y supervisar el cumplimiento de políticas, procedimientos y normas en materia de seguridad informática.

- b) Efectuar permanente análisis de riesgo y pruebas de vulnerabilidad de los activos informáticos.
- c) Fomentar y apoyar la investigación y análisis de tecnología requerida en materia de seguridad informática.
- d) Normar los esquemas de seguridad de los procesos, aplicaciones e infraestructura informática y de telecomunicaciones a ser implementados por la Intendencia Nacional de Sistemas de Información.
- e) Apoyar en la implementación de soluciones integrales de seguridad informática.

1.3.6.6 Oficina de Investigación Tecnológica

Funciones

- a) Realizar las actividades de investigación y análisis de las tecnologías de información existentes y emergentes, así como de la utilidad e impacto que tendría su implementación en la Institución.
- b) Seleccionar las tecnologías, productos y herramientas informáticas para el proceso de desarrollo de sistemas de información.
- c) Apoyar y centralizar los requerimientos de adquisición de la Intendencia Nacional de Sistemas de Información.
- d) Elaborar las especificaciones técnicas para la adquisición de las tecnologías e infraestructura informática y de telecomunicaciones requeridas.
- e) Administrar el proceso de incorporación del personal de la Intendencia de Sistemas de Información en el empleo de las tecnologías seleccionadas para SUNAT.
- f) Centralizar y administrar los requerimientos de capacitación del personal de la Intendencia Nacional de sistemas de Información.
- g) Normar los lineamientos de arquitectura de aplicaciones e infraestructura informática y de telecomunicaciones a ser empleada para el desarrollo de sistemas de información.

1.3.6.7 Oficina de Planeamiento y Estrategias de Servicios Electrónicos

Funciones

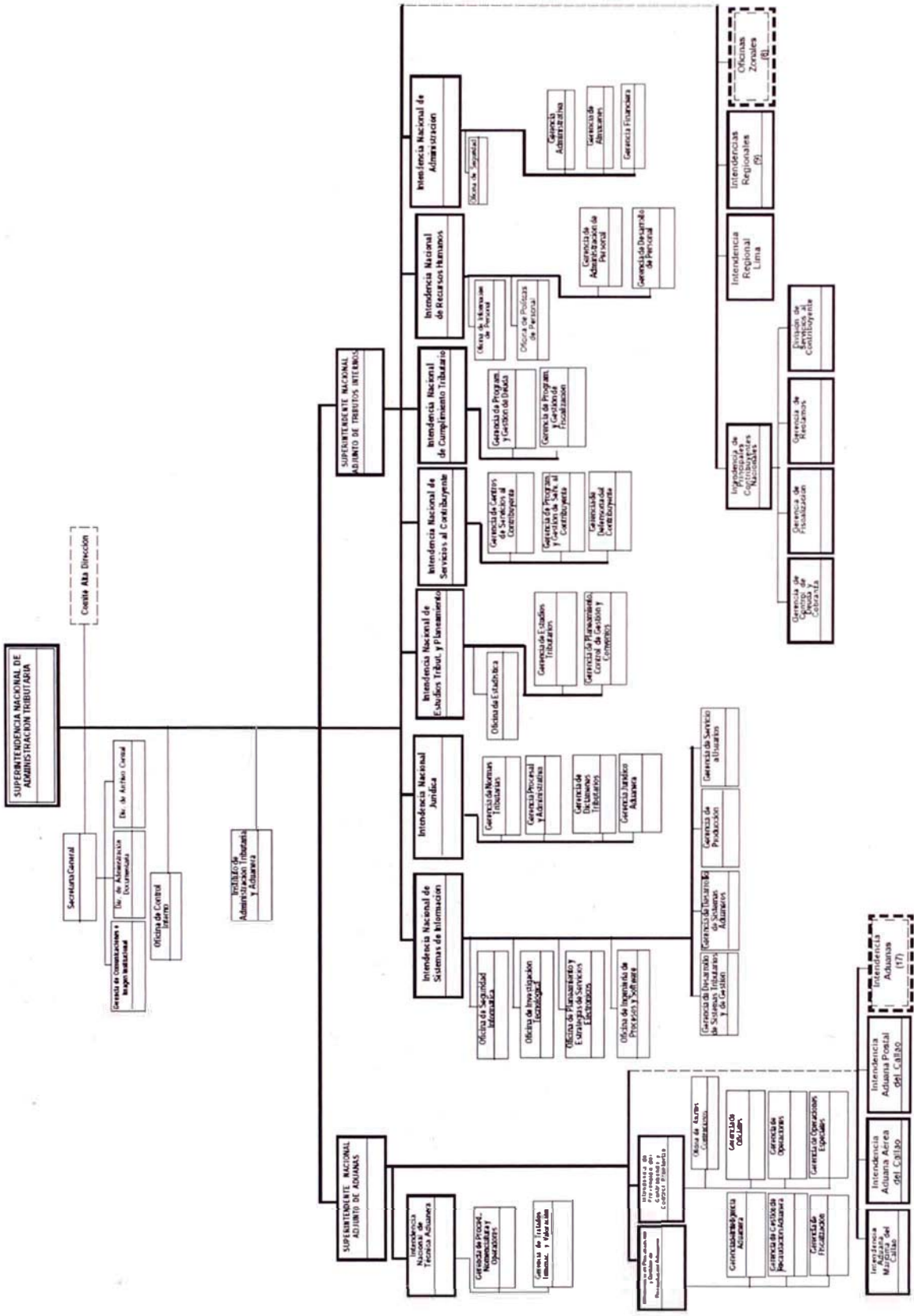
- a) Asesorar a la SUNAT para el desarrollo de gobierno electrónico.
- b) Proponer la estrategia de SUNAT para el desarrollo de gobierno electrónico.
- c) Establecer los lineamientos que permitan la adecuación de los procesos a las políticas de gobierno electrónico que se establezcan a nivel de Gobierno Nacional.
- d) Coordinar los esfuerzos de gobierno electrónico con las diferentes áreas de SUNAT.
- e) Participar en la implementación de gobierno electrónico en SUNAT.
- f) Evaluar el éxito de la implementación de gobierno electrónico y sugerir medidas de corrección u optimización en el proceso.

1.3.6.8 Oficina de Ingeniería de Procesos y Software

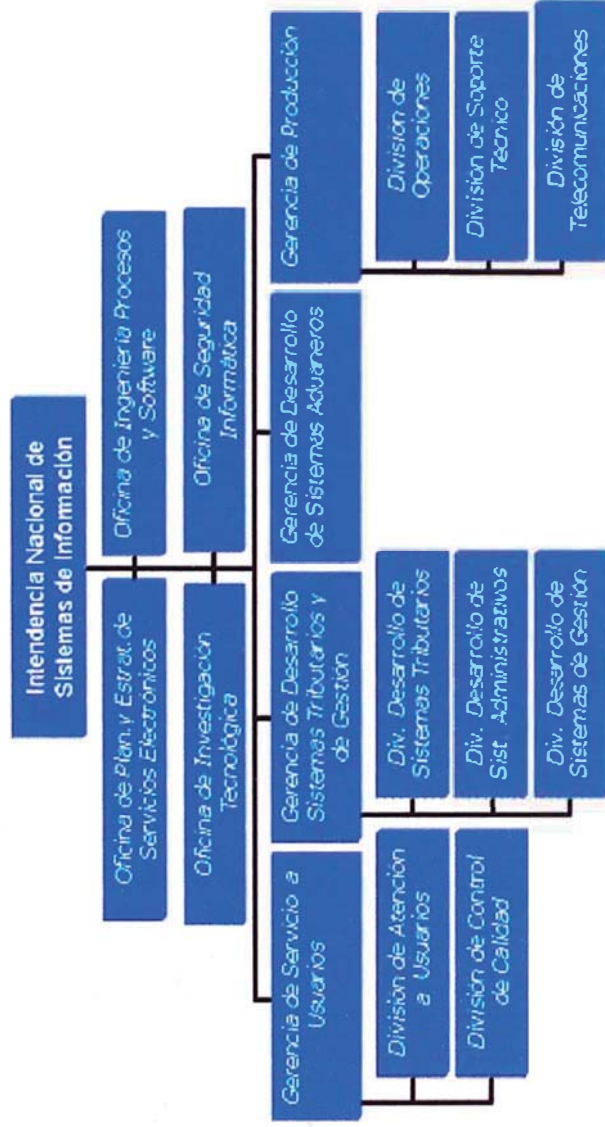
Funciones

- a) Actualizar la metodología de desarrollo e implementación de sistemas de información.
- b) Administrar la arquitectura de procesos, datos y de aplicaciones de la SUNAT, así como las versiones de los mismos.
- c) Establecer los estándares de documentación de las diversas etapas de desarrollo de los sistemas de información.
- d) Normar y actualizar los estándares de documentación técnica interna de la Intendencia Nacional de Sistemas de Información.
- e) Supervisar el cumplimiento de la metodología de desarrollo de sistemas de información, así como el empleo de los estándares establecidos.
- f) Difundir y actualizar el Plan Maestro de la INSI.
- g) Establecer las métricas de las diversas etapas y tareas establecidas en la metodología de desarrollo de sistemas de información

A continuación se muestra el organigrama de la SUNAT y de la INSI respectivamente



ORGANIGRAMA INSI



Estructura Organizacional de las Intendencia Nacional de Sistemas de Información

Fuente: Diario El Peruano 28/10/2002

CAPÍTULO II

MARCO TEORICO

Para el presente estudio es necesario definir algunos conceptos generales, como el aseguramiento de la calidad de software, la Norma ISO 9126 y la Metodología de Desarrollo de Sistemas MDSI, éste último como parte esencial del proceso del desarrollo de software y sobre el cual se pondrán los hitos de control para asegurar su calidad. Un producto será de calidad si el proceso para desarrollarlo también lo es, en este proceso también es importante definir el concepto de herramientas de productividad como un elemento importante para el apoyo a las pruebas.

2.1 Definiciones Generales de Calidad

Definición de Calidad

La calidad se puede definir como "una característica o atributo de una cosa". De esta forma se podría decir que la calidad de los productos puede medirse como una comparación de sus características y atributos. Así, este concepto puede aplicarse a cualquier producto.

La calidad se define como el "grado en que un conjunto de características inherentes que cumple con unos requisitos" [ISO 9000:2000].

Definición de Calidad de Software

Es el grado de cumplimiento (concordancia) con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo

explícitamente documentados, y con las características implícitas que se espera de todo software desarrollado profesionalmente. Existen productos de software los cuales deseamos conocer y deseamos poder controlar y mejorar. Algunos de las razones para medir el software son las siguientes :

- Para conocer la calidad del Producto
- Para evaluar la productividad de la personas que la producen
- Para evaluar los beneficios de nuevas herramientas de software
- Para formar una base a fin de estimar

Métricas

Las métricas son un buen medio para entender, monitorizar, controlar, predecir y probar el desarrollo software y los proyectos de mantenimiento.

Es una forma de medir (método de medición, función de cálculo o modelo de análisis) y una escala (conjunto de números naturales, grados centígrados, etc.), definidas para realizar mediciones de uno o varios atributos .

Ejemplos

La métrica “líneas de código” puede ser definida para realizar mediciones del “tamaño” de un “módulo en C” y para realizar mediciones del “tamaño” de un “programa en Ada”.

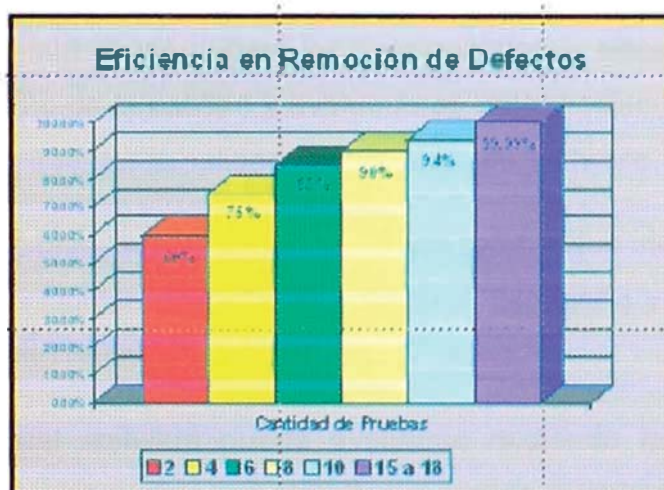
Medición

Proceso por el cual se asignan números o símbolos a atributos a entidades del mundo real , de tal forma que se describan de acuerdo a reglas claramente definidas.

Eficiencia en eliminación de errores

Indica el % de remoción de defectos según la cantidad de pruebas distintas que se realice. Según Caper Jones (www.spr.com) parte en 60% de remoción (eficiencia) con solo 2 pruebas, hasta un 99.99% de eficiencia con

15 o más pruebas. El promedio en USA es de 85% de eficiencia en la remoción de defectos con 6 pruebas. Actualmente el promedio de eficiencia en la eliminación de defectos en SUNAT es de 33%.



Eficiencia de eliminación de defectos

2.2 Definición de Aseguramiento de la Calidad

Son todas aquellas acciones planificadas y sistemáticas necesarias para proporcionar la confianza adecuada de que un producto o servicio satisface los requisitos de calidad establecidos (ISO 8402).

El aseguramiento de la Calidad en la INSI implica un proceso continuo que garantiza y evalúa la calidad del software a través del ciclo de vida del sistema.

2.3 Normas ISO

2.3.1 Normas ISO

ISO (La Organización Mundial de Normalización) es una federación mundial de organismos nacionales de normalización (organismos miembros de la ISO).

La norma, es un conjunto de disposiciones especificaciones y procedimientos. Las Normas ISO son un conjunto de normas internacionales genéricas que establecen sistemas de gestión de la calidad aplicados por organizaciones de cualquier tipo o tamaño que fabrican productos o

componentes (hardware), fabrican software, fabrican materiales procesados, ofrecen servicios, desempeñan funciones de administración pública.

2.3.2 Normas ISO 9126

Esta norma está pensado para los desarrolladores, adquirentes, personal de aseguramiento de la calidad y evaluadores independientes, responsables de especificar y evaluar la calidad del producto de software.

Por tanto, puede servir para validar la completitud de una definición de requisitos de calidad de software, objetivos de diseño y prueba, criterios de aseguramiento de la calidad, etc.

La calidad del software puede evaluarse midiendo los atributos internos (visualización del código) o atributos externos (comportamiento del código cuando se ejecuta).

Según la ISO 9126 la calidad se mide de acuerdo a 6 criterios :

Funcionalidad: capacidad del producto software para proporcionar funciones que satisfagan las necesidades especificadas e implícitas.

Fiabilidad: capacidad del producto software para mantener un nivel especificado de rendimiento.

Usabilidad: la capacidad del producto software de ser entendido, aprendido, utilizado y atractivo al usuario.

Eficiencia: la capacidad del producto software para proporcionar el rendimiento apropiado, relativo a la cantidad de recursos utilizados.

Mantenibilidad: la capacidad del producto software para ser modificado. Las modificaciones pueden incluir correcciones, mejoras o adaptación del software a cambios en el entorno, en los requisitos o en las especificaciones funcionales.

Portabilidad: la capacidad del producto software de ser transferido de un entorno a otro.

2.3.2.1 Funcionalidad

La **funcionalidad** se subdivide en cinco subcaracterísticas:

Adecuación: la capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas específicas y objetivos de los usuarios.

Exactitud: la capacidad del producto software para proporcionar los resultados o efectos correctos y con el grado de precisión acordado.

Interoperabilidad: la capacidad del producto software para interactuar con uno o más sistemas especificados.

Seguridad: referido a la capacidad del producto software para proteger la información y los datos.

Conformidad: la capacidad del producto software para adaptarse a los estándares, convenciones o regulaciones en leyes y prescripciones relativos a la funcionalidad.

2.3.2.2 Fiabilidad

La **fiabilidad** se subdivide en cuatro subcaracterísticas:

Madurez: la capacidad del producto software para evitar fallos provocados por errores en el software.

Tolerancia a fallos: la capacidad del producto software para mantener un nivel de rendimiento determinado en caso de defectos en el software o incumplimiento de su interfaz.

Recuperabilidad: la capacidad del producto software para restablecer un determinado nivel de rendimiento y recuperar los datos afectados directamente en caso de ocurrir un fallo.

Conformidad: la capacidad del producto software para adaptarse a estándares, convenciones y regulaciones referidas a la fiabilidad.

2.3.2.3 Usabilidad

La **usabilidad** se subdivide en cinco subcaracterísticas:

Comprensibilidad: la capacidad del producto software para permitir al usuario que entienda si el software es adecuado, y como debe utilizarse para determinadas tareas y bajo ciertas condiciones de uso.

Facilidad de aprendizaje: la capacidad del producto software para permitir al usuario aprender su aplicación.

Operabilidad: la capacidad del producto software para permitir que el usuario lo opere y lo controle.

Atracción: la capacidad del producto software para atraer al usuario.

Conformidad: la capacidad del producto software para adaptarse a estándares, convenciones, guías de estilo y regulaciones relacionadas con la usabilidad.

2.3.2.4 Eficiencia

La **eficiencia** se subdivide en tres subcaracterísticas:

Comportamiento temporal: la capacidad del producto software para proporcionar tiempos de respuesta y de procesamiento apropiados cuando realiza sus funciones bajo condiciones determinadas.

Utilización de recursos: la capacidad del producto software para utilizar cantidades y tipos de recursos apropiados cuando el software realiza su función bajo determinadas condiciones.

Conformidad: la capacidad del producto software para adaptarse a estándares o convenciones relacionadas con la eficiencia.

2.3.2.5 Mantenibilidad

La **mantenibilidad** se subdivide en cinco subcaracterísticas:

Analizabilidad: Capacidad del producto software de diagnosticar sus deficiencias o causas de fallos, o de identificar las partes que deben ser modificadas.

Cambiabilidad: Capacidad del producto software de permitir implementar una modificación especificada. La implementación incluye los cambios en el diseño, el código y la documentación.

Estabilidad: Capacidad del producto software de evitar los efectos inesperados de las modificaciones.

Facilidad de prueba: Capacidad del producto software de permitir validar las partes modificadas.

Conformidad: Capacidad del producto software de cumplir los estándares o convenciones relativas a la mantenibilidad.

2.3.2.6 Portabilidad

La **portabilidad** se subdivide en cinco subcaracterísticas:

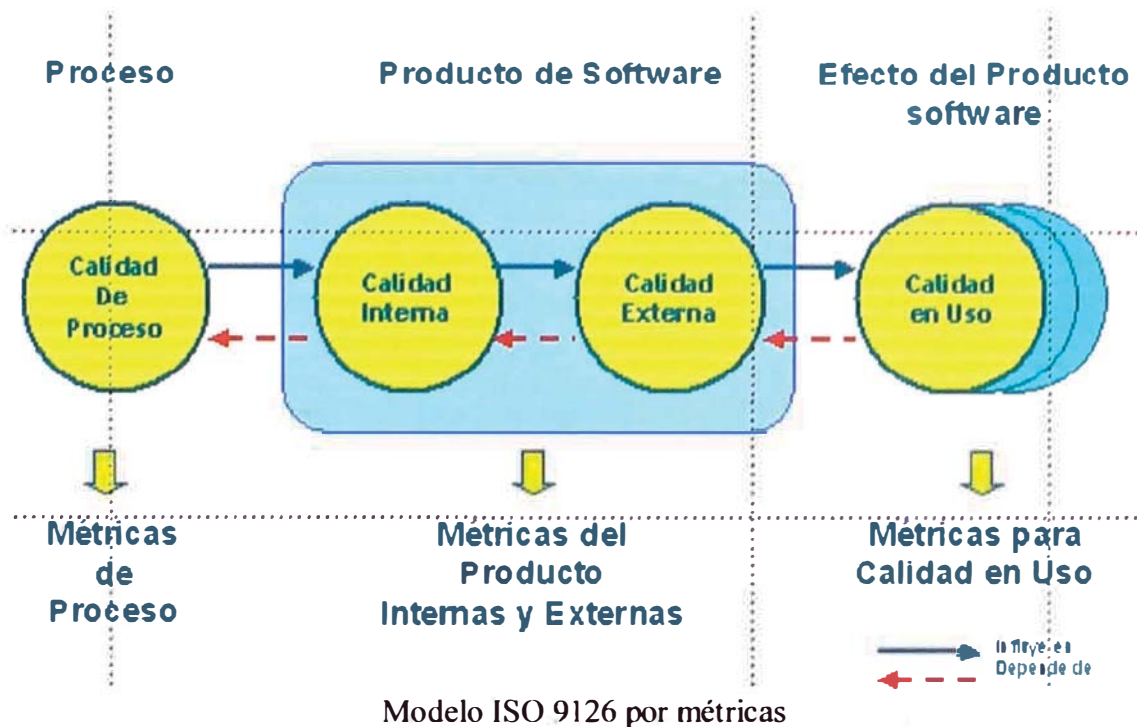
Adaptabilidad: la capacidad del producto software para ser adaptado para ambientes determinados sin realizar acciones o aplicar medios, más que los proporcionados para este propósito para el software considerado.

Facilidad de instalación: la capacidad del producto software para ser instalado en un ambiente determinado.

Coexistencia: la capacidad del producto software para coexistir con otro software independiente en un ambiente común compartiendo recursos.

Reemplazabilidad: la capacidad del producto software para ser utilizado en lugar de otro producto de software para el mismo propósito en el mismo ambiente.

Conformidad: la capacidad del producto software para adaptarse a estándares relacionados con la portabilidad.



2.4 Metodología de Desarrollo de Sistemas MDSI

La MDSI es la metodología de desarrollo de sistemas de información basada en la metodología española Métrica y RUP (Racional Unified Process); ha

sido adaptada a las necesidades de la SUNAT por la Oficina de Ingeniería de Procesos de Software (OIPS) en base a los métodos de desarrollo mas extendidos en la actualidad, así como los últimos estándares de ingeniería de software y calidad.

MDSI versión 2.0 contiene todas las actividades y tareas que se deben llevar a cabo para desarrollar un sistema (producto software) cubriendo las siguientes etapas

2.4.1 Modelamiento del Negocio (MPN)

El objetivo de esta fase es iniciar el modelamiento visual del proceso de negocio propuesto, utilizando para ello los diagramas de caso de uso de negocio y los diagramas de actividades de acuerdo a un estándar establecido. Las siguientes son las actividades que comprende el modelamiento del negocio

- Identificación de los Procesos de Negocio (MPN 1.1)
- Identificación de los Actores del Entorno del Negocio (MPN 1.2)
- Descripción de los Casos de Uso del Negocio (MPN 1.3)
- Especificación de las Reglas del Negocio (MPN 1.4)
- Especificación de Necesidades Externas a la INSI (MPN 1.5)

2.4.2 Modelamiento de Requerimientos (ASI)

El objetivo de esta fase es obtener una especificación detallada del sistemas de información que satisfaga las necesidades de información de los usuarios. Las siguientes son las actividades que comprende el modelamiento de Requerimientos :

Modelamiento de Requerimientos del Sistemas e Información (ASI 1)

- Determinación del alcance del Sistema (ASI 1.1)
- Obtención de Requerimientos (ASI 1.2)
- Obtención de los Casos de Uso del Sistema (ASI 1.3)

- Determinación de los Subsistemas de Análisis (ASI 1.4)
- Especificación de la Interface de Usuario (ASI 1.5)
- Identificación de Perfiles y Diálogos (ASI 1.6)
- Especificación del Comportamiento Dinámico de la Interface (ASI 1.7)
- Especificación de Formatos de Impresión (ASI 1.8)

Análisis Orientado a Objetos (ASI 2, ASI 3 y ASI 4)

- Análisis de los Casos de Uso (ASI 2)
- Análisis de Clases (ASI 3)
- Análisis de Paquetes (ASI 4)

Especificación de Interfaces con otros sistemas (ASI 7)

Especificación de las necesidades de Migración de datos y carga inicial (ASI 8)

Análisis de consistencia y especificación de Requerimientos (ASI 9)

- Verificación de la Calidad Técnica de los Modelos (ASI 9.1)
- Análisis de Consistencia entre Modelos (ASI 9.2)

2.4.3 Modelamiento de Tecnología (DSI)

El objetivo de la fase de Modelamiento de Tecnología es la definición de arquitectura del sistema de información y del entorno tecnológico, junto con la especificación detallada de los componentes del sistema de información. A partir de dicha información, se generan todas las especificaciones de construcción relativas al propio sistema de información. A continuación se especifican las actividades relacionadas al modelamiento de tecnología :

Definición de la Arquitectura del Sistema (DSI 1)

Diseño de la Arquitectura de Soporte (DSI 2)

Diseño Orientado a Objetos (DSI 3 y DSI 4)

- Diseño de Casos de Uso Reales (DSI 3)
- Diseño de Clases (DSI 4)

Generación de Especificaciones de Construcción (DSI 8)

- Especificación del Entorno de Construcción (DSI 8.1)
- Descripción de Subsistemas de Construcción (Gráfico Diagrama de Componentes y de Despliegue) (DSI 8.2)
- Especificación Detallada de Componentes (DSI 8.3)

Diseño de la Migración y Carga Inicial de Datos (DSI 9)

Requerimientos de Implantación (DSI 10)

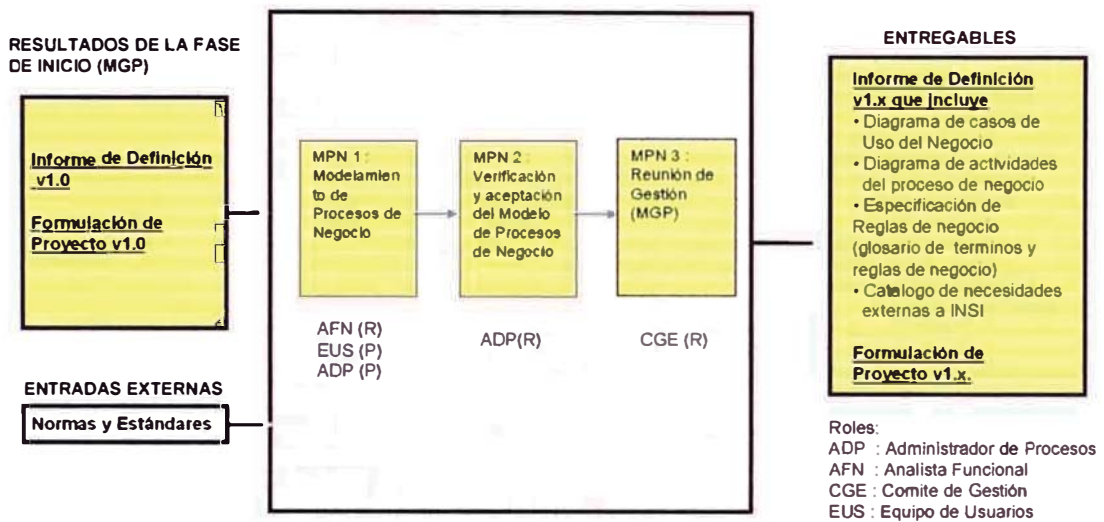
2.4.4 Construcción y Pruebas (CPS)

EL objetivo de ésta fase es generar el código de los componentes del sistema, desarrollar los procedimientos y documentación a fin de asegurar el correcto funcionamiento del sistema y desarrollar las pruebas respectivas para asegurar la calidad del sistema. A continuación se especifican las actividades de la fase de construcción y pruebas

- Preparación del entorno de construcción (CPS 1)
- Creación de código (CPS 2)
- Elaboración de procedimientos técnicos (CPS 3)
- Elaboración de procedimientos normativos (CPS 4)
- Ejecución de las pruebas unitarias (CPS 5)
- Ejecución de las pruebas de integración (CPS 6)
- Revisión de estándares de construcción (CPS 7)
- Elaboración de manuales de usuario (CPS 9)
- Pruebas del sistema (CPS 10)
- Definición de capacitación de usuarios finales (CPS 11)
- Pruebas de aceptación del sistema (CPS 12)

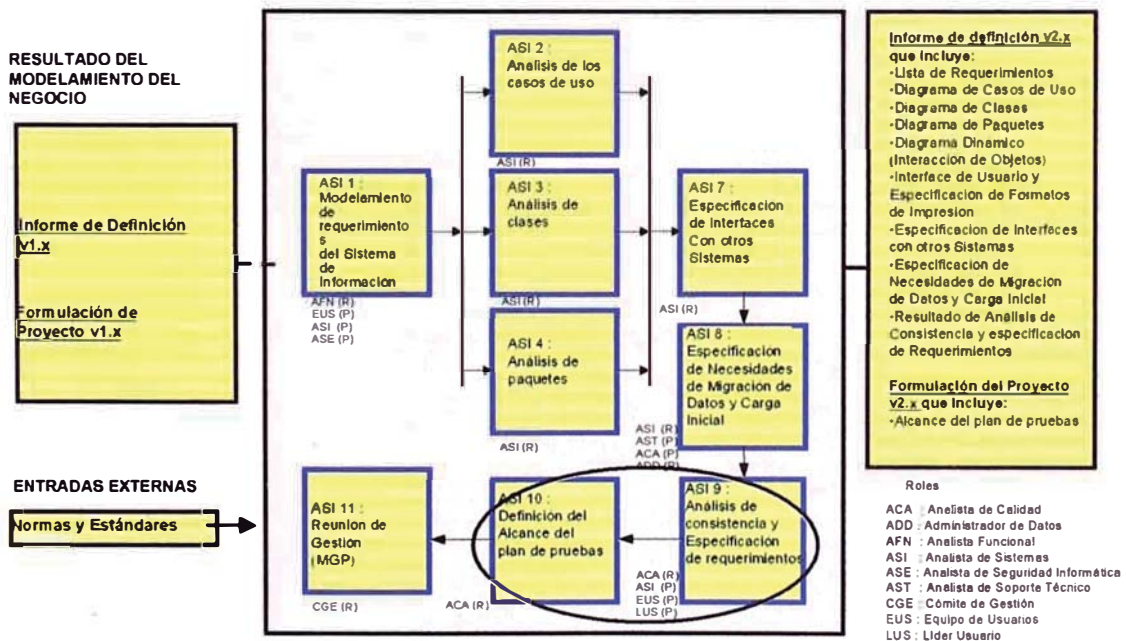
MPN Flujo de trabajo

Modelamiento del Negocio



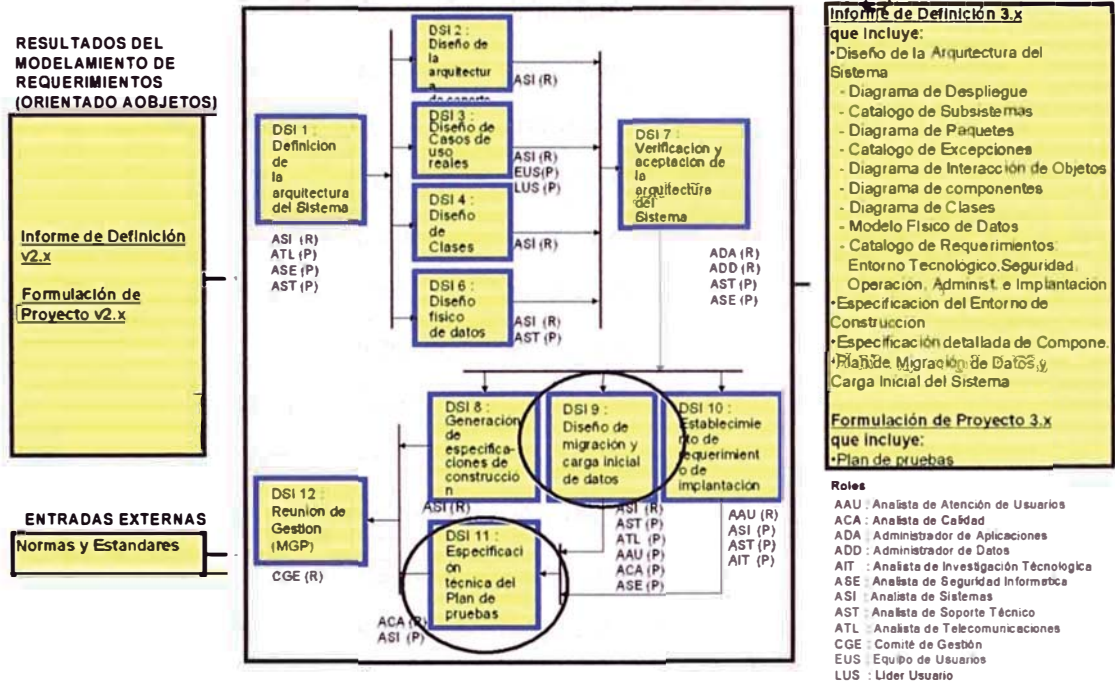
ASI - (O.Objetos)

Modelamiento de Requerimientos

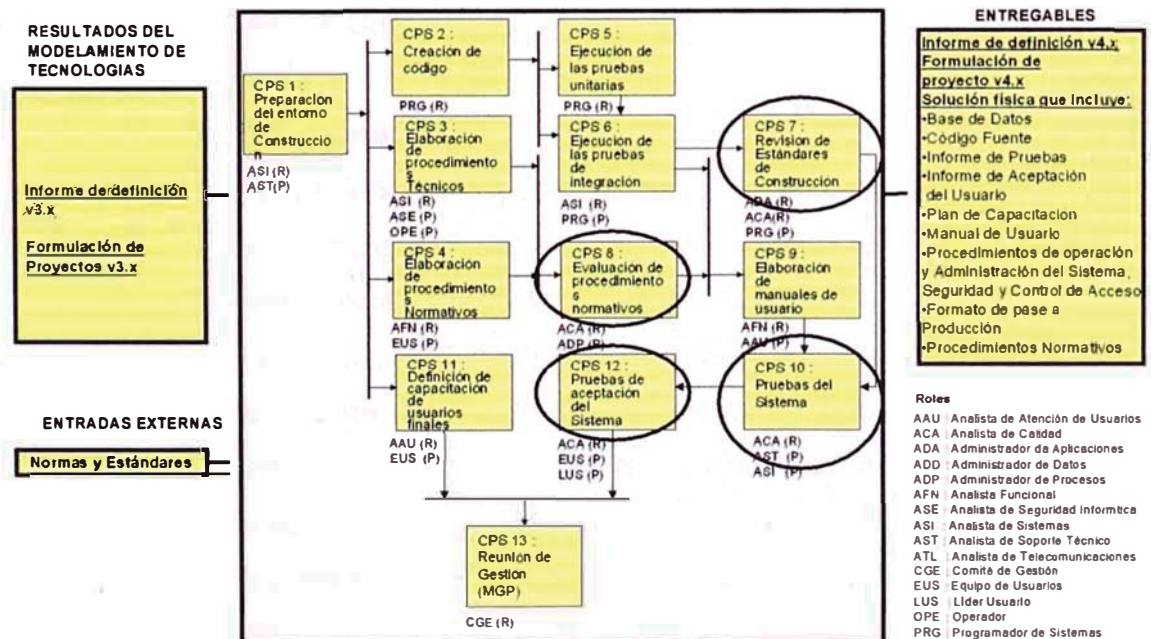


DSI - (O.Objetos)

Modelamiento de Tecnología



CPS - Construcción y Pruebas



2.5 Herramientas de Productividad para el apoyo a las Pruebas

Son herramientas BTO que sirven de apoyo para la solución integral de Aseguramiento de Calidad en el ciclo de vida de los sistemas, la cual permite y facilita la administración de requerimientos, pruebas funcionales y de regresión automatizadas, administración de pruebas, administración de datos de prueba y seguimiento de ocurrencias y de proyectos; permitiendo además realizar pruebas de carga y stress, identificar cuellos de botella en las aplicaciones y analizar el tiempo de respuesta de las aplicaciones a nivel transaccional, permitiendo además mejorar el desempeño y la fiabilidad de las aplicaciones y lo que redundará además en minimizar la cantidad de errores y el tiempo de las pruebas realizadas. Los productos Mercury, Compuware y Rational son grandes referentes de estas herramientas. Para el presente estudio se ha tomado en cuenta en la evaluación de alternativas la solución Compuware por tal motivo a continuación se describen algunas de las herramientas que propone según la siguiente clasificación

- Pruebas Funcionales
- Pruebas de Performance
- Aseguramiento de la Calidad en el ciclo de vida de las aplicaciones
- Pruebas de Caja Blanca

2.5.1 Pruebas Funcionales

QARun .- QARun brinda la capacidad de automatización necesaria para crear y ejecutar de manera rápida y productiva scripts de pruebas, verificar pruebas y analizar resultados de las pruebas. QARun permite mantener la sincronización con los ciclos acelerados de versiones de las aplicaciones durante todo su ciclo de vida, mejorar el ambiente del proceso de pruebas y distribuir aplicaciones con la calidad que los usuarios finales y la administración esperan obtener. Es una herramienta automática de pruebas funcionales que permite a las organizaciones validar las aplicaciones críticas de negocios, ya sean cliente / servidor, ecommerce (web / java) o CRM/ERP. Las capacidades automáticas de QARun ayudan a mejorar la productividad

de los “testers”, capacitándolos para crear eficientes pruebas repetibles, verificar la respuesta de la aplicación y analizar los resultados de la prueba. . Trabaja en base a una definición del script que debe ejecutar, el script se prepara aprox. en un 90% gráficamente siguiendo los pasos a ejecutar y un 10% de actualización manual sobre todo en los controles que se puedan poner (chequeos).

2.5.2 Pruebas de Performance

Se componen de las siguientes herramientas :

QALoad .- Herramienta para prueba de carga y concurrencia, captura todos los requerimientos que pasan por un determinado puerto de comunicación. QALoad también posee la habilidad para simular una aplicación Cliente/Servidor sin comprometer al usuario final o al equipo de usuarios finales. Lleva a cabo la prueba de carga capturando el middleware y creando usuarios virtuales que ejecuta y simula usuarios físicos. QALoad genera gráficos y reportes que le ayudan a entender el desempeño de la base de datos, aplicación o servidor Web

Permite hacer mediciones de la escalabilidad y desempeño de las aplicaciones bajo tráfico ilimitado, brindando la facilidad de predecir el tiempo de respuesta de las aplicaciones en ambientes diferentes. QALoad permite:

- Emular desde un punto central, cualquier número de usuarios y su impacto en los sistemas
- Realizar pruebas de gran escala con recursos de hardware mínimos
- Ajustar niveles de volumen de usuarios virtuales
- Identificar y corregir problemas de escalabilidad tempranamente
- Correlacionar tiempos de respuesta pobres con niveles de usuarios virtuales.
- Medir y analizar los tiempos de respuesta de las aplicaciones y utilización de sistemas sometidos a cargas similares a producción
- Utilizar data y reportes para desarrollar acuerdos de niveles de servicio realistas

- Obtener data objetiva en ambientes diferentes
- Aplicar actividad de red predecible y carga multi-protocolo durante las pruebas de carga
- Variabilizar datos de entrada
- Correlacionar tiempos de respuesta pobres con el desempeño de los servidores

ApplicationVantage

Identificación de problemas de desempeño de las aplicaciones, brindando información detallada del comportamiento de las aplicaciones a nivel interno. ApplicationVantage permite identificar y corregir las causas del tiempo de respuesta pobre de las aplicaciones donde se presenten, evitando la pérdida de tiempo en tareas de búsqueda de estas causas. Application Vantage permite además

- Resolver problemas de desempeño de las aplicaciones en ambientes de pre-producción y producción
- Identificar problemas de desempeño de las aplicaciones, a nivel cliente, red, servidor o la misma aplicación
- Validar acciones correctivas utilizando sus características de análisis predictivo
- Certificar el éxito de desarrollos nuevos y modificados de las aplicaciones
- Establecer y cumplir con los acuerdos de niveles de servicio

2.5.3 Aseguramiento en la Calidad en el ciclo de vida de las aplicaciones

El Aseguramiento de Calidad en el ciclo de vida de las aplicaciones alinea los requerimientos con las pruebas de calidad, permitiendo:

- Priorizar las actividades dentro del proceso de pruebas
- Alinear los requerimientos de pruebas con los logros del negocio
- Medir rápidamente el progreso y la efectividad de las actividades de pruebas

- Administrar y ejecutar centralizadamente tareas manuales y automatizadas
- Automatizar el proceso de registro, seguimiento y resolución de problemas encontrados durante la ejecución de las pruebas

Está compuesto por las siguientes Herramientas

Reconcile.- Define, rastrea y gestiona sus necesidades a través del ciclo de vida completo del proyecto. Reconcile es un sistema de gestión de requerimientos para toda la empresa. Permite a los equipos de proyectos crear, modificar, rastrear, evaluar e informar de los requerimientos del proyecto. Reconcile combina el poder de Microsoft Word, la flexibilidad de una base de datos relacional y la simplicidad de su interfaz Project Explorer, para ofrecer una solución amplia a las necesidades de gestión de requisitos. Yendo aún más lejos de la gestión de las necesidades básicas, la integración con TrackRecord de Compuware y QA Director le permite supervisar el progreso del proyecto, desde la fase de planificación a través de las fases de pruebas y depuración del desarrollo del producto. Reconcile ayuda a todo el mundo que esté relacionado con un proyecto a entender los requisitos funcionales, de pruebas y de negocio del proyecto, y la relación entre ellos, eliminando desviaciones en el programa de costes y fallos en la aplicación. Con su habilidad para relacionar los requerimientos así como integrarse con TrackRecord, Reconcile hace más fácil y rápido a cualquier miembro del equipo.

QADirector.- Herramienta que permite automatizar la ejecución de las pruebas funcionales. Permite establecer la relación existente entre los requerimientos previamente definidos en el Reconcile con los casos de pruebas automatizados también denominados Script de Testing. Muy útil en el caso de pruebas de regresión. Le dá un marco de trabajo para gestionar el proceso de las pruebas por completo desde diseñar, ejecutar, hasta el análisis del resultado. Sus capacidades de pruebas distribuidas y la

arquitectura abierta permiten a los equipos de pruebas y desarrollo controlar las actividades de pruebas a través de entornos heterogéneos desde un único punto de control. QADirector promueve pruebas amplias y consistentes, permitiendo a los desarrolladores, “testers” y administradores de QA compartir las valoraciones de pruebas y la información actual e histórica sobre las pruebas y los resultados de éstas.

Track Record .- Herramienta para el registro y gestión de los errores de pruebas, puede instalarse incluso una interfase. Trabaja en base a un Workflow que para nuestro caso tendría que definirse de acuerdo a nuestra modalidad de trabajo. TrackRecord es una herramienta para rastrear fallos y determinar el estado del proyecto, que sirve como depósito central y eje de comunicaciones para todas las actividades y los datos relacionados con el desarrollo.

File AID Client Server.- Se utiliza para guardar backup de las entidades a ser actualizadas durante las pruebas, para luego poder restaurarlas y así poder realizar las pruebas de regresión en caso correspondan.

2.5.4 Pruebas de Caja Blanca

Se compone de las siguientes herramientas :

DevPartner Java Edition .- Mejoramiento del desempeño y confiabilidad del código java. DevPartner Java Edition es un grupo de herramientas que acelera el desarrollo de aplicaciones y componentes java de alto desempeño. DevPartner Java Edition brinda la capacidad de realizar un análisis detallado de las aplicaciones java, ya sea que estas se ejecuten sobre un solo computador o sobre multiples computadores en un ambiente distribuido. DevPartner Java Edition permite realizar tareas productivas en el desarrollo de aplicaciones Java, incluyendo:

- Análisis de aplicaciones distribuidas
- Análisis de memoria

- Análisis de desempeño
- Análisis de cobertura de código
- Detección bloqueos de hilos
- Revisión de código
- Análisis de diseño

DevPartner Studio Professional Edition .- DevPartner Studio Professional Edition es una herramienta que ayuda a acelerar las tareas productivas en la construcción de aplicaciones multi-lenguaje. DevPartner Studio Professional permite la detección rápida, diagnóstico y resolución de errores de software, maximización del desempeño del código y el aseguramiento de una cobertura de código óptima. DevPartner Studio Professional Edition incluye las siguientes características:

- Revisión de Código Fuente
- Detección y Diagnóstico de errores
- Análisis de memoria
- Análisis de cobertura de código
- Análisis de desempeño
- Simulador de Fallas

CAPITULO III

PROCESO DE TOMA DE DECISIONES

3.1 Planteamiento del Problema

Hacia finales del año 2005 la problemática de la calidad de software en la SUNAT era la siguiente

- Aproximadamente el 60% de los recursos de las áreas encargadas del desarrollo de Sistemas, se dedican a realizar tareas de mantenimiento.
- Existe un bajo nivel de cumplimiento en el uso de la metodología de desarrollo de sistemas MDSI.
- Las pruebas de software se realizan una vez que el producto ha culminado su desarrollo, modificar un error al final del desarrollo es mas costoso que si se hubiese detectado de manera temprana.
- Muchas veces los requerimientos funcionales no alcanzan a satisfacer las necesidades de los usuarios.
- Existe una deficiente gestión de la configuración en el ambiente de pre-producción utilizadas para realizar las pruebas.
- No existen métricas para medir la calidad de software.
- Más del 50% de los productos de Software contienen errores en el momento de remitirse al área de pruebas.
- Aproximadamente el 30% de los pases a producción tienen prioridad de urgentes, lo que complica a los tiempos requeridos para unas pruebas eficientes.

- El costo de las pruebas en término de tiempo es demasiada alta.
- No se realizan planes de pruebas, ni se cuentan con herramientas adecuadas para su administración.
- No existe un adecuado control y registro de los casos de pruebas realizados, ni se cuentan con herramientas adecuadas para su administración.
- Complejidad para verificar la trazabilidad del software probado, es decir, no se disponen de herramientas que permitan realizar el análisis de cobertura de código ejecutado en las pruebas, lo que permitiría medir la completitud o lo contrario de los casos de prueba empleados.
- Los frecuentes errores detectados en las pruebas conllevan a la modificación del software y por ende a realizar pruebas de regresión que se traducen en tiempo y esfuerzo adicional, que muchas veces no se realizan por los tiempos limitados con que se cuentan.
- No existen formatos de check list, ni procedimientos homogéneos de trabajo, esto ocurre en la División de Control de Calidad como en las Areas de Desarrollo de Software. En los anexos 4 y 5 se muestran algunos de los formatos que son utilizados en la DCC pero sin que hayan sido formalmente aprobados.
- No existe una cultura organizacional sobre la Calidad en la INSI.

Según la Metodología MDSI los pases a producción en la SUNAT se clasifican de la siguiente manera

Proyecto : Aquellos que contienen productos de software relacionados a los proyectos operativos institucionales planteados por la alta dirección.

Funcionalidad : Aquellos que contienen desarrollos de nuevos productos de software para la solución de un problema y cuyo tiempo de desarrollo es mayor a los 20 días hábiles.

Mantenimiento : Aquellos que contienen modificaciones y /o desarrollos de nuevas funcionalidades y cuyo desarrollo no es mayor a los 20 días hábiles.

Corrección : Aquellos software que resuelven los errores de información en la base de datos.

En el año 2005 la División de Control de Calidad atendió aproximadamente 2,356 pases a producción de los cuales en 1,248 pases que representa el 52.57% del total se han detectado al menos un error en el Software.

A continuación se muestra la estadística del total de errores detectados en el año 2005, se debe tener en cuenta que un pase a producción podría contener uno o muchos errores.

Numero de Errores detectados			
Funcionalidad	Leve	Grave	Muy Grave
5755	2178	2566	1111

El siguiente cuadro muestra uno de los reportes mensuales en el año 2005 de la eficiencia en la eliminación de errores. El indicador promedio del año fue aproximadamente del 35%, se calcula a partir de la cantidad de errores detectados en un periodo, dividido entre la cantidad total de tickets de errores recibidos en el mismo periodo (ED3+E01+E03). Se denomina tickets a aquellos reportes electrónicos remitidos a la División de Atención a Usuarios los cuales se originan por algún problema del software. La desventaja de este indicador es que la cantidad de tickets recibidos comprende el total histórico de errores de Software, lo cual muchas veces ocasiona que los errores de un periodo sean arrastrados en el resultado del indicador de los próximos periodos.

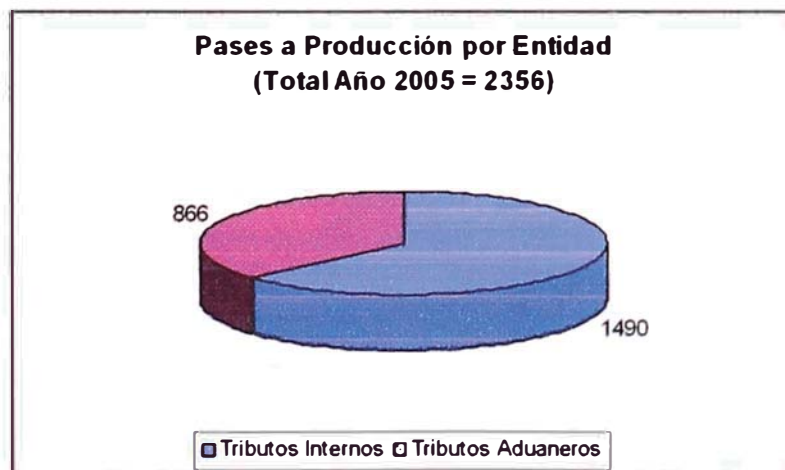
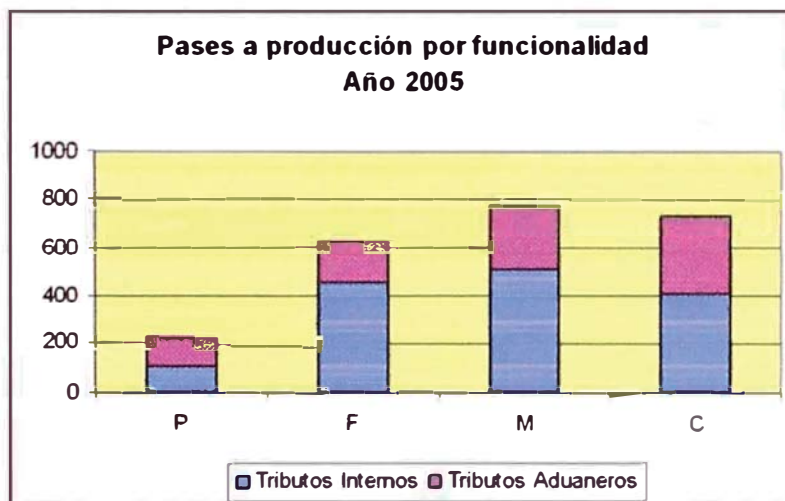
Tickets de Errores Recibidos	
ED3 - Error de Datos x Mal Funcionamiento Sistema	688
E01 - Error de Software	68
E03 - Error de Documentación del Usuario	58
Total de Tickets de Errores Recibidos	814

Errores Detectados en Control de Calidad	
En Definición	14
En Instalación del Pase a Producción	42
En Documentación del Usuario / Instructivo	74
En Software	312
Total de Errores Detectados en Control de Calidad	442
EFICIENCIA DE ELIMINACIÓN DE ERRORES	35%

A continuación se muestra las estadísticas según su clasificación de los pases a producción durante el año 2005.

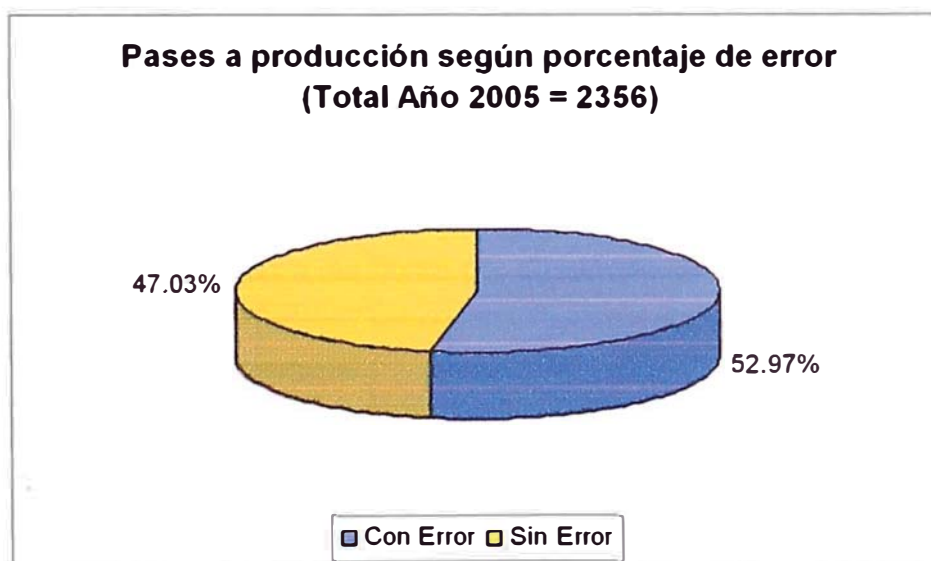
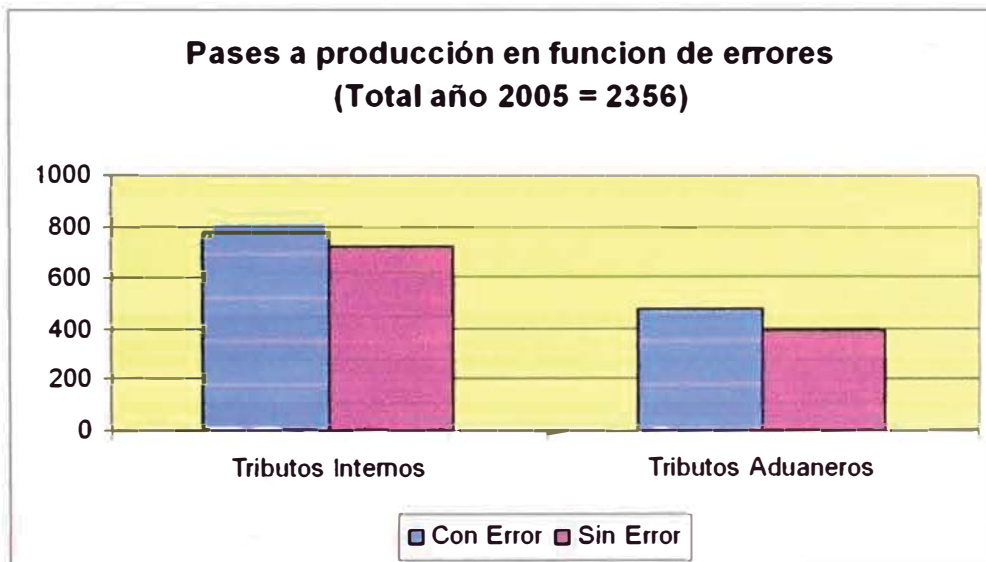
Por Funcionalidad					
Funcionalidad		Tributos Internos	Tributos Aduaneros	Total	%
Proyecto	P	114	115	229	9.72%
Funcionalidad	F	458	166	624	26.49%
Mantenimiento	M	509	265	774	32.85%
Corrección	C	409	320	729	30.94%
Total		1490	866	2356	100.00%

Nótese que mas del 60% de los pases a producción son por mantenimiento de software y corrección de errores, lo que refleja un problema en el proceso de aseguramiento de la calidad.



El siguiente cuadro indica que en año 2005 se detectaron en el área de pruebas aproximadamente el 52.97% de pases a producción con errores.

Por Error					
Pases		Tributos Internos	Tributos Aduaneros	Total	%
Con Error	OKE	771	477	1248	52.97%
Sin Error	OK	719	389	1108	47.03%
Total		1490	866	2356	100.00%

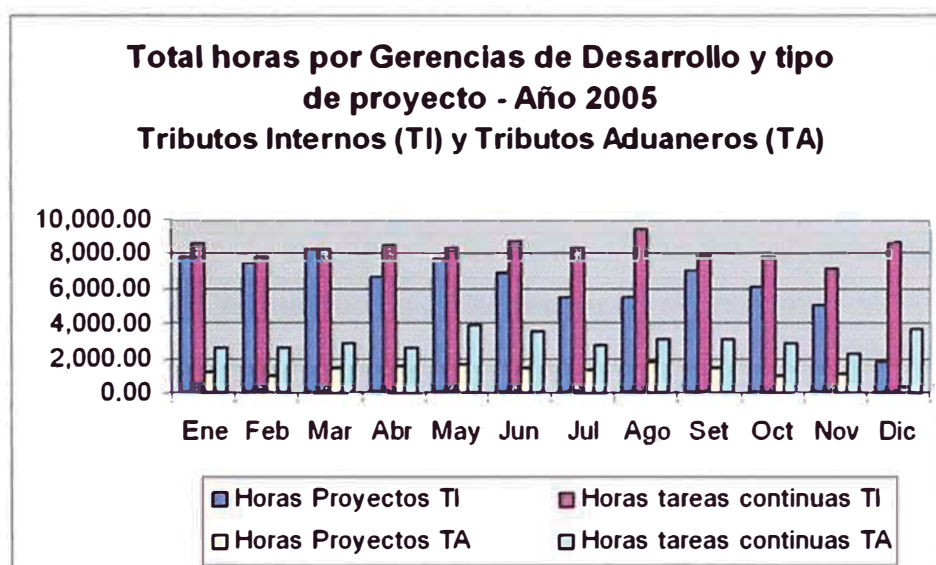
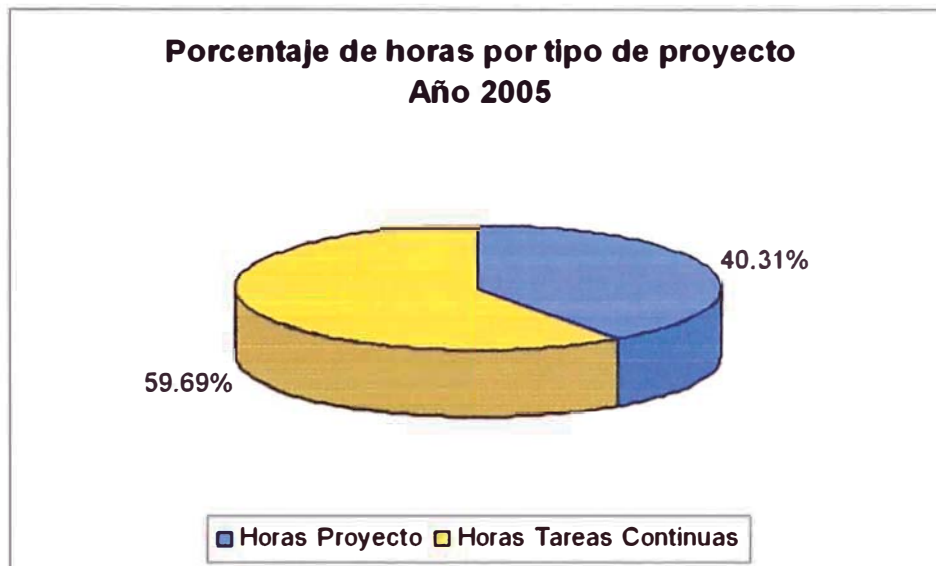


Aproximadamente el 60% de los recursos de las Gerencias de Desarrollo de Sistemas están dedicados a las tareas de mantenimiento, también llamado tareas continuas en el control de proyectos de la INSI. Tal como se muestra en el cuadro siguiente, el registro de horas está clasificado según corresponda a la "Gerencia de Desarrollo de Sistemas Tributarios y de Gestión GDSTG" y la "Gerencia de Desarrollo de Sistemas Aduaneros GDSA", los cuales se dedican al desarrollo de los Sistemas de Desarrollo de Tributos Internos (TI) y Sistemas de Desarrollo de Tributos Aduanero (TA) respectivamente.

Meses	Cuadro de registro de horas en Gerencias de Desarrollo - Año 2005					
	GDSTG			GDSA		
	Horas proyectos	Horas tareas continuas	Total Horas	Horas proyectos	Horas tareas continuas	Total Horas
Ene	7,792.00	8,585.50	16,377.50	1,212.00	2,683.00	3,895.00
Feb	7,423.50	7,820.50	15,244.00	1,018.00	2,629.00	3,647.00
Mar	8,248.50	8,320.50	16,569.00	1,484.00	2,862.00	4,346.00
Abr	6,651.00	8,450.00	15,101.00	1,615.00	2,618.00	4,233.00
May	7,650.50	8,372.50	16,023.00	1,673.00	3,866.00	5,539.00
Jun	6,907.00	8,769.00	15,676.00	1,508.00	3,617.00	5,125.00
Jul	5,554.00	8,353.50	13,907.50	1,358.00	2,741.00	4,099.00
Ago	5,571.50	9,426.00	14,997.50	1,831.00	3,102.00	4,933.00
Set	7,000.30	7,880.50	14,880.80	1,530.00	3,080.00	4,610.00
Oct	6,142.60	7,790.50	13,933.10	1,043.00	2,830.00	3,873.00
Nov	5,047.00	7,183.50	12,230.50	1,106.00	2,291.00	3,397.00
Dic	1,814.50	8,592.00	10,406.50	390.00	3,715.00	4,105.00
Total	75,802.40	99,544.00	175,346.40	15,768.00	36,034.00	51,802.00
100%	33.37%	43.82%	77.19%	6.94%	15.86%	22.81%

El porcentaje de horas dedicadas al desarrollo de proyectos representa aproximadamente el 40% del total, se puede observar que tanto en las Gerencias GDSTG como en la GDSA las horas dedicadas a tareas de mantenimiento superan a las horas por desarrollo de nuevos proyectos.

Tributos	Horas Proyecto	Horas Tareas Continuas
TI	33.37%	43.82%
TA	6.94%	15.86%
TOTAL	40.31%	59.69%



Nótese en la figura anterior que en general existe mayor cantidad de horas registradas en la GDSTG (TI) con respecto a la GDSA (TA), esto es directamente proporcional a la cantidad de productos de software existentes en éstas gerencias. Se observa que las tareas de mantenimiento superan a las tareas de desarrollo de proyectos durante todo el año en cada una de las dos gerencias.

3.2 Alternativas de Solución

Frente a la problemática expuesta anteriormente se cuenta con las siguientes alternativas de solución :

Alternativa 1 : Proceso del aseguramiento de la calidad de software basado en la metodología MDSI sin herramientas complementarias..

Alternativa 2 : Proceso automatizado del aseguramiento de la calidad de software basado en el MDSI complementándose con herramientas de productividad de las pruebas de software. El producto elegido es compuware.

3.3 Metodología de Solución

Para la elección de la alternativa de solución, se utilizó la evaluación por criterios de selección y el criterio de evaluación financiera. La solución debe proponer un proceso de aseguramiento de la calidad del software basado en el cumplimiento de la metodología de desarrollo de sistemas MDSI y que además permita mejorar la productividad de las Areas de Desarrollo.

3.3.1 Evaluación por Criterios de Selección

Se consideraron los siguientes criterios :

Impacto de la calidad : Mejoramiento de la calidad interna y externa del software y la disminución de costos por tareas dedicadas al mantenimiento de software.

Bajo Costo : Bajo costo de inversión para la implantación de la solución.

Plazo Corto : Plazo corto para la implantación de la solución.

Aprendizaje : Facilidad de aprendizaje.

Mejoramiento de la productividad de pruebas : Mejorar la productividad en la identificación de errores en menor tiempo.

Eficiencia Técnicas : Eficiencia en las pruebas de volumen, trazabilidad, pruebas de stress, pruebas de comunicaciones.

Se plantearon las siguientes escalas :

1 : Muy Bajo

2 : Poco

3 : Regular

4 : Mucho

Matríz de Selección de Alternativas	Impacto de Calidad	Bajo Costo	Plazo Corto	Aprendizaje	Mejoramiento en identificación de errores (Productividad de pruebas)	Eficiencia Tecnicas	Puntaje Total
	25%	10%	10%	5%	25%	25%	100%
Alternativa 1	2	4	3	3	2	1	2.1
Alternativa 2	4	1	2	2	4	4	3.4

Del análisis anterior se puede concluir que la **alternativa 2** es la de mayor calificación, debido a que el impacto en la calidad de Software, el mejoramiento de la eficiencia en identificación de errores y el mejoramiento en eficiencia técnicas es mejor que la alternativa 1.

3.3.2 Evaluación Financiera

Para la evaluación financiera de las alternativas 1 y 2 se considerará la siguiente premisa : “El costo por dedicación a la tarea de mantenimiento de software disminuirá en la medida que la calidad del software mejore”.

El aseguramiento de la calidad de software permitirá mejorar la eficiencia de detección de errores y mejorar la satisfacción de los usuarios, además que los errores podrán identificarse en una etapa mas temprana del desarrollo del software desde la definición de requisitos funcionales y no funcionales lo que evitará corregir el error una vez finalizado el producto final lo que siempre significará un esfuerzo mucho mayor traducido en costos. La

disminución del costo de mantenimiento por mejora en la calidad de software se le denominará ahorro en costo.

Basándonos en la presentación de Software S.A., representante de la empresa Compuware Corporation, que indica que la aplicación conjunta de una metodología y una herramienta de prueba aumentan en 60% la productividad en el largo plazo, se considerará el supuesto de que “utilizando solo la metodología o la herramienta de pruebas pero no ambas, se obtendrá la mitad de la productividad que si se aplicara ambas a la vez”.

Las estadísticas de la SUNAT indican que aproximadamente el 60% de los recursos de las Areas de Desarrollo de Sistemas se dedican a labores de mantenimiento de software. Se deduce que del total de las 127 personas que conforman las Gerencias de Sistema de Tributos Internos y Sistemas Aduaneros, el 60% es aproximadamente 76 personas.

Se considerará costo de mantenimiento de software al costo total por pago de remuneraciones al personal dedicado a estas tareas.

A continuación se muestra el cuadro de costos que permite realizar el cálculo financiero de las alternativas de solución

Calculo Financiero

Descripción	Valor	Campo	Fórmula
Tipo de cambio	3.3	A1	
Jornal diario en soles	166	A2	
Numero de recursos verifica requisitos	4	A3	
Tiempo proceso de verifica requisitos	1 Mes	A4	
Sueldo mensual promedio en soles	5,000.00	A5	
Inversión por proceso de verificación requisitos en dólares	6,060.61	A6	$(A5 \times A3)/A1$
Numero de horas de capacitación	120	A7	
Numero de recursos capacitados	30	A8	
Cantidad de semanas capacitacion	3	A9	
Inversion por capacitación de software en soles	112,500.00	A10	$(A5 \times A9/4) \times A8$
Inversión por capacitación de software en dólares	34,090.91	A11	$A10 / A1$
Número de recursos DCC capacitación MDSI	25	A12	
Número de recursos resto capacitación MDSI	275	A13	
Cantidad de semanas capacitación MDSI	2	A14	
Inversion en capacitación DCC MDSI en soles	62,500.00	A15	$(A5 \times A14/4) \times A12$
Inversion en capacitación DCC MDSI en dólares	18,939.39	A16	$A15 / A1$
Inversión en capacitación resto MDSI en soles	687,500.00	A17	$(A5 \times A14/4) \times A13$
Inversión en capacitación resto MDSI en dólares	208,333.33	A18	$A17 / A1$
Número de sueldos por año	14	A19	
Ingreso anual en soles por persona	70,000.00	A20	$A5 \times A19$
Ingreso anual en dólares por persona	21,212.12	A21	$A20 / A1$
Costo anual en dólares en DCC	530,303.03	A22	$A21 \times A12$
Cantidad de personal en GDSTG	91	A23	
Cantidad de personal en GDSA	36	A24	
Porcentaje de tareas de mantenimiento	60.00%	A25	
Total personas dedicadas a mentenimiento	76	A26	$(A23 + A24) \times A25$
Costo mantenimlento anual en dólares	1,616,363.64	A27	$A21 \times A26$

3.3.2.1 Alternativa 1

Se han considerado los siguientes costos en la alternativa 1

Horas de capacitación en la metodología MDSI = 80 Hrs o 2 semanas calendario.

Costo de RRHH personal = 166 Nuevos Soles por jornal diario

Promedio de sueldo mensual = 5000 Nuevos Soles

Tipo de Cambio = 3.3

Cantidad de personal en la DCC = 25

Inversión en capacitación MDSI en DCC en Dólares = $(25 \cdot 5000 \cdot 2/4)/3.3 = 18,939.39$ US\$

Inversión en capacitación MDSI Resto de Personal = $(275 \cdot 5000 \cdot 2/4)/3.3 = 208,333.33$ US\$

Costo anual en Dólares en DCC = $25 \cdot 70,000/3.3 = 530,303$ US\$ aproximadamente.

Cantidad de personas dedicadas al mantenimiento de sistemas = $60\% \cdot (91+36) = 76$

Costo Anual de Mantenimiento en Dólares = $70,000 \cdot 76/3.3 = 1,616,363.64$ US\$

Supuesto alternativa 1 : El ahorro en costo de mantenimiento es del 15%,30%,30%,30%,30% en el primer, segundo, tercer, cuarto y quinto año respectivamente; es decir, 242,454.55 US\$, 484,909.09 US\$, 484,909.09 US\$, 484,909.09 US\$, 484,909.09 US\$ respectivamente.

Costo Anual de Mantenimiento en Dólares = $70,000 \cdot 76/3.3 = 1,616,363.64$ US\$

Cuadro de evaluación financiera de la alternativa 1

Alternativa 1	Año 0	Año 1	Año 2	Año 3	Año 4	Año 5
	\$	\$	\$	\$	\$	\$
Total Gastos del Proyecto						
Inversión en Capacitación DCC MDSI	-18,939.39					
Inversión en Capacitación Resto MDSI	-208,333.33					
Costo Anual por Remuneraciones DCC	0	-530,303.03	-530,303.03	-530,303.03	-530,303.03	-530,303.03
Ingresos Tangibles						
Ahorro Anual en costos de mantenimiento (15%, 30%, 30%, 30%, 30%)	0.00	242,454.55	484,909.09	484,909.09	484,909.09	484,909.09
Flujo de Caja	-227,272.73	-287,848.48	-45,393.94	-45,393.94	-45,393.94	-45,393.94
Factor de Descuento por interés 15%	1.0000	0.8696	0.7561	0.6575	0.5718	0.4972
Flujo de caja descontado	-227,272.73	-250,313.04	-34,322.36	-29,846.52	-25,956.25	-22,569.87
Valor Presente Neto	-590,280.76					

3.3.2.2 Alternativa 2

La herramienta de soporte a las pruebas de Compuware se complementará a la metodología MDSI para el aseguramiento de la calidad de Software. Se han considerado los siguientes costos para la alternativa 2

Costo de Adquisición de Licencias de Software = 128,450 US\$

Servicio de aplicación del software en un sistema SUNAT = 19,040 US\$

Servicio de Capacitación de Software = 6,608.70 US\$

Costo por mantenimiento de software por 2 años = 20,552 US\$

Número de recursos en proceso de verificación de requisitos = 4 personas por 1 mes a dedicación exclusiva.

Inversión de proceso de verificación de requisitos del Software = $5,000 * 4 / 3.3$
= 6.060.61 US\$

Los costos por capacitación en la metodología MDSI son los mismos de la Alternativa 1.

Costo Anual de Mantenimiento en Dólares = $70,000 * 76 / 3.3$ = 1,616,363.64 US\$

Supuesto alternativa 2 : El ahorro en costo de mantenimiento es del 30%,60%,60% en el primer, segundo y tercer año respectivamente; es decir, 484,909.09 US\$, 969,818.18 US\$ y 969,818.18 US\$ respectivamente.

Cuadro de evaluación financiera alternativa 2

Alternativa 2	Año 0	Año 1	Año 2	Año 3
	\$	\$	\$	\$
Total Gastos del Proyecto				
Licencias perpetuas de software	-128.450.00			
Servicio de aplicación de prueba de un sistema de SUNAT	-19.040.00			
Servicio de Capacitación	-6.608.70			
Costo por Mantenimiento de Software 2 años	-20.552.00			
Inversión por Capacitacion de Software	-34.090.91			
Inversión por proceso de verificación requisitos	-6.060.61			
Inversión en Capacitación DCC MDSI	-18.939.39			
Inversión en Capacitación Resto MDSI	-208.333.33			
Costo Anual por Remuneraciones DCC	0.00	-530.303.03	-530.303.03	-530.303.03
Ingresos Tangibles				
Ahorro Anual en costos de mantenimiento (30%,60%,60%)	0.00	484.909.09	969.818.18	969.818.18
Flujo de Caja	-442,074.94	-45,393.94	439,515.15	439,515.15
Factor de Descuento por interés 15%	1.00	0.8696	0.7561	0.6575
Flujo de caja descontado	-442,074.94	-39,474.57	332,317.41	288,981.21
Valor Presente Neto	139,749.11			
Tasa de Retorno	0.32	PR (meses) =		30

La Tasa de Retorno = 0.32 y el Periodo de retorno de 32 meses.

3.4 Toma de Decisiones

Se observa que tanto la evaluación por criterios de selección como la evaluación financiera, concluyen que la alternativa 2 es la mas recomendable con un periodo de retorno aproximado de 16 meses.

La alternativa 2 además de requerir la implementación de procedimientos y acciones orientadas a garantizar la calidad, las cuales deben estar alineadas a la metodología MDSI, requiere también de una herramienta para el apoyo al aseguramiento de la calidad, contribuyendo de esta manera a que los costos por el tiempo invertido en las inspecciones y verificaciones exhaustivas, sean contrarrestadas con la mejora en la productividad de los procesos de desarrollo del software.

3.5 Estrategias Adoptadas

La división de Control de Calidad DCC se encarga del Control de Calidad de los sistemas, el cual abarca además, la verificación de los procedimientos contenidos tanto en las normas legales como en circulares, el control del cumplimiento de la metodología de sistemas de información y el desarrollo de manuales de usuario.

La estrategia propuesta en la SUNAT para el aseguramiento de la calidad de software se basa en realizar las siguientes acciones :

- Aplicación de una Metodología de desarrollo durante el ciclo de vida de las aplicaciones.
- Implementación de métricas de calidad de software.
- Utilización de herramientas BTO para el soporte al proceso de aseguramiento de calidad.
- Para minimizar el riesgo de adquirir una herramienta que no satisfaga los requerimientos esperados, se convocaría a los 3 grandes líderes mundiales.
- Complementando al apéndice anterior, se trataría de minimizar el costo, adquiriendo licencias flotantes para los productos que proveería la herramienta.

3.5.1 Aplicación de una Metodología de Desarrollo

La Metodología de Desarrollo de Sistemas de Información MDSI, permitirá cumplir un conjunto de procedimientos, reglas y estándares de desarrollo para conseguir un producto de calidad. Es esencial el cumplimiento de la metodología por parte de las Áreas de Desarrollo, así como el rol que cumplirá la División de Control de Calidad DCC en cada fase de la metodología al tener la función de validar, verificar, identificar errores o deficiencias y plantear mejoras a la solución propuesta.

A continuación se presenta un cuadro de tareas de la metodología donde interviene la Gerencia de Servicios GSU y directamente la División de Control de Calidad DCC, en la que se tiene un producto de salida para cada actividad común dentro de las etapas de la metodología.

Nro. MDSI	TAREAS DE LA DCC
FASE	MODELAMIENTO DE REQUERIMIENTOS
Tarea ASI 9	Análisis de Consistencia y especificación de Requerimientos
Tarea ASI 10	Definición del alcance de las pruebas
FASE	MODELAMIENTO DE TECNOLOGIA
Tarea DSI 11.1:	Especificaciones del entorno de pruebas
Tarea DSI 11.2:	Especificaciones técnicas de las pruebas
FASE	MODELAMIENTO DE CONSTRUCCION Y PRUEBAS
Tarea CPS 7	Revisión de estándares de construcción
Tarea CPS 8	Evaluación de procedimientos normativos
Tarea CPS 10	Pruebas del Sistema
Tarea CPS 11	Pruebas de aceptación del sistema

Además de las actividades arriba mencionadas, se debe rescatar la importancia que representa la ejecución de las Pruebas Unitarias y de Integración por parte de las Areas de Desarrollo de Sistemas, a través de las tareas CPS 5 y CPS 6 del MDSI, además de la participación de la División de Control de Calidad como apoyo en la tarea DSI 9, demostrando que el proceso de aseguramiento de la calidad del software no es exclusiva responsabilidad de la DCC.

Procedimiento en la fase de Modelamiento de Requerimientos

3.5.1.1 ASI 9: Análisis de Consistencia y Especificación de Requerimientos

GENERALIDADES	
Objetivo :	Después de :
El objetivo de esta actividad es validar los modelos respecto al catálogo de requisitos y de interface a partir de las especificaciones de interface de usuario.	Especificación de necesidades de migración de datos y carga inicial (ASI 9.2)
	Antes de :
	Definición del Alcance del Plan de Pruebas (ASI 10).
DESARROLLO	
Entrada	Salida
<ul style="list-style-type: none"> • Informe de Definición F2 v.1.x • Formulación de Proyecto v1.x <p>Análisis Orientado a Objetos:</p> <ul style="list-style-type: none"> • Modelo de Clases de Análisis • Diagrama de casos de uso • Modelo de Paquetes de Subsistemas • Diagrama de navegación • Diagrama de colaboración 	<ul style="list-style-type: none"> • Informe de verificación (anexo 9)
PROCEDIMIENTO	
<p>a) Registro de requerimientos</p> <p>Todos los requerimientos que se encuentran en el informe de definición F2, deben ser registrados y almacenados en el repositorio de requerimientos provista por la herramienta BTO para apoyo al aseguramiento de la calidad. Para ello la herramienta debe proveer una plantilla que permita exportar a su entorno los requerimientos definidos en el informe F2 y hechos en Microsoft Word. Los requerimientos estarán clasificados de la siguiente manera :</p> <ul style="list-style-type: none"> • MPN Modelamiento de procesos del negocio 	

- RF Requerimientos Funcionales
- RNF Requerimientos no Funcionales
- RPS Requerimientos de pruebas del sistema

Además se establecerán las relaciones o dependencias “Trace To” o “Trace From” de cada requerimiento permitiendo identificar aquellos que impactan en el resto de ellos cuando sufran cambios.

Para el caso de estudio la herramienta propuesta a utilizar será el “Reconcile” de Compuware Corporation.

b) Análisis de consistencia de requerimientos

El analista de control de calidad debe realizar el análisis de consistencia de los requerimientos definidos en el Formato F2, para ello debe realizar las siguientes tareas :

- Revisar el catálogo de requerimientos Funcionales y No funcionales del sistema.
- Validar que los requerimientos de software tengan una explicación clara, precisa y completa del problema que facilite el análisis de errores y la generación de casos de prueba.
- Se aplicarán los criterios de consistencia, completitud y ambigüedad :
 - **Completitud** : Grado en que los requerimientos abarcan el alcance del problema. Es decir no debe existir requerimientos omitidos.
 - **Carencia de ambigüedad** : Un único requerimiento debe tener una única interpretación.
 - **Consistencia** : Ausencia de requerimientos contradictorios. El modelamiento del negocio debe ser coherente con el criterio de legalidad.
 - **Legalidad** : El requerimiento no contradice las leyes y normas tributarias o las circulares de la organización.
- Para la evaluación de requerimientos el criterio de legalidad estará comprendido dentro del criterio de consistencia.

Calificación de Requerimientos CR:

El proceso de evaluación se basa en los siguientes puntajes :

1.0 = Puntaje para los requerimientos consistentes.

0.5 = Puntaje para los requerimientos no claros, o ambiguos.

0.0 = Puntaje para requerimientos inconsistentes.

Para todos los requerimientos existentes se calculan sus puntajes según tabla anterior y se suman estos puntajes obteniéndose el valor CR.

Indicador de Requerimientos IF1 : Se calculará este indicador mediante un ajuste de la calificación de requerimientos CR, el ajuste realizado será sobre el valor 10 en base a la siguiente fórmula :

$$IF1 = (CR*10)/Cantidad_total_requerimientos$$

Si existen requerimientos omitidos o inconsistentes se aplicará el ajuste al valor 5 de la siguiente manera :

$$IF1 = (CR*5)/Cantidad_total_requerimientos$$

Criterio de Aprobación : Se aprobará el análisis de requerimientos sólo si el indicador de requerimientos es mayor a 5.0. La corrección será responsabilidad del analista de sistemas y el líder usuario.

c) Validación de los modelos a través del catálogo de requerimientos.

- Se debe realizar la comprobación de modelos, verificando que satisfagan los requisitos trazados en el catálogo de requisitos.
- Determinación de la validación de los modelos, si es aceptable o no.
- Coordinación de la actualización de los modelos, en el caso de que se consideren errores en la validación de los modelos.

Diagrama de Casos de Uso CU : Se valida que los modelos de CU esté técnicamente correcto y sin ambigüedad..

Validaciones para determinar si el CU es Correcto : Deberá determinarse si la descripción del caso de uso y el flujo de eventos es clara y consistente,

notación correcta, actores, secuencias de Interacción correctas (Generalización, Inclusión, Extensión).

Criterio de ambigüedad de CU: Se verifica la ambigüedad en nombres de actores, descripción clara de los caso de uso y flujo de eventos y la relación entre ellos.

Subsistema de análisis : Se verifica que el diagrama de paquetes esté técnicamente correcto y sin ambigüedad,

Validaciones para determinar si el Subsistema de Análisis es Correcto : se valida la notación y se verifica su completitud, es decir que no se omita ningún sub-sistema relevante.

Criterio de ambigüedad de Subsistemas de análisis : Nombres de subsistemas ambiguos o con mas de una interpretación.

Especificación dinámica de la Interface EDI : Se verifica que el diagrama de navegación esté técnicamente correcto, consistente y completo. Se valida la notación, se verifica que la navegación del sistema cumpla con los requerimientos del usuario, los objetos de interface y los mensajes deben estar completos y ser consistentes con los requerimientos Funcionales y No Funcionales del sistema. Se comprueba que cada elemento que active la navegación entre pantallas, debe estar asociado con una mensaje del diagrama de interacción de objetos

Criterio de ambigüedad de EDI : Nombres de mensajes y nombre de objetos de interface ambiguos o no claros.

Diagrama de Interacción DI: Se verifica que los diagramas de Realización o Colaboración estén técnicamente correctos, consistentes y completos. Se valida la correcta notación del diagrama, la interacción de objetos debe ser

consistente con la descripción de flujo de eventos de los C.U., no debe omitirse ningún objeto relevante en el sistema, todos los enlaces entre objetos deben tener un mensaje.

Criterio de ambigüedad de DI : Nombres de mensajes y nombre de objetos ambiguos o no claros.

Análisis de Clases : Se verifica que los diagramas de clases estén técnicamente correctos, consistentes y completos. Se valida la correcta notación del diagrama (nombre de clases, atributos, operaciones, relación). Se verifica que no se omita ninguna clase de control o de entidad relevante en el sistema. Deben ser consistente y satisfacer los requerimientos Funcionales y No Funcionales del sistema. Se verifica que la relación entre clases sea consistente.

Criterio de ambigüedad de análisis de clases : Nombre de clases, atributos, operaciones, relaciones ambiguos.

Calificación de Modelos de Requerimientos CR2:

El proceso de evaluación se basa en los siguientes puntajes :

1.0 = Puntaje para el modelo válido sin errores.

0.5 = Puntaje para modelos parcialmente correctos o ambiguos.

0.0 = Puntaje para modelos incorrectos.

Para todos los requerimientos existentes se calculan su puntaje según tabla anterior y se suman estos puntajes obteniéndose el valor CR2.

Para todos los modelos existentes se calculan su puntaje según tabla anterior y se suman estos puntajes obteniéndose el valor CR2.

Indicador de Conformidad de Modelos IF2 : Se calculará este indicador mediante un ajuste a la calificación de requerimientos CR2, el ajuste realizado será sobre el valor 10 en base a la siguiente fórmula :

$IF2 = (CR2*10)/Cantidad_total_modelos_revisados$ <p>Si existen modelos omitidos o inconsistentes se aplicará el ajuste al valor 5 de la siguiente manera :</p> $IF1 = (CR2*5)/Cantidad_total_modelos_revisados$ <p>Criterio de Aprobación : Se aprobará los modelos si el indicador de conformidad del modelo es mayor a 5. La corrección será responsabilidad del analista de sistemas y el líder usuario.</p> <p>Al final de esta actividad se prepara el informe de verificación.</p>	
COLABORADORES	
Participación	Responsable
<ul style="list-style-type: none"> • Analista de Control de Calidad • Analista de sistemas • Equipo de usuarios 	<ul style="list-style-type: none"> • Analista de Control de Calidad
HERRAMIENTAS BTO	
<ul style="list-style-type: none"> • Compuware Reconcile • Compuware QADirector 	

3.5.1.2 ASI 10: Definición del alcance de las pruebas

GENERALIDADES	
Objetivo :	Después de :
<p>El objetivo de esta tarea es la definición o recopilación de los requisitos relativos al entorno de pruebas, completando así el plan de pruebas. Es decir de definen las especificaciones necesarias para la correcta ejecución de las distintas pruebas del sistema de información.</p>	<p>Análisis de Consistencia y Especificación de Requisitos (ASI 9)</p>
	Antes de :

DESARROLLO	
Entrada	Salida
<ul style="list-style-type: none"> • Informe de Definición F2 v.1.x • Formulación de Proyecto v1.x 	<ul style="list-style-type: none"> • Plan de Pruebas (anexo 6) • Cronograma inicial
PROCEDIMIENTO	
<ul style="list-style-type: none"> • Se definen los planes de prueba y se registran en la herramienta "Reconcile" de Compuware Corporation. Un plan de pruebas es un producto formal que define los objetivos de la prueba de un sistema, establece y coordina una estrategia de trabajo y provee del marco adecuado para elaborar una planificación paso a paso de la actividades de prueba, identifica las tareas de las pruebas a ser ejecutadas, el personal responsable y los riesgos asociados con este plan. • El plan de prueba se va completando y detallando a medida que se avanza en las restantes fases del ciclo de vida del software, Modelamiento de Tecnología, Construcción y Pruebas del sistema de información e Implantación del sistema. • El plan de pruebas a implementar está basado en los estándares propuestos por la IEEE Std 829 que se compone de los siguientes elementos : <ul style="list-style-type: none"> Identificador : Único identificador asignado al plan de pruebas Descripción : Se describen brevemente el objetivo y las características del software a ser probados. Características a ser probados : Identificar todas las características del software a ser probados. Diseñar un caso de prueba en un nivel de abstracción mas alto asociado con las características del software. Características a no ser probadas : Identificar todas las características del software que por alguna razón no podrían ser probados. Actividades : Identificar todas las tareas a realizar en las pruebas y las estimaciones de tiempo de cada una de ellas. Técnicas de Pruebas : Se plantean los siguientes niveles de pruebas : <ul style="list-style-type: none"> ➤ <i>Pruebas Unitarias.</i> 	

- *Pruebas de integración*, que comprenden verificaciones asociadas a grupos de componentes, tienen por objetivo verificar el correcto ensamblaje entre los distintos componentes.
- *Pruebas del sistema*, son pruebas de integración del sistema de información mas completo. Permiten probar el sistema en su conjunto y con otros sistemas con los que se relaciona para verificar que las especificaciones funcionales y técnicas se cumplen.
- *Pruebas de aceptación*, son pruebas dirigidas a validar que el sistema cumple los requerimientos y conseguir la aceptación por parte del usuario final de operación.
- *Pruebas de implantación*, son pruebas que permiten asegurar que el sistema funcionará correctamente en el entorno de operación.

Las pruebas funcionales o de caja negra, las pruebas de caja blanca y las pruebas de carga o de esfuerzo son técnicas no excluyentes que son aplicadas a las pruebas unitarias, pruebas de integración y pruebas del sistema,

Según acuerdo entre las Areas de Desarrollo y la DCC, las pruebas unitarias son obligatoriamente ejecutados por las Areas de Desarrollo antes de que el sistema sea entregado a la DCC para las pruebas correspondientes. Las pruebas funcionales o de caja blanca son características intrínsecas a las pruebas unitarias.

Herramientas : Son las herramientas que se emplearán para las pruebas del software.

Criterios de Aprobación : Especificar el criterio para determinar cuando un Test es aprobada o desaprobada. Por ejemplo para un determinado software de uso interno la usabilidad orientada a la calidad externa no será prioritaria, entonces podrá darse por aprobada sin demasiadas exigencias en este aspecto.

En cualquier caso se aprobará las pruebas cuando se dejan de encontrar errores.

Criterios de suspensión : Las especificadas en CPS10.

Entregables de las pruebas : Los entregables al finalizar las pruebas son el plan de pruebas (anexo 6), la especificación de casos de pruebas (anexo 7), el reporte de incidencias (anexo 8), el log de pruebas entre otros como son el informe de pruebas y el informe de aceptación.

En el log de pruebas, la estructura mínima recomendada por la IEEE std 829 es la siguiente : Identificador, Descripción de actividades, eventos producidos. Se registrará los resultados del proceso visualmente observable adjuntando pantallas del evento exitoso o los mensajes de errores producidos o abortados, se registrará el día y hora de los eventos ejecutados y los resultados obtenidos. Es un nivel mas detallado de los casos de prueba, los mismos que serán registrados en la Herramienta Track Record de Compuware o algún otro repositorio similar.

Necesidades de entorno : De Definición de los requisitos de configuración de entorno: Hardware, comunicaciones, software base, niveles de seguridad que deben ser provistas para realizar las pruebas, librerías, bases de datos, ficheros, procesos, necesidades de almacenamiento, configuración de accesos, configuración de uso de equipo en stand alone o red, etc.

Participantes : Se identifican los grupos o personas que participarán interactuando durante las pruebas, preparando, diseñando, dando testimonio, verificando, resolviendo y preparando el entorno necesario.

Staff necesario para entrenamiento :

Identificar el staff necesario para brindar entrenamiento al personal participante de ser necesario.

Cronogramas :

Se identifican los hitos en el proyecto. Se distribuye el tiempo de cada actividad de prueba en una cronograma inicial, que va ir mejorando por etapas. Se define el tiempo de las actividades de las pruebas de aceptación.

Riesgos y Contingencias :

Identificar los riesgos asumidos en el plan de pruebas. Especificar el plan de contingencia (Ejemplo un retraso en las pruebas pueden requerir realizar las pruebas en horario nocturno)

Aprobaciones : Especificar los nombres y cargos de las personas que aprueban este plan, firmas y fechas.

- La actividad ASI 10 requiere que el plan de pruebas tenga las siguientes especificaciones :

- Descripción y objetivos de las pruebas
- Actividades y tiempos estimados
- Definición inicial de requisitos del entorno de pruebas
- Participantes
- Staff necesario para capacitaciones
- Cronograma inicial
- Riesgos y contingencias

COLABORADORES

Participación	Responsable
• Analista de Control de Calidad	• Analista de Control de Calidad

HERRAMIENTAS BTO

- Compuware Reconcile
- Compuware QADirector

Procedimiento en la fase de Modelamiento de Tecnología

3.5.1.3 DSI 9: Diseño de la Migración y Carga Inicial de Datos

GENERALIDADES

Objetivo :	Después de :
El objetivo de esta tarea es definir el plan de migración y carga inicial, concretando el plan de trabajo de acuerdo a los procedimientos y procesos y migración y carga inicial definidos	Verificación y aceptación de la arquitectura del sistema (DSI 7)
	Antes de : Especificación Técnica del Plan de Pruebas (DSI 11)

DESARROLLO

Entrada	Salida
<ul style="list-style-type: none"> Informe de definición F2 incluyendo el requerimiento de Migración y Carga Inicial de Datos 	<ul style="list-style-type: none"> Plan de Migración y Carga Inicial de Datos Planificación de Migración y Carga Inicial
PROCEDIMIENTO	
<ul style="list-style-type: none"> Determinación de los recursos necesarios Estimar los recursos humanos que se requerirán Establecer los procedimientos y procesos de migración y carga inicial de datos, para planificación 	
COLABORADORES	
Participación	Responsable
<ul style="list-style-type: none"> Analista de Sistemas Analista de Soporte Técnico Analista de Telecomunicaciones Analista de Atención a usuarios Analista de Seguridad Informática Analista de Control de Calidad 	<ul style="list-style-type: none"> Analista de Sistemas

3.5.1.4 DSI 11: Especificación Técnica del Plan de Pruebas

3.5.1.4.1 DSI 11.1: Especificación del Entorno de Pruebas

GENERALIDADES	
Objetivo :	Después de :
El objetivo de esta tarea es la definición detallada y completa del entorno necesario para la realización de las pruebas del sistema: unitarias, de integración, de implantación y de aceptación.	Diseño de la Migración y Carga Inicial de Datos (DSI 9)
	Antes de : Especificación Técnica de Prueba (DSI 11.2)

DESARROLLO	
Entrada	Salida
<ul style="list-style-type: none"> • Informe de Definición F2 v2.x • Formulación del Proyecto v2.x • Diseño de la Arquitectura del Sistema • Entorno Tecnológico del Sistema • Modelo Físico de Datos Optimizado • Esquemas Físicos de Datos • Asignación de Esquemas Físicos de Datos a Nodos • Especificaciones de Construcción del Sistema de Información 	<ul style="list-style-type: none"> • Plan de pruebas con la especificación detallada del entorno de pruebas (anexo 6).
PROCEDIMIENTO	
<ul style="list-style-type: none"> • Establecer el entorno tecnológico: hardware, software y comunicaciones en el entorno de pruebas necesarios para asegurar que el sistema funcionará en el entorno de operación del usuario final (producción). • Determinar las restricciones técnicas del entorno. Si algunos de los entornos tecnológicos no pudiese adecuarse a las pruebas necesarias, se solicitará el apoyo de los siguientes encargados : <ul style="list-style-type: none"> ➤ El analista de Soporte Técnico quien es el encargado de verificar y de proveer de los equipos de hardware y software base, configurar los pools de conexión, otorgar accesos a las bases de datos, accesos de cuentas de usuario al dominio de las redes, o de configurar laboratorios de pruebas en el caso de que la DCC no disponga de un ambiente similar al operacional. ➤ El Analista de Telecomunicaciones quien es el encargado de proponer y asegurar que los modelos sean óptimos para no afectar las comunicaciones, aperturar puertos o nodos de acceso, permisos a las 	

DMZ, al Firewall, etc.

- Determinar los **requisitos de operación y seguridad del entorno de pruebas** para lo cual se coordinará también con los analistas de Soporte Técnico y de Telecomunicaciones, quienes además deben verificar y velar por el cumplimiento de las políticas de seguridad de la Institución.
- La Herramienta "File AID C/S" de Compuware Corporation, así como otras herramientas de recuperación (programas TCU06 y TCU07) para la vuelta atrás de las pruebas (BaseLine) deben considerarse en la preparación del entorno de pruebas.
- Las especificaciones del entorno de pruebas conjuntamente con el plan de pruebas deberán ser registradas en la Herramienta "Reconcile" de Compuware Corporation.

COLABORADORES

Participación	Responsable
<ul style="list-style-type: none">• Analista de Control de Calidad• Analista de Sistemas• Analista de Soporte Técnico• Analista de Telecomunicaciones	<ul style="list-style-type: none">• Analista de Control de Calidad

HERRAMIENTAS BTO

- Compuware Reconcile
- File AID C/S

3.5.1.4.2 DSI 11.2: Especificación Técnica de las Pruebas

GENERALIDADES	
Objetivo :	Después de :
El objetivo de esta tarea es el diseño detallado de los casos de pruebas. Además se validan los modelos de la fase de construcción de la metodología.	Especificación del Entorno de Pruebas (DSI 11.1)
	Antes de :
	Reunión de Gestión (DSI 12)
DESARROLLO	
Entrada	Salida
<ul style="list-style-type: none"> • Informe de Definición v2.x • Formulación del Proyecto v2.x • Diseño de la Arquitectura del Sistema • Entorno Tecnológico del Sistema • Diseño Detallado de los Subsistemas de Soporte • Modelo Físico de Datos Optimizado • Esquemas Físicos de Datos • Asignación de Esquemas Físicos de Datos a Nodos • Diseño de Interface de Usuario • Especificaciones de Construcción del Sistema de Información • <i>Diseño Orientado a Objetos:</i> • Diseño de la Realización de los Casos de Uso • Modelo de Clases de Diseño • Comportamiento de Clases de 	<ul style="list-style-type: none"> • Plan de Pruebas (anexo 6) • Especificación de casos de pruebas. (anexo 7) • Informe de verificación (anexo 9)

Diseño	
PROCEDIMIENTO	
<p>Se especifican los aspectos funcionales y no funcionales del sistema, tales como:</p> <ul style="list-style-type: none"> • Casos de prueba asociados : se definen en detalle los casos de prueba en relación a la tecnología, hardware y software, es decir, según la arquitectura de comunicación, tiempos de respuesta, volumen de información, concurrencia. • La definición de casos de pruebas deben estar almacenados en el repositorio centralizado que provee la Herramienta “Compuware Reconcile”. • Requisitos especiales o acciones necesarias para iniciar la ejecución. • Se prepara el plan de implantación inicial. <p>Se validan los siguientes modelos del diseño del Sistema de Información :</p> <p>Diagrama de Arquitectura : se valida el cumplimiento del diagrama de arquitectura, tiene influencia en la preparación del entorno de prueba.</p> <p>Catalogo de subsistemas : Se verifica la consistencia y no ambigüedad en la descripción de los sistemas.</p> <p>Catalogo de excepciones : Este catálogo tiene influencia en el indicador de fiabilidad, por consiguiente no se le considera para la evaluación del indicador de modelamiento de tecnología IM1; sin embargo, el concepto prevalece en la fase de Tecnología de la MDSI. Se determina si existen excepciones no precisadas, excepciones no controladas. Se recomienda un análisis de riesgo de ocurrencia de fallos. Se debe definir procedimientos de emergencia o recuperación.</p> <p>Diseño de casos de uso reales : Se verifica que el diagrama de CU reales esté técnicamente correcto.</p> <p>Tiene influencia en el indicador de a</p>	

Analizabilidad, es decir que tan analizable es el software a partir del diseño de CU reales.

Diagrama de clases : Se verifica que el diagrama de clases esté técnicamente correcto.

Calificación de Modelos de Tecnología CR3:

El proceso de evaluación se basa en los siguientes puntajes :

1.0 = Puntaje para el modelo válido sin errores.

0.5 = Puntaje para modelos parcialmente correctos o ambiguos.

0.0 = Puntaje para modelos incorrectos.

Para todos los requerimientos existentes se calculan su puntaje según tabla anterior y se suman estos puntajes obteniéndose el valor CR3.

Para todos los modelos existentes se calculan su puntaje según tabla anterior y se suman estos puntajes obteniéndose el valor CR3.

Indicador de Modelamiento de Tecnología IM1 : Se calculará este indicador mediante un ajuste a la calificación de requerimientos CR3, el ajuste realizado será sobre el valor 10 en base a la siguiente fórmula :

$$IM1 = (CR3*10)/Cantidad_total_modelos_revisados$$

Si existen modelos omitidos o inconsistentes se aplicará el ajuste al valor 5 de la siguiente manera :

$$IM1 = (CR3*5)/Cantidad_total_modelos_revisados$$

Criterio de Aprobación : Se aprobará los modelos si el indicador es mayor a 5.0. La corrección será responsabilidad del analista de sistemas.

- La actividad DSI 11.2 requiere que el plan de pruebas tenga las siguientes especificaciones :

- Características del software que se probarán y aquellas que no también.

- Técnicas de pruebas.

- Herramientas de pruebas a utilizar.
- Criterios de aprobación y suspensión de las pruebas.
- Cronograma final.
- Ajustes al plan de pruebas inicial de ser necesario.

COLABORADORES

Participación	Responsable
<ul style="list-style-type: none"> • Analista de Control de Calidad • Analista de Sistemas 	<ul style="list-style-type: none"> • Analista de Control de Calidad

HERRAMIENTAS BTO

- Compuware Reconcile

Procedimiento en la construcción y Pruebas del Sistema

3.5.1.5 CPS 5 – CPS 6: Ejecución de las Pruebas Unitarias y las Pruebas de Integración.

GENERALIDADES	
Objetivo :	Después de :
Las Gerencias de Desarrollo de Sistemas son responsables del cumplimiento de la Metodología MDSI, el personal a su cargo realizarán únicamente las pruebas unitarias de los sistemas de información y prepararán el informe de pruebas unitarias respectivo.	Creación de Código (CPS 2)
	Antes de : Revisión de Estándares de Construcción (CPS 7).
DESARROLLO	
Entrada	Salida
<ul style="list-style-type: none"> • Estándares de programación • Informe de Definición F2 V.x 	<ul style="list-style-type: none"> • Informe de Pruebas Unitarias
PROCEDIMIENTO	
<ul style="list-style-type: none"> • Debe ser responsabilidad del coordinador del proyecto verificar el cumplimiento del uso de la Metodología MDSI por parte del personal a su cargo. • Se deberán utilizar los estándares de programación aprobados por la Oficina de Investigación de Procesos de Software OIPS. • Para verificación de los estándares de programación de aplicativos basados en el lenguaje de programación Visual Basic se utilizará la herramienta Compuware DevPartner Studio o similar, para aplicativos basados en el lenguaje de programación Java se utilizará la herramienta Compuware DevPartner Java Edition o similar, ambas permitirán identificar sentencias de programación no óptimas o que escapen de los estándares de programación establecidos por la Institución o fuera del 	

estándar recomendado por el lenguaje de programación.

- Otros aplicativos SUNAT basados en el lenguaje de programación Fox, Visual Fox y Power Builder deberán de cumplir con los estándares de programación que la Oficina de Ingeniería y Procesos de Software OIPS precise.
- La Herramienta Compuware DevPartner Studio o similar, proveerá para su uso las sentencias recomendadas por Microsoft Corporation y en el caso de Compuware DevPartner Java Edition o similar deben ser capaces de registrar los estándares que la OIPS determine.
- Compuware DevPartner Studio y Compuware Devpartner Java Edition o sus similares brindarán un análisis de trazabilidad automático respecto a la ejecución de los casos de pruebas, permitiendo identificar las posibles condiciones, funciones o módulos no recorridos por el caso de prueba.
- En la medida que la División de control de calidad implemente los Scripts necesarios sobre los sistemas ya probados y ya se disponga de ellos, los mantenimientos realizados al sistema podrán utilizar en sus pruebas unitarias estos Scripts mediante la herramienta QARun para verificar que los cambios realizados no impacten sobre alguna otra funcionalidad no deseada.
- Para las pruebas de Carga el analista programador dispondrá de la herramienta "Compuware QALoad".
- Para el seguimiento del error se utilizará alguna herramienta work flow tal como el "Compuware Track Record" u otro similar.
- El analista de sistemas y/o el programador serán los responsables de realizar las pruebas unitarias del sistema para lo cual deberán preparar el informe de pruebas con la siguiente información :
 - Nombre del Coordinador responsable
 - Nombre del Programador responsable
 - Por cada regla del Negocio según el MDSI, indicar los casos de pruebas, datos de entrada y respuesta esperada por el sistema.

<ul style="list-style-type: none"> ○ Listado de subsistemas, módulos u opciones del sistema posiblemente afectados por los cambios o nuevas funcionalidades implementadas. ○ Adjuntar pantallas de los módulos probados. <ul style="list-style-type: none"> ● Las pruebas unitarias y de integración que comprenden verificaciones asociadas a grupos de componentes deben ser realizadas por la Areas encargadas del Desarrollo del Sistema de Información. ● Las Pruebas del Sistema serán realizadas por el Analista de Calidad. ● Nótese que se propone el uso de las herramientas BTO para el personal de desarrollo, toda vez que permitirán encontrar posibles errores antes de remitir el software a la División de Control de Calidad, conduciendo al ahorro en tiempo y costos del proyecto. 	
COLABORADORES	
Participación	Responsable
<ul style="list-style-type: none"> ● Analista de Sistemas ● Programador de Sistemas ● Analista de Calidad 	<ul style="list-style-type: none"> ● Analista de Sistemas
HERRAMIENTAS BTO	
<ul style="list-style-type: none"> ● Compuware QARun ● Track Record ● QALoad ● Compuware DevPartner Studio ● Compuware DevPartner Java Edition 	

3.5.1.6 CPS 7: Revisión de estándares de construcción

GENERALIDADES	
Objetivo :	Después de :
El objetivo de esta tarea es verificar el cumplimiento de los estándares de	Ejecución de las Pruebas de Integración (CPS 6)

programación y modelo de datos.	Antes de :
	Pruebas del Sistema (CPS 10)
DESARROLLO	
Entrada	Salida
<ul style="list-style-type: none"> • Programa fuente • Plan de pruebas 	<ul style="list-style-type: none"> • Estándares validados. • Informe de verificación
PROCEDIMIENTO	
<ul style="list-style-type: none"> • Verificar el cumplimiento de los estándares de programación es responsabilidad del Analista de Calidad . • Verificar el cumplimiento de los estándares de modelamiento de datos. Esta tarea es responsabilidad del Administrador de Datos, sin embargo, el Analista de Calidad ha de reportar a éste cuando un sistema ha modificado la estructura de su modelo sin haber comunicado de los cambios al administrador encargado, la tarea se realiza al instalar el pase a producción en el ambiente de pre-producción (entorno de pruebas) utilizadas para realizar las pruebas de la DCC. • De existir estándares del lenguaje de programación del software materia de verificación y que sean formalmente aprobados en la INSI, se validará su cumplimiento, caso contrario, se registrará la no existencia de éste en el momento de su calificación. • Herramientas disponibles para la depuración automática para la verificación de nomenclaturas de estándares y de optimización de sentencias serán el software Compuware DevPartner Studio y Compuware DevPartner Java o cualquier similar. Éstas herramientas pueden ser utilizadas en la fase de desarrollo del sistema, de ser así se adjuntará el informe de verificación de estándares automático, debidamente visado por el Coordinador del proyecto del Área de Desarrollo. 	
COLABORADORES	
Participación	Responsable
<ul style="list-style-type: none"> • Analista Control de Calidad 	<ul style="list-style-type: none"> • Analista de Control de Calidad

<ul style="list-style-type: none"> • Programador • Administrador de datos 	
HERRAMIENTAS BTO	
<ul style="list-style-type: none"> • Compuware DevPartner Studio • Compuware DevPartner Java Edition 	

3.5.1.7 CPS 8: Evaluación de Procedimientos Normativos

GENERALIDADES	
Objetivo :	Después de :
<p>El objetivo de esta tarea es asegurar la consistencia entre una propuesta de Circular y el Modelo de Negocio. En caso de existir discrepancias que no pueden ser resueltas, la decisión de aprobación la tomará la Intendencia.</p>	<p>Elaboración de procedimientos normativos (CPS 4)</p>
	<p>Antes de :</p> <p>Elaboración de manuales de usuario (CPS 8)</p>
DESARROLLO	
Entrada	Salida
<ul style="list-style-type: none"> • Modelo de Negocio • Propuesta de Circular 	<ul style="list-style-type: none"> • Propuesta de Circular validada
PROCEDIMIENTO	
<ul style="list-style-type: none"> • Validar la Propuesta de Circular conforme al Modelo de Negocio. • Encontrar observaciones de consistencia. • Actualizar las propuesta de Circular 	
COLABORADORES	
Participación	Responsable
<ul style="list-style-type: none"> • Analista Control de Calidad • Administrador de Procesos 	<ul style="list-style-type: none"> • Analista de Control de Calidad

3.5.1.8 CPS 10: Pruebas del Sistema

GENERALIDADES	
Objetivo :	Después de :
El objetivo de esta tarea es preparar todos los recursos necesarios para realizar las pruebas del sistema y comprobar la integración de todos los sub-sistemas y componentes del sistema de información	Evaluación de Procedimientos (CPS7)
	Antes de :
	Pruebas de Aceptación del Sistema (CPS 12)
DESARROLLO	
Entrada	Salida
<ul style="list-style-type: none"> • Informe de Definición F2 v3.x • Formulación de Proyecto v3.x • Solicitud de Pase a Producción • Plan de pruebas • Casos de prueba 	<ul style="list-style-type: none"> • Reporte de incidencias (anexo 8) • Informe de verificación (anexo 9) • Informe de pruebas
PROCEDIMIENTO	
<ul style="list-style-type: none"> • Se revisan y ejecutan los casos de prueba definidos en DSI 11.2. • Verificar las entradas de las pruebas. Se utilizarán los datos de entrada de los casos de prueba o se prepararán otros de ser necesario. • Verificar las ejecuciones. • Verificar las salidas de las pruebas. • Informar los resultados de las pruebas. • Registrar los resultados de los errores encontrados a través de la herramienta work flow para el seguimiento del error "Compuware Track Record". • Registrar estadísticas de la prueba realizada. • Se prepara el informe de verificación. • Se prepara el reporte de incidencias. 	

- Para la ejecución de la prueba funcional, de ser necesario se convocará a los usuarios expertos del negocio con la finalidad de generar una participación activa.
- Para las pruebas de esfuerzo, volumen, comportamiento y almacenamiento es necesario coordinar con el Departamento de Soporte Técnico y el Departamento de Telecomunicaciones con la finalidad de apoyar en la verificación del comportamiento del sistema informático con la ejecución de estos tipos de pruebas.
- Para las pruebas de seguridad del sistema informático, se coordinará con la Oficina de Seguridad Informática para contar con el apoyo técnico respectivo.
- El reporte de errores se comunicará al responsable encargado a través de la herramienta "Compuware Track Record".

Las pruebas del Sistema deberán ser ejecutadas por la DCC, luego de determinar el alcance de las pruebas y sin excepción de lo definido en ASI 10 se ejecutarán las siguientes pruebas :

- a) Pruebas Funcionales o de Caja Negra
- b) Pruebas de Carga o Stress
- c) Pruebas de Caja Blanca

a) Pruebas Funcionales

Para el apoyo en las pruebas funcionales el Analista de Calidad utilizará las herramientas manuales que dispone, así como también las herramientas BTO con el producto "Compuware QARun" o similar, que permitirá realizar pruebas automatizadas a través de los Scripts que en un 90% son generados automáticamente.

Se debe tomar en cuenta lo que comúnmente se conoce como fracasos falsos, producidos debido a diferencias simples, inocuas que pueden

ocasionar una falla de comparación de Bitmaps o ejecutando programas en máquinas con resoluciones diferentes u otros cambios cosméticos, que podrían enviar un mensaje de error al reproducir el Script.

b) Pruebas de Carga o Stress

Para el apoyo de las pruebas de carga o stress se dispondrá de la Herramienta “Compuware QALoad” la cual permitirá realizar las siguientes acciones :

- Emular, desde un punto central, cualquier número de usuarios y su impacto en los sistemas
- Ajustar niveles de volumen de usuarios virtuales
- Identificar y corregir problemas de escalabilidad tempranamente.
- Medir y analizar los tiempos de respuesta de las aplicaciones y utilización de sistemas sometidos a cargas similares a producción.
- Variabilizar datos de entrada.
- Correlacionar tiempos de respuesta pobres con el desempeño de los servidores.

Análisis de cuellos de botella : Se dispondrá de la Herramienta Compuware Application Vantage la cual permitirá :

- Identificar los cuellos de botella.
- Resolver problemas de desempeño de las aplicaciones en ambientes de pre-producción y producción.
- Identificar problemas de desempeño de las aplicaciones, a nivel cliente, red, servidor o la misma aplicación.
- Validar acciones correctivas utilizando sus características de análisis predictivo.
- Certificar el éxito de desarrollos nuevos y modificados de las aplicaciones.
- Realizar el análisis de trazabilidad e identificación de Querys no óptimos.

Horarios de Pruebas de Carga : Se debe tener en cuenta para la carga y

análisis de los cuellos de botella los horarios de máxima concurrencia, para tener resultados mas próximos a la realidad.

Los horarios de máxima concurrencia en SUNAT son :

11:00 a.m. a 12:00 m

15:00 p.m. a 16:00 p.m.

17:00 p.m. a 18:00 p.m. en calendarios de vencimientos.

c) Pruebas de Caja Blanca

Análisis de Trazabilidad : Este análisis permitirá medir el porcentaje e identificación de funciones, procedimientos, o código fuente ejecutado con los casos de pruebas utilizados. Su importancia radica en que dicha información permitirá medir la cobertura de código ejecutado en las pruebas, si ésta medida es muy baja, la posibilidad de existencia de errores por el código no probado es alta y viceversa; de ser necesario los casos de pruebas deben redefinirse para ampliar la cobertura de código. Las herramientas a utilizar serán el Compuware DevPartner Studio para los aplicativos desarrollados en el Lenguaje Visual Basic y el Compuware DevPartner Java Edition para aplicativos en lenguaje Java.

Criterio de fin de las Pruebas

Al principio del plan de prueba, se planea la cantidad de pruebas que se deben efectuar. El proceso de pruebas del sistema se detendrá por cualquiera de los siguientes motivos :

- Cuando se termina de ejecutar todos los casos de pruebas y se dejan de encontrar errores.

Criterio de suspensión de las Pruebas

Las pruebas se darán por suspendidas por cualquiera de las siguientes razones :

- Cuando la documentación remitida no contenga información suficiente para la realización de las pruebas.

<ul style="list-style-type: none"> • Cuando los errores reportados al Area de Desarrollo no tienen respuestas y han transcurrido mas de 2 días hábiles. • Cuando se detectan errores reiterativos del software que hacen suponer que no se han realizado las suficientes pruebas unitarias. <p>En cualquiera de los casos anteriores se devuelve el trámite al Área de Desarrollo para que se revisen detenidamente las observaciones reportadas.</p>	
COLABORADORES	
Participación	Responsable
<ul style="list-style-type: none"> • Analista Control de Calidad • Analista de Soporte Técnico • Analista de Sistemas 	<ul style="list-style-type: none"> • Analista de Control de Calidad
HERRAMIENTAS BTO	
<ul style="list-style-type: none"> • Compuware QARun • Track Record • QADirector • File Aid C/S • QALoad • Application Vantage • Compuware DevPartner Studio • Compuware DevPartner Java Edition 	

3.5.1.9 CPS 11: Pruebas de Aceptación del sistema

GENERALIDADES	
Objetivo :	Después de :
El objetivo de esta tarea es determinar si están completos o no los caso de prueba de aceptación y evaluar los resultados de la prueba de aceptación.	Pruebas del Sistema (CPS 10)
	Antes de :
	Reunión de Gestión
DESARROLLO	

Entrada	Salida
<ul style="list-style-type: none"> • Plan de pruebas • Modelo de Procesos de Negocio • Catálogo de Requerimientos • Solicitud de Pase a Producción 	<ul style="list-style-type: none"> • Informe de Aceptación
PROCEDIMIENTO	
<ul style="list-style-type: none"> • Se analizan los criterios de aceptación establecidos por el usuario y recogidos en las verificaciones del plan de pruebas, por si fuera necesario incorporar algún caso de prueba adicional. • Se comunica el plan de pruebas de aceptación a los usuarios implicados según los distintos perfiles identificados. • Se ejecutan y evalúan los resultados obtenidos con los esperados. Identificar el origen de cada problema para poder remitirlo a quién proceda y determinar qué acciones o medidas correctoras es preciso llevar a cabo para resolverlo de forma satisfactoria. • Indicar qué pruebas se deben volver a realizar, o si será necesario contemplar nuevos casos de prueba. • Una vez realizadas las medidas correctoras necesarias, y comprobado que su comportamiento es adecuado, se documenta el resultado global de la evaluación de las pruebas de aceptación que incluye la aprobación del sistema por parte del usuario final (informe de aceptación). • El Analista de Control de Calidad con la información enviada por el Departamento de Operaciones, registrará la fecha final de la instalación del pase con lo cual se concluye la atención del pase a producción. 	
COLABORADORES	
Participación	Responsable
<ul style="list-style-type: none"> • Analista Control de Calidad • Usuarios Expertos, • Equipo de usuarios 	<ul style="list-style-type: none"> • Analista de Control de Calidad

3.5.2 Implementación de Métricas de Calidad de Software

Para la evaluación de la calidad interna y externa del software es necesario implementar una serie de medidas que permitan calificar el software, puesto que no se podrá mejorar un software si éste no puede ser medido. La solución propuesta se basa en los criterios que propone la ISO-9126.

Para la evaluación de las características de la calidad interna y externa del software, se deberá tener en cuenta la importancia de cada una de ellas respecto a la otra, lo que dependerá del tipo de software presentado. Como se ha descrito anteriormente, el software en la SUNAT se desarrolla para dos tipos de clientes : Clientes Directos o usuarios internos y Clientes Indirectos o usuarios externos. La siguiente clasificación se tomará en cuenta para el tipo de software a calificar :

SW1 : Software desarrollado para Clientes Directos de SUNAT

SW2 : Software desarrollado para Clientes Indirectos SUNAT

En el siguiente cuadro se muestra la matriz de calificación del software. Se puede observar los indicadores y el nivel de influencia según el tipo de clasificación del software. Los niveles de influencia se pueden ponderar según información obtenida por medio de encuestas y se propone que sea un valor entre 0 y 5, donde "0" represente un nivel de no influencia y "5" represente un nivel de gran influencia; las columnas de calificación inicial y final tienen el propósito de comparar la evolución de la mejora o disminución de la calidad del software producto de desarrollos o mantenimientos efectuados.

Matriz de Calificación del Software

Características	Sub Características	Indicador (I)	Influencia (P)		Calificación Inicial (I x P)	Calificación Final (I x P)	Evolución de la calidad
			SW1	SW2			
Funcionalidad	Consistencia de requerimientos (ASI 9) : Los requerimientos cumplen con el analisis de consistencia, ambigüedad y completitud de los requerimientos.	IF1					
	Conformidad de modelos de requerimientos (ASI 9) : El Software refleja a través de su modelo el cumplimiento de las necesidades del usuario.	IF2					
	Precisión : Capacidad del software para proveer los resultados de manera precisa	IF3					
	Seguridad : Capacidad del software para proteger la información y los datos, de manera que no puedan leerlos o modificarlos.	IF4					
Eficiencia	Comportamiento en el tiempo : Capacidad del software para proveer de tiempos apropiados de respuesta y procesamiento.	IE1					
	Utilización de Recurso : La capacidad del software para utilizar cantidades de recursos apropiados.	IE2					
	Conformidad de concurrencia : El software cumple con los niveles de concurrencia requeridos.	IE3					
Mantenimiento	Analizabilidad (DSI 11.2) : La capacidad del software para ser diagnosticado por deficiencias o causas de falla, así como la identificación de partes a ser modificadas.	IM1					
	Cambiability : La capacidad del software para permitir que una determinada modificación sea implementada.	IM2					
	Testabilidad : la capacidad del software para permitir que las modificaciones puedan ser validadas.	IM3					
Usabilidad	Conformidad de codificación : Capacidad de cumplimiento de estándares de codificación.	IM4					
	Entendibilidad : Capacidad del software para permitir al usuario entender si el software es aplicable y cómo puede ser utilizado	IU1 - IU2					
	Atractividad : Capacidad del software de ser atractivo al usuario.	IU3					
	Conformidad de usabilidad : El software cumple con los estándares de usabilidad.	IU4					
Fiabilidad	Madurez : Capacidad del software de evitar fallas como resultado de errores en el software.	IC1					
	Tolerancia a Fallos : Capacidad del software para mantener un nivel de funcionamiento en caso de errores de software o en el caso de una interfaz incompleta.	IC2					
	Recuperabilidad : Capacidad del software para reestablecer un nivel de funcionamiento y recuperar datos en caso de fallas.	IC3					
Portabilidad	Portabilidad : Mide la facilidad de instalación del software en un ambiente operacional	IP1					

El siguiente cuadro muestra el cálculo de los indicadores propuestos, se deberá considerar un puntaje de aprobación mayor a 5.0 en una escala de calificación de 0 a 10 :

CRITERIO	CALCULO	INDICADOR
FUNCIONALIDAD		
Consistencia de requerimientos	Indicador de requerimientos (ASI 9)	IF1
Conformidad del modelo	Indicador de conformidad de modelos de requerimientos (ASI 9)	IF2
Precisión	Fórmula : $(10 - (CE * 10 / CCP))$ CE = Cantidad de errores encontrados CCP = Cantidad de casos probados	IF3
Seguridad	Se considera alta seguridad de la información a : Encryptamiento, VPN, SSL, Acceso mediante Login y Password.	IF4
EFICIENCIA		
Comportamiento en el tiempo	Fórmula = $CQO * 10 / (CQO + CQNO)$ CQO = Cantidad de Querys Optimos CQNO = Cantidad de Querys No Óptimos CQT = CQO + CQNO	IE1
Utilizacion de Recurso	Prueba de volumen y comunicaciones 10 : Soporta volumen de datos mayor al 25% de lo esperado 6 : Soporta volumen de datos igual a lo esperado 1 : No soporta volumen de datos esperado. 0 : No se evaluo	IE2
Conformidad de concurrencia	Indicador de concurrencia : 10 : Soporta numero de concurrencia mayor al 25% de lo esperado 6 : Soporta numero de concurrencia igual a lo esperado 1 : No soporta numero de concurrencias esperadas.	IE3
MANTENIMIENTO		
Analizabilidad	Indicador de modelamiento de tecnología (DSI 11.2)	IM1
Cambiabilidad	Verificación de nivel de documentación y claridad de código fuente: 10 : Bueno 6 : Regular 1 : Malo	IM2
Testeabilidad	Verificación de nivel de especificación de flujo de Eventos de CU o miniespecificaciones : 10 : Bueno 6 : Regular 1 : Malo	IM3
Cumplimiento de estándares	10 : Cumple eficientemente estandares de codificación 6 : Cumple medianamente estandares de codificación 1 : No cumple 0 : No se dispone de estandares de codificación	IM4
USABILIDAD		
Entendibilidad	Indicador de entendibilidad	IU1-IU2
Atractividad	Indicador de atractividad	IU3
Cumplimento de usabilidad	10 : Cumple eficientemente estandares de usabilidad 6 : Cumple medianamente estandares de usabilidad 1 : No cumple 0 : No se dispone de estandares de usabilidad	IU4
FIABILIDAD		

Madurez	10 : El software evito la prueba de fallo. (dato inconsistente) 0: No evito Fallo	IC1
Tolerancia a Fallos	10 : Se mantuvo un nivel de funcionamiento al fallar el software. 6 : Se mantuvo un regular nivel de Funcionamiento al fallar el software 1 : Dejo de funcionar	IC2
Recuperabilidad	10: Se recupero los datos despues de Fallo 6 : Recuperacion Parcial 1 : No se recupero	IC3
PORTABILIDAD		
Portabilidad	$IP1=10 \cdot A/BA$ =Número de casos en que el usuario es exitoso en la operación de instalación.B=Número total de casos en que el usuario intenta ejecutar la operación de instalación	IP1

3.5.2.1 Indicadores de Funcionalidad

Característica : Funcionalidad	
Criterio : Cumplimiento de requisitos	
Propósito : Tiene el propósito de medir el cumplimiento de los requerimientos funcionales y no funcionales en términos de consistencia, ambigüedad y completitud de los requerimientos.	
Indicador	Medición o fórmula de cálculo
IF1	<p>Puntaje inicial : Puntaje obtenido del primer documento de definición F2 obtenido en la fase de requerimientos del sistema.</p> <p>Puntaje final : Puntaje obtenido del documento de definición F2 obtenido en la fase de requerimientos del sistema final.</p> <p>N1 = Cantidad de requerimientos funcionales definidos en el sistema. N2 = Cantidad de requerimientos no funcionales definidos en el sistema. A1 = Cantidad de requerimientos funcionales consistentes. A2 = Cantidad de requerimientos no funcionales consistentes. B1 = Cantidad de requerimientos funcionales ambiguos. B2 = Cantidad de requerimientos no funcionales ambiguos. C1 = Cantidad de requerimientos funcionales inconsistentes. C2 = Cantidad de requerimientos no funcionales inconsistentes.</p> <p>$P1 = A1*1.0 + B1*0.5 + C1*0.0$ (Puntaje inicial) $P2 = A2*1.0 + B2*0.5 + C2*0.0$ (Puntaje inicial) $F1 = A1*1.0 + B1*0.5 + C1*0.0$ (Puntaje final) $F2 = A2*1.0 + B2*0.5 + C2*0.0$ (Puntaje final) $P = P1 + P2$ (Puntaje inicial) $F = F1 + F2$ (Puntaje final) $IF1 = P*10/N$ (Indicador Inicial) ó $F*10/N$ (Indicador Final)</p> <p>Ver cuadros de matriz de cálculo de indicadores.</p>
Criterio : Conformidad del modelo	
Propósito : El Software refleja a través de su modelo el cumplimiento de las necesidades del usuario	
Indicador	Medición o fórmula de cálculo
IF2	<p>Puntaje inicial : Puntaje obtenido del primer documento de definición F2 obtenido en la fase de requerimientos del sistema</p> <p>Puntaje final : Puntaje obtenido del documento de definición F2 obtenido en la fase de requerimientos del sistema final.</p> <p>Tipos de diagramas evaluados :</p> <p>Diagramas de caso de uso del modelo equivalente al elemento $i = 1$</p>

	<p>Diagramas de paquetes del modelo equivalente al elemento $i = 2$</p> <p>Diagramas de navegación del modelo equivalente al elemento $i = 3$</p> <p>Diagramas de realización del modelo equivalente al elemento $i = 4$</p> <p>Diagramas de clases del modelo equivalente al elemento $i = 5$</p> <p>$N[i]$ = Cantidad de diagramas del tipo $[i]$, donde $i=[1..5]$</p> <p>$A[i]$ = Cantidad de diagramas correctamente diseñados del tipo $[i]$, donde $i=[1..5]$</p> <p>$B[i]$ = Cantidad de diagramas ambiguamente diseñados del tipo $[i]$, donde $i=[1..5]$</p> <p>$C[i]$ = Cantidad de diagramas incorrectamente diseñados del tipo $[i]$, donde $i=[1..5]$</p> <p>$P[i] = A[i]*1.0 + B[i]*0.5 + C[i]*0.0$ (Puntaje inicial), donde $i=[1..5]$ (puntaje inicial)</p> <p>$F[i] = A[i]*1.0 + B[i]*0.5 + C[i]*0.0$ (Puntaje inicial), donde $i=[1..5]$ (puntaje final)</p> <p>$P = P1 + P2 + P3 + P4 + P5$ (Puntaje inicial)</p> <p>$F = F1 + F2 + F3 + F4 + F5$ (Puntaje final)</p> <p>$IF2 = P*10/N$ (Indicador Inicial) ó $F*10/N$ (Indicador Final)</p> <p>Ver cuadros de matriz de cálculo de indicadores.</p>
Criterio : Precisión	
Propósito : Capacidad del Software para proveer los resultados de manera precisa.	
Indicador	Medición o fórmula de cálculo
IF3	<p>Fórmula : $(10 - (CE*10/CCP))$</p> <p>CE = Cantidad de errores encontrados</p> <p>CCP = Cantidad de casos probados</p>
Criterio : Seguridad	
Propósito : Capacidad del Software para proteger la información y los datos, de manera que no puedan leerlos o modificarlos.	
Indicador	Medición o fórmula de cálculo
IF4	<p>Se considera a alta seguridad a la información a :</p> <p>Encryptamiento, VPN, SSL, Acceso mediante Login y Password.</p>

3.5.2.2 Indicadores de Eficiencia

Característica : Eficiencia	
Criterio : Comportamiento en el Tiempo	
Propósito : Capacidad del Software para proveer de tiempos apropiados de respuesta y procesamiento.	
Indicador	Medición o fórmula de cálculo
IE1	Fórmula = $CQO*10/(CQO + CQNO)$

	<p>CQO = Cantidad de Querys Optimos</p> <p>CQNO = Cantidad de Querys No Óptimos</p> <p>CQT = CQO + CQNO</p>
Criterio : Utilización de recurso	
Propósito : La capacidad del Software para utilizar cantidades de recursos apropiados.	
Indicador	Medición o fórmula de cálculo
IE2	<p>Prueba de Volumen y Comunicaciones</p> <p>10 : Soporta volumen de datos mayor al 25% de lo esperado</p> <p>6: Soporta volumen de datos igual a lo esperado</p> <p>1: No soporta volumen de datos esperado.</p> <p>0 : No se evaluó</p>
Criterio : Conformidad de concurrencia	
Propósito : El Software cumple con los niveles de concurrencia requeridos.	
Indicador	Medición o fórmula de cálculo
IE3	<p>Indicador de concurrencia :</p> <p>10 : Soporta numero de concurrencia mayor al 25% de lo esperado.</p> <p>6: Soporta numero de concurrencia igual a lo esperado.</p> <p>1: No soporta numero de concurrencias esperadas.</p>

3.5.2.3 Indicadores de Mantenimiento

Característica : Mantenimiento	
Criterio : Analizabilidad	
Propósito : La capacidad del Software para ser diagnosticado por deficiencias o causas de falla, así como la identificación de partes a ser modificadas.	
Indicador	Medición o fórmula de cálculo
IM1	<p>Puntaje inicial : Puntaje obtenido del primer documento de definición F2 obtenido en la fase de Modelamiento de Tecnología.</p> <p>Puntaje final : Puntaje obtenido del documento de definición F2 obtenido en la fase de Modelamiento de Tecnología del sistema final.</p> <p>Tipos de modelos evaluados :</p> <p>Diagramas de arquitectura equivalente al elemento i = 1</p> <p>Catálogo de subsistemas equivalente al elemento i = 2</p> <p>Diseño de casos de usos reales equivalente al elemento i = 3</p> <p>Diagramas de clases equivalente al elemento i = 4</p>

	<p>$N[i]$ = Cantidad de diagramas del tipo $[i]$, donde $i=[1..4]$</p> <p>$A[i]$ = Cantidad del modelo del tipo$[i]$ correctamente diseñados, donde $i=[1..4]$</p> <p>$B[i]$ = Cantidad del modelo del tipo$[i]$ ambiguamente diseñados, donde $i=[1..4]$</p> <p>$C[i]$ = Cantidad del modelo del tipo$[i]$ incorrectamente diseñados, donde $i=[1..4]$</p> <p>$P[i] = A[i]*1.0 + B[i]*0.5 + C[i]*0.0$, donde $i=[1..4]$ (puntaje inicial)</p> <p>$F[i] = A[i]*1.0 + B[i]*0.5 + C[i]*0.0$, donde $i=[1..4]$ (puntaje final)</p> <p>$P = P1 + P2 + P3 + P4 + P5$ (Puntaje inicial)</p> <p>$F = F1 + F2 + F3 + F4 + F5$ (Puntaje final)</p> <p>$IF2 = P*10/N$ (Indicador Inicial) ó $F*10/N$ (Indicador Final)</p> <p>Ver cuadros de matriz de cálculo de indicadores.</p>
Criterio : Cambiabilidad	
Propósito : La capacidad del Software para permitir que una determinada modificación sea implementada.	
Indicador	Medición o fórmula de cálculo
IM2	<p>Verificación de nivel de documentación y claridad de código fuente:</p> <p>10 : Bueno</p> <p>6 : Regular</p> <p>1 : Malo</p>
Criterio : Testeabilidad	
Propósito : La capacidad del Software para permitir que las modificaciones puedan ser validadas.	
Indicador	Medición o fórmula de cálculo
IM3	<p>Verificación de nivel de especificación de flujo de Eventos de CU o mini especificaciones :</p> <p>10 : Bueno</p> <p>6 : Regular</p> <p>1 : Malo</p>
Criterio : Conformidad de codificación	
Propósito : Capacidad de cumplimiento de estándares de codificación.	
Indicador	Medición o fórmula de cálculo
IM4	<p>10 : Cumple eficientemente estándares de codificación</p> <p>6 : Cumple medianamente estándares de codificación</p> <p>1 : No cumple</p> <p>0 : No se dispone de estándares de codificación</p>

3.5.2.4 Indicadores de Usabilidad

Característica : Usabilidad

Criterio : Entendibilidad

Propósito : Medida que indica la capacidad de los usuarios potenciales para entender la descripción del producto final.

Indicador	Medición o fórmula de cálculo
IU1	<p>Puntaje inicial : Puntaje obtenido de la primera primer versión del sistema.</p> <p>Puntaje final : Puntaje obtenido luego de la última revisión del sistema.</p> <p>Tipos de pantallas evaluadas :</p> <p>Pantallas de entrada equivalente al elemento $i = 1$</p> <p>Pantallas de consulta equivalente al elemento $i = 2$</p> <p>Pantallas de reportes equivalente al elemento $i = 3$</p> <p>$N[i]$ = Cantidad pantallas del tipo $[i]$, donde $i=[1..3]$</p> <p>$A[i]$ = Cantidad de pantallas del tipo$[i]$ con mensajes 100% intuitivos, donde $i=[1..3]$</p> <p>$B[i]$ = Cantidad de pantallas del tipo$[i]$ con al menos un mensaje ambiguo, donde $i=[1..3]$</p> <p>$C[i]$ = Cantidad de pantallas del tipo$[i]$ con al menos un mensaje errado, teniendo prioridad el errado sobre el ambiguo, donde $i=[1..3]$</p> <p>$P[i] = A[i]*1.0 + B[i]*0.5 + C[i]*0.0$, donde $i=[1..3]$ (puntaje inicial)</p> <p>$F[i] = A[i]*1.0 + B[i]*0.5 + C[i]*0.0$, donde $i=[1..3]$ (puntaje final)</p> <p>$P = P1 + P2 + P3 + P4 + P5$ (Puntaje inicial)</p> <p>$F = F1 + F2 + F3 + F4 + F5$ (Puntaje final)</p> <p>$IU1 = P*10/N$ (Indicador Inicial) ó $F*10/N$ (Indicador Final)</p> <p>Ver cuadros de matriz de cálculo de indicadores.</p>

Indicador	Medición o fórmula de cálculo
IU2	<p>$IU2 = 10*A / B$</p> <p>A = Número de funciones (casos de uso) o tipos de funciones descritas en la documentación de usuario, ayudas, etc.</p> <p>B = Número total de funciones (caso de uso) o tipos de funciones.</p>

Criterio : Atractividad

Propósito : Capacidad del Software de ser atractivo para el usuario.

Indicador	Medición o fórmula de cálculo
IU3	<p>Puntaje inicial : Puntaje obtenido de la primera primer versión del sistema.</p> <p>Puntaje final : Puntaje obtenido luego de la última revisión del sistema.</p> <p>Tipos de pantallas evaluadas :</p> <p>Pantallas de entrada equivalente al elemento $i = 1$</p> <p>Pantallas de consulta equivalente al elemento $i = 2$</p> <p>Pantallas de reportes equivalente al elemento $i = 3$</p> <p>$N[i]$ = Cantidad pantallas del tipo $[i]$, donde $i=[1..3]$</p> <p>$A[i]$ = Cantidad de pantallas del tipo$[i]$ muy atractivas, donde $i=[1..3]$</p>

	<p>$B[i]$ = Cantidad de pantallas del tipo[i] regularmente atractivas, donde $i=[1..3]$</p> <p>$C[i]$ = Cantidad de pantallas del tipo[i] pobremente atractivamente, donde $i=[1..3]$</p> <p>$P[i]$ = $A[i]*1.0 + B[i]*0.5 + C[i]*0.0$, donde $i=[1..3]$ (puntaje inicial)</p> <p>$F[i]$ = $A[i]*1.0 + B[i]*0.5 + C[i]*0.0$, donde $i=[1..3]$ (puntaje final)</p> <p>P = $P1 + P2 + P3 + P4 + P5$ (Puntaje inicial)</p> <p>F = $F1 + F2 + F3 + F4 + F5$ (Puntaje final)</p> <p>$IU3$ = $P*10/N$ (Indicador Inicial) ó $F*10/N$ (Indicador Final)</p> <p>Ver cuadros de matriz de cálculo de indicadores.</p>
Criterio : Cumplimiento de usabilidad	
Propósito : El Software cumple con los estándares de usabilidad	
Indicador	Medición o fórmula de cálculo
IU4	<p>10 : Cumple eficientemente estándares de usabilidad</p> <p>6 : Cumple medianamente estándares de usabilidad</p> <p>1 : No cumple</p> <p>0 : No se dispone de estándares de usabilidad</p> <p>Se puede medir en función al grado de satisfacción de usuario según las respuestas a un cuestionario :</p> <p>$IU4 = 10*A/B$</p> <p>A=Valor producida por el cuestionario</p> <p>B=Mayor valor de la escala que puede producir el cuestionario.</p>

3.5.2.5 Indicadores de Fiabilidad

Característica : Fiabilidad	
Criterio : Madurez	
Propósito : Capacidad del Software de evitar fallas como resultado de errores en el Software.	
Indicador	Medición o fórmula de cálculo
IC1	<p>10 : El software evito la prueba de fallo. (dato inconsistente)</p> <p>0 : No evito Fallo</p>
Criterio : Tolerancia a Fallos	
Propósito : Capacidad del Software para mantener un nivel de funcionamiento en caso de errores de software o en el caso de una interfaz incompleta.	
Indicador	Medición o fórmula de cálculo

IC2	10 : Se mantuvo un nivel de funcionamiento al fallar el software. 6 : Se mantuvo un regular nivel de Funcionamiento al fallar el software 1 : Dejo de funcionar
Criterio : Recuperabilidad	
Propósito : Capacidad del Software para reestablecer un nivel de funcionamiento y recuperar datos en caso de fallas.	
Indicador	Medición o fórmula de cálculo
IC3	10: Se recuperó los datos después de Fallo 6 : Recuperación Parcial 1 : No se recuperó

3.5.2.6 Indicador de Portabilidad

Característica : Portabilidad	
Criterio : Facilidad de instalación	
Propósito : Mide la facilidad de instalación del software en un ambiente operacional.	
Indicador	Medición o fórmula de cálculo
IP1	$IP1=10 \cdot A/B$ <p>A=Número de casos en que el usuario es exitoso en la operación de instalación. B=Número total de casos en que el usuario intenta ejecutar la operación de instalación.</p>

CUADROS DE MATRIZ DE CALCULO DE INDICADORES

Indicador de requerimientos IF1							
Modelo Revisado	Detalle	Cuenta	N° requerimientos		Puntaje inicial	Puntaje final	
			Consistentes	Ambiguos Inconsistentes			
Catálogo de requerimientos	Requerimientos Funcionales	N1	A1	B1	C1	P1	
Catálogo de requerimientos	Requerimientos no Funcionales	N2	A2	B2	C2	F2	
Total		N	A	B	C	P	
Indicador						P*10/N	F*10/N

Indicador de conformidad del modelo IF2							
Modelo Revisado	Diagrama	Cuenta	N° Técnicamente		Puntaje inicial	Puntaje final	
			Correctos	Ambiguos Incorrecto			
Diagrama de Casos de Uso	Diagrama de Casos de Uso	N1	A1	B1	C1	F1	
Subsistemas de análisis	Diagrama de Paquetes	N2	A2	B2	C2	F2	
Comportamiento dinámico de la Interface	Diagrama de Navegación	N3	A3	B3	C3	F3	
Análisis de Casos de Uso	Diagrama de Realización ó Diagrama de Colaboración	N4	A4	B4	C4	F4	
Análisis de diagramas de clases	Diagrama de clases	N5	A5	B5	C5	F5	
Total		N	A	B	C	P	
Indicador						P*10/N	F*10/N

Indicador de Modelamiento de Tecnología IM1									
Modelo Revisado	Técnica	Cuenta	N° técnicamente		Puntaje inicial	Puntaje final			
			Correctos	Incorrectos					
			1.00	0.00					
Arquitectura del sistema	Diagrama de arquitectura	N1	A1	B1	C1	P1	F1		
Arquitectura del sistema	Catálogo de subsistemas	N2	A2	B2	C2	P2	F2		
Diseño de realización de CU	Diseño de CU reales	N3	A3	B3	C3	P3	F3		
Diseño de clases	Diagrama de clases	N4	A4	B4	C4	P4	F4		
Total		N	A	B	C	P	F		
Indicador									
					P*10/N	F*10/N			

Indicador de Entendibilidad IU1									
Modelo Revisado	Técnica	Cuenta	N° Pantallas Intuitivas		Puntaje inicial	Puntaje final			
			Intuitivo	Mensajes errados					
			1.00	0.00					
Pantallas de entrada	Inspección de mensajes en SW final o Prototipos	N1	A1	B1	C1	P1	F1		
Pantallas de consulta	Inspección de mensajes en SW final o Prototipos	N2	A2	B2	C2	P2	F2		
Pantallas de Reportes	Inspección de mensajes en SW final o Prototipos	N3	A3	B3	C3	P3	F3		
Total		N	A	B	C	P	F		
Indicador									
					10*P/N	10*F/N			

Indicador de Atractividad IU3								
Modelo Revisado	Técnica	Cuenta	Nº Pantallas Atractivas			Puntaje inicial	Puntaje final	
			Bueno	Regular	Malo			
			1.00	0.50	0.00			
Pantallas de entrada	Inspección de Estética botones, campos, columnas, Colores,	N1	A1	B1	C1	P1	F1	
Pantallas de consulta	Inspección de Estética botones, campos, columnas, Colores,	N2	A2	B2	C2	P2	F2	
Pantallas de Reportes	Inspección de Estética botones, campos, columnas, Colores,	N3	A3	B3	C3	P3	F3	
Total		N	A	B	C	P	F	
Indicador							10*P/N	10*F/N

3.5.3 Utilización de Herramientas BTO

En la SUNAT se decidió adquirir una suite de herramientas BTO para mejorar el proceso de aseguramiento de la calidad. La DCC realizará un conjunto de pruebas exhaustivas al software con la finalidad de identificar defectos y posibles fuentes de errores que impidan que éste satisfaga todos los requerimientos del sistema, permitiendo que cumpla con las restricciones y condicionantes establecidas formalmente en el documento de definiciones del software.

La prueba de software no puede asegurar la ausencia de defectos, sólo puede demostrar que existen defectos en el software. La División de Control de Calidad debe ejercer su función de velar por la Calidad del Sistema en todas las etapas de la metodología de desarrollo de sistemas en la cual participa, y así garantizar una implementación segura.

Una de las estrategias adoptadas para minimizar el riesgo de adquirir una herramienta no adaptable a la realidad de la Intendencia Nacional de Sistemas de Información, fue convocar a los 3 líderes mundiales de las herramientas de productividad de Testing, según el cuadrante mágico de Gartner estas fueron : Mercury, Compuware Corporation e IBM, resultando ganador de la licitación la segunda de ellas en Diciembre del 2005.

La herramienta tenía que cumplir una relación de requisitos mínimos las cuales se especifican en el anexo 10.

3.5.3.1 Plan de implantación de herramientas BTO

En enero del año 2006 se conformó un equipo de trabajo con personal de la DCC y la OIPS para trabajar en el plan de capacitación, al decidirse la compra del software de Compuware para el apoyo automatizado de las pruebas.

La capacitación técnica al personal de SUNAT sería impartida para 30 personas por la empresa Software S.A., representantes en el Perú de Compuware y constaría de 120 horas con los siguientes tópicos :

- Arquitectura detallada del software
- Administración y programación de la herramienta
- Talleres prácticos con ejemplos de aplicativos programados en Java con acceso a base de datos Informix 7.31, aplicativo programado en Power Builder con acceso a base de datos Informix 9.31, Aplicativo programado en Visual Basic con acceso a base datos Access, aplicativo en lenguaje 4gl de Informix que acceda a una base de datos Informix 9.31 y aplicativo programado en Visual Fox Pro que acceda a una base de datos Oracle 8.

Se pensó que la capacitación teórica y talleres prácticos de la nueva herramienta no sería suficiente, además de ello sería necesario el aprendizaje de las buenas prácticas al aplicar las pruebas en dos aplicativos reales de SUNAT con la asesoría de un equipo con personal de la empresa representante de Compuware en el Perú (Software S.A.), éste equipo se denominaría “equipo de consultores” y sus servicios durarían 15 días calendarios. A su vez en SUNAT se conformaría un equipo de trabajo conformado por personal técnico y funcional que se encargaría de las coordinaciones con el equipo de consultores. El equipo de consultores sería responsable del 100% de las actividades.

Las características del software a ser probado serían

- Aplicación Cliente/Servidor con acceso a una base de datos Informix y lenguaje Power Builder .

- Aplicación Web con acceso a una base de datos Informix, servidor de aplicaciones Web Logic y programado en java bajo estándares J2EE.

Las herramientas Compuware serían utilizadas por todo el personal de la DCC luego de finalizada la capacitación del Software; seguirían existiendo algunas excepciones como el Memorandum de la INSI N° 2005-23951 y 2005-24656 que autoriza a un sistema a no ser probado por la División de Control de Calidad dependiendo de la urgencia para ser atendido.

En el mes de Marzo del 2006 se conformó un equipo de 4 profesionales pertenecientes a la DCC con la finalidad de evaluar los requisitos mínimos especificados (anexo 10), el equipo de trabajo debía estar asignado a ésta tarea a dedicación exclusiva durante 1 mes calendario.

A continuación se muestra el cronograma de implantación de las diferentes actividades de capacitación y talleres

CAPÍTULO IV

EVALUACION DE RESULTADOS

4.1 Beneficios Tangibles

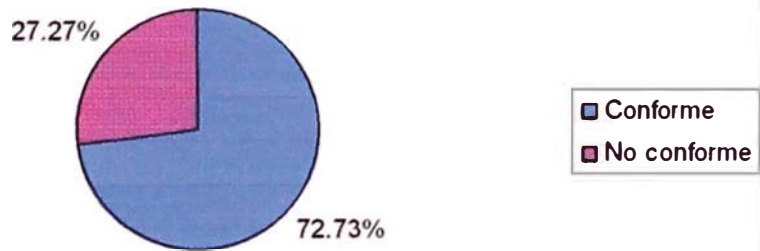
Como resultado de la aplicación de los 3 componentes propuestos, es decir, una metodología de desarrollo aplicada, la implementación de métricas de calidad y la utilización de herramientas para el aseguramiento de la calidad, se espera la reducción del porcentaje de recursos dedicados al mantenimiento del 60% al 33.82% a partir del segundo año debido al proceso de aseguramiento de la calidad de software.

4.2 Estadísticas

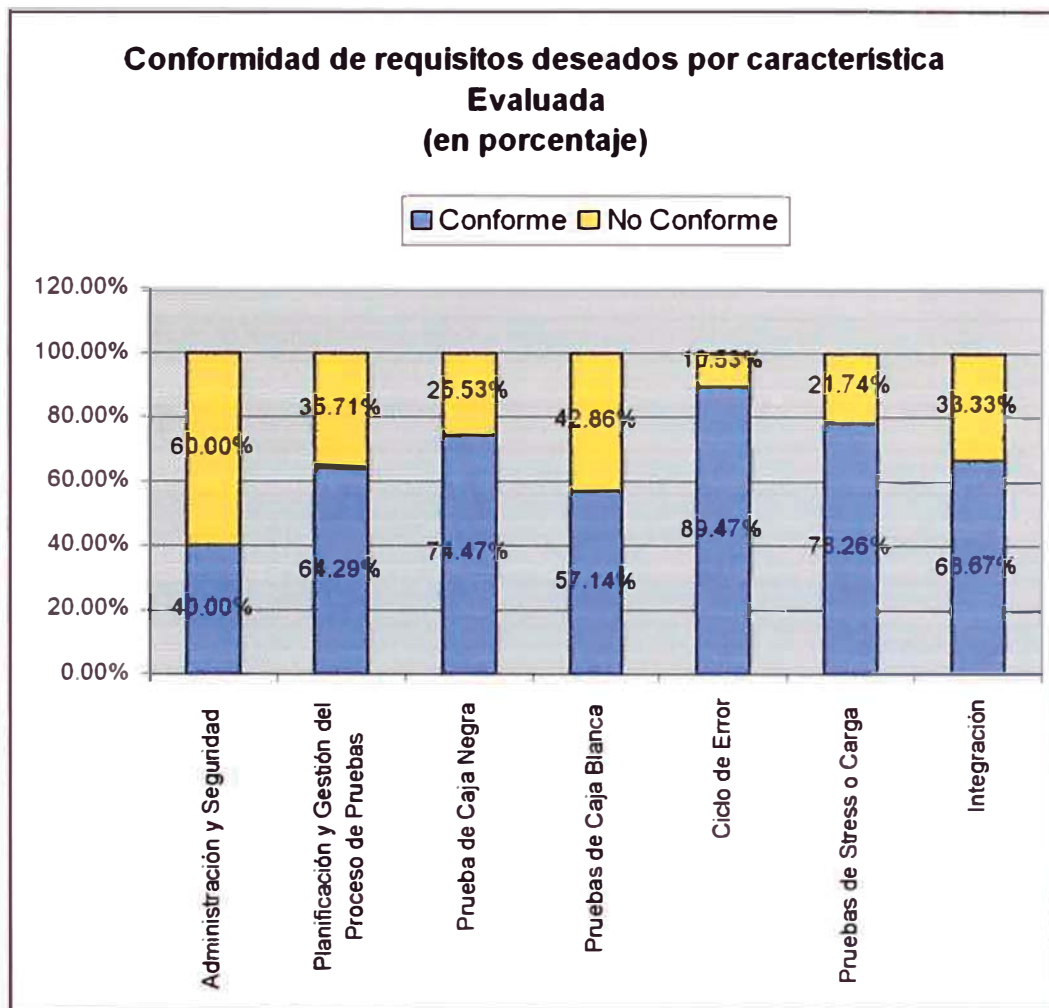
Los resultados de la verificación de requisitos respecto a las características de las herramientas Compuware y solicitadas en las bases, arrojan que el 27.27% de los requisitos exigidos no satisfacen lo requerido, por lo que la premisa inicial planteada que suponía una reducción del porcentaje de los costos de mantenimiento en 60%, sufriría una desviación, conllevando a la disminución de éste indicador al 43.64%. A continuación se muestra el cuadro de conformidad de requisitos por característica deseada :

Característica	Conforme	No conforme	Total
Administración y Seguridad	2	3	5
Planificación y Gestión del Proceso de Pruebas	18	10	28
Prueba de Caja Negra	35	12	47
Pruebas de Caja Blanca	4	3	7
Ciclo de Error	17	2	19
Pruebas de Stress o Carga	18	5	23
Integración	2	1	3
Total	96	36	132
%	72.73%	27.27%	100.00%

**Conformidad de requisitos deseados
(en porcentaje)**



Característica	Conforme	No Conforme	Total
Administración y Seguridad	40.00%	60.00%	100.00%
Planificación y Gestión del Proceso de Pruebas	64.29%	35.71%	100.00%
Prueba de Caja Negra	74.47%	25.53%	100.00%
Pruebas de Caja Blanca	57.14%	42.86%	100.00%
Ciclo de Error	89.47%	10.53%	100.00%
Pruebas de Stress o Carga	78.26%	21.74%	100.00%
Integración	66.67%	33.33%	100.00%



Por otro lado, teniendo como información del registro de horas que el porcentaje de recursos dedicados al mantenimiento es del 60% y que la reducción del mantenimiento es en 43.64%, se espera que el porcentaje de mantenimiento disminuya de 60% al 33.82% tal como se observa en los siguientes cuadros :

Tareas	Horas	Porcentaje
Tareas dedicadas a Proyectos	91,570.40	40%
Tareas dedicadas a Mantenimientos	135,578.00	60%
Total	227,148.40	100.00%

Disminución esperada del mantenimiento en 60% (antes de evaluación del software)

Porcentaje de tareas dedicadas a Proyectos	76.0%
Porcentaje de tareas dedicadas al Mantenimiento	24.0%

Disminución esperada del mantenimiento en 43.6% (luego de evaluación del software)

Porcentaje de tareas dedicadas a Proyectos	66.18%
Porcentaje de tareas dedicadas al Mantenimiento	33.82%

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

- Disminución en costos. El costo y la dificultad de introducir cambios en un proyecto es directamente proporcional al grado de avance del mismo. Esto significa que cuanto más avanzado esté un proyecto en su desarrollo, más costoso y difícil será introducir cambios en él. Por esta razón al final de cada fase se verifica que se hayan cumplido los objetivos propuestos en forma satisfactoria y se evalúa si es factible continuar con la etapa siguiente.
- Necesidad de usar herramientas de productividad. La calidad tiene un costo y ésta se manifestará en el tiempo invertido en la verificación o validación de los puntos de control al final de cada fase de desarrollo de la metodología MDSI, un aumento en las acciones necesarias para asegurar la calidad del software significaría mas inversión en tiempo de parte de a División de Control de Calidad DCC, por lo que siempre será necesario el uso de herramientas automatizadas que complementen estas acciones con el objetivo de minimizar el tiempo de las pruebas exhaustivas.
- Cuidado con las fallas falsas. La pruebas automatizadas para caja negra pueden presentar fallas falsas producto de verificaciones sobre la interfase de usuario, debido a que las diferencias simples, inocuas, como el cambio de resolución, reubicación de objetos gráficos, entre otras

características gráficas podrían enviar mensajes aparentes de errores, debido a los cambios cosméticos en el software.

- No es posible automatizar todas las pruebas de software por las siguientes razones, primero, se ha comprobado que generalmente los scripts de testing para las pruebas funcionales siempre requieren de un mínimo de modificación manual en el código automático generado, lo que difícilmente se podrá realizar cuando los proyectos no disponen del tiempo suficiente; en segundo lugar, actualmente no todos los sistemas de la INSI se adaptan a las pruebas funcionales automáticas, esto se debe a que la información resultante de algunas consultas pueden variar en diferentes momentos por la complejidad que implica implementar un baseline de la información a probar, mayor aún cuando muchos usuarios de la DCC pueden estar modificando la misma información en otras pruebas.
- El proceso de aseguramiento de la calidad de software debe contar con el apoyo de la alta gerencia de la INSI por ser un factor crítico de éxito en el proyecto, su apoyo facilitará el cumplimiento del uso de la metodología MDSI.
- La calidad debe formar parte de la cultura organizacional de la INSI, el éxito del proyecto depende de que todos sus integrantes y a todo nivel se vean identificados con ésta filosofía, lo que al final debe verse reflejada en la satisfacción del usuario final.
- El aseguramiento de la calidad de software es una estrategia que contribuye a solucionar parte de los problemas de la INSI, pero estaríamos equivocados si pensamos que es la única forma. Existen otras estrategias que se aúnan a la misma causa, como son los proyectos para la gestión del conocimiento, gestión de la configuración, procesos de data cleaning, entre otros, que son una urgente necesidad actualmente en la Intendencia de Sistemas de Información.
- Nuevas funciones de la División de Control de Calidad. La función de verificar el cumplimiento de la metodología MDSI, los estándares de

programación y los procedimientos que involucra el aseguramiento de la calidad con excepción el de la administración del modelo de datos, deben ser asumidos por la DCC, funciones que hoy en día las ejecuta parcialmente la Oficina de Ingeniería y Procesos de Software OIPS.

RECOMENDACIONES

- Las soluciones BTO de apoyo al aseguramiento de la calidad del software se componen de varias herramientas cada una de las cuales tienen una mayor o menor fortaleza en alguna de las fases del ciclo de vida de las aplicaciones o según el tipo de prueba al que se especialice, por lo tanto se recomienda evaluar y/o adquirir los productos tomando lo mejor del mercado y no necesariamente una suite de productos del mismo proveedor, toda vez que éstas herramientas hoy en día podrían no estar totalmente integradas.

REFLEXIONES

“La Calidad no es un acto, es un hábito”. (Aristóteles)

“Para hacer Calidad , tiene que haber liderazgo”.

“La Calidad conlleva a la excelencia, el camino hacia la excelencia es la Calidad”.

GLOSARIO DE TÉRMINOS

BTO .- Bussines Technology Optimize

DCC .- División de Control de Calidad

INSI .- Intendencia Nacional de Sistemas de Información

INI .- Intendencia Nacional de Informática

OIPS .- Oficina de Ingeniería y Procesos de Software

Tareas Continuas .- Equivalente a las actividades de desarrollo por mantenimiento de los sistemas

MDSI .- Metodología de Desarrollo de Sistemas de Información.

Pase a producción .- Documento formal que contiene el procedimiento de instalación del software en el entorno de producción.

Informe de definición F2 .- Documento escrito en Microsoft Word donde se detalla toda la información requerida por la metodología MDSI.

Baseline .- Línea de base que almacena las condiciones de las pruebas iniciales con el fin de restablecerlas en otro momento.

Sripts de testing .- Código de programa generado automáticamente para las pruebas funcionales mediante las herramientas BTO.

BIBLIOGRAFÍA

- Ingeniería de Software - Roger S. Pressman
- NTP ISO-IEC 12207-2004 Ciclo de Vida del Software
- IEEE Standard for Software Test Documentation.
- IEEE STD 829
- ISO-IEC 9126
- Metodología de Desarrollo de Sistemas de Información – MDSI

ANEXOS

ANEXO 1

TRIBUTOS INTERNOS				
TEMA	COBRANZA	FISCALIZACION	ADMINISTRATIVOS	
SERVICIOS				
REGISTRO	FRACCIONAMIENTO REFT	DEVOLUCIONES	TRAMITE DOCUMENTARIO - SIGED	
CAPTURA DE DOCUMENTOS	VALORES	RSIRAT- FISCALIZACION	SERVICIOS INFORMATICOS - SIGESA	
ESSALUD	FRACCIONAMIENTO F36	MASIVOS	FINANCIERA -SIPF	
RED BANCARIA	FRACCIONAMIENTO RESIT	RSIRAT - OTROS	RECURSOS HUMANOS	
SOL-SUNAT EN LINEA	TRANSFERENCIA DE INFORMACION	PDT-INFORMATIVOS	GESTION OPERATIVA - SCAD	
PROCESOS	COBRANZA COACTIVA	GEMA	PORTAL INTRANET	
TRAMITES MULTIPLES	IMPUTACION DE PAGOS	FISCALIZACION - SIF	LOGISTICA - SISA	
COMPROBANTES DE PAGO	RECURSOS IMPUGNATORIOS	FISCALIZACION - SIFA (SICTA)	PORTAL INTERNET	
PDT DETERMINATIVOS	RELIQUIDACION	COA	TRAMITE DOCUMENTARIO - SIGAD TD	
CAJA	FRACCIONAMIENTO SEAP	FISCALIZACION - SICTA	FINANCIERA SIPF-CAFAE	
RECAUDA	INFRACCIONES	DETRACCIONES	FINANCIERA-SCCI	
SQS	FRACCIONAMIENTO F848	TRAMITES MULTIPLES	PORTAL DE GESTION	
NOVEDADES	REC	REGISTRO	TRAMITE DOCUMENTARIO - SID	
CONFIGURACION	RESOLUCIONES DE INTENDENCIA	SOL-SUNAT EN LINEA	GESTION OPERATIVA - SABEC	
DEVOLUCIONES	CONFIGURACION	QTOOLS	GESTION - CUBO	
LEY DE TRANSPARENCIA	NOTIFICACION SCAD		GESTION - SIGE	
DCTOS VALORADOS	REIMPUTACION		COMUNICADOS WEBMASTER	
RELIQUIDACION	INGRESO AL SISTEMA		GESTION	
LISTADOS RECAUDACION	CENTRALES DE RIESGO		FINANCIERA SIPF-BID	
SIRAT	PRIVAT.y PROC.CONCURSAL		PROCESOS	
REGIMENES DEFINITIVOS	SIDCOT		PRESUPUESTO	
DETRACCIONES	SIRAT		FINANCIERO	
FRACCIONAMIENTO F36	SIAF		MOVIMIENTO DE FONDOS	
RESOLUCION DE INTENDENCIA	PROVEEDORES DEL ESTADO			
	DEVOLUCIONES			

ANEXO 2

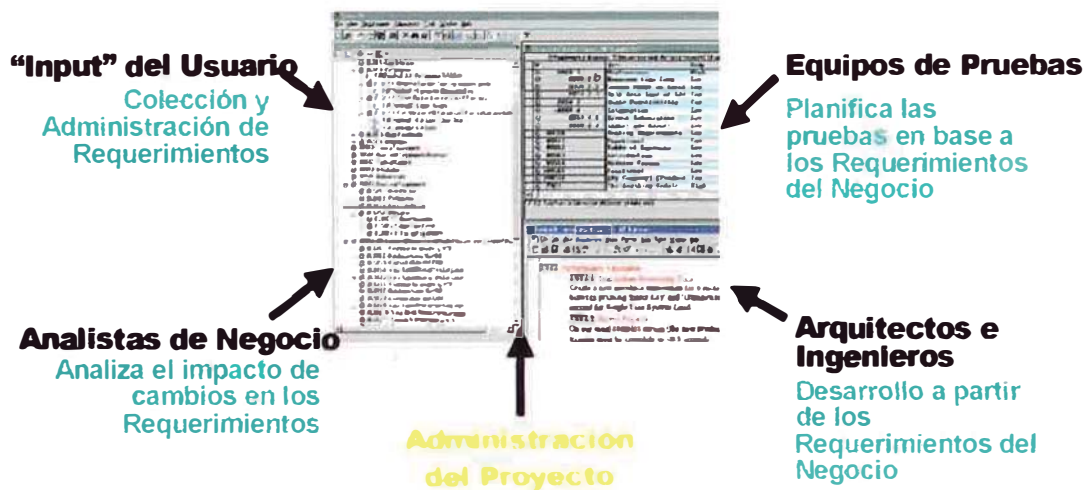
TEMA
TRIBUTOS ADUANEROS
ADUANAS
RECAUDACION
REGIMENES DEFINITIVOS
OTROS
MANIFIESTOS
ALMACENES
REGIMENES NO DEFINITIVOS
CETICOS
FISCALIZACION
PREVENCION Y REPRESION
TECNOLOGIA DE INFORMACION
ADMINISTRACION
RECURSOS HUMANOS
REEMBARQUE
TABLAS GENERALES
PAGINA WEB
SIGAD-TD
CAJA CHICA
ESTADISTICA
VALORACION
CONFIGURACION
COA-EXPORTADORES
CONSULTAS DE PERSONAL
INTRANET EN LINEA
CENTRALES DE RIESGO
RELIQUIDACION
SISTEMAS DE INFORMACION
DEVOLUCIONES
REGISTRO DE PERSONAL
SIGED

ANEXO 3

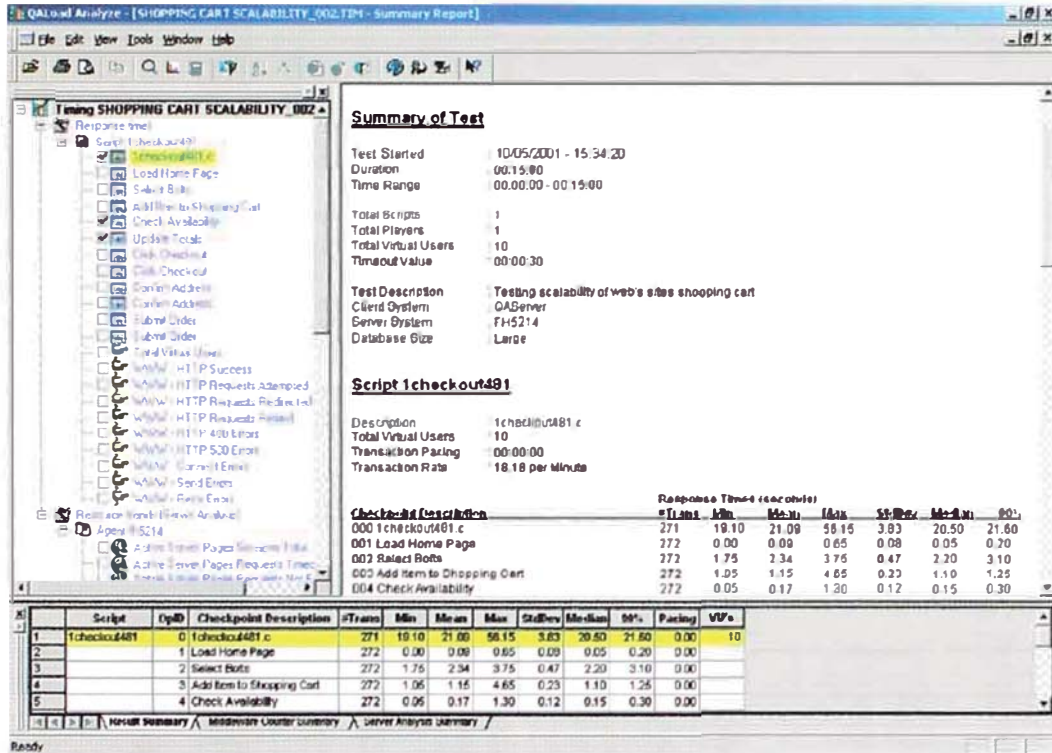
Herramientas Compuware

Pruebas de Caja Blanca	•DEVPartner S.P.
Pruebas de Código	• DEVPartner S.P. • DEVPartner J.E.
Pruebas de Caja Negra	• QACenter E.E.
Pruebas Funcionales	• QACenter E.E.
Pruebas de Carga y Stress	• QACenter P.E.
Pruebas Remotas	• Point Forward – Remote Agent
Pruebas de Aceptación	• QACenter E.E.

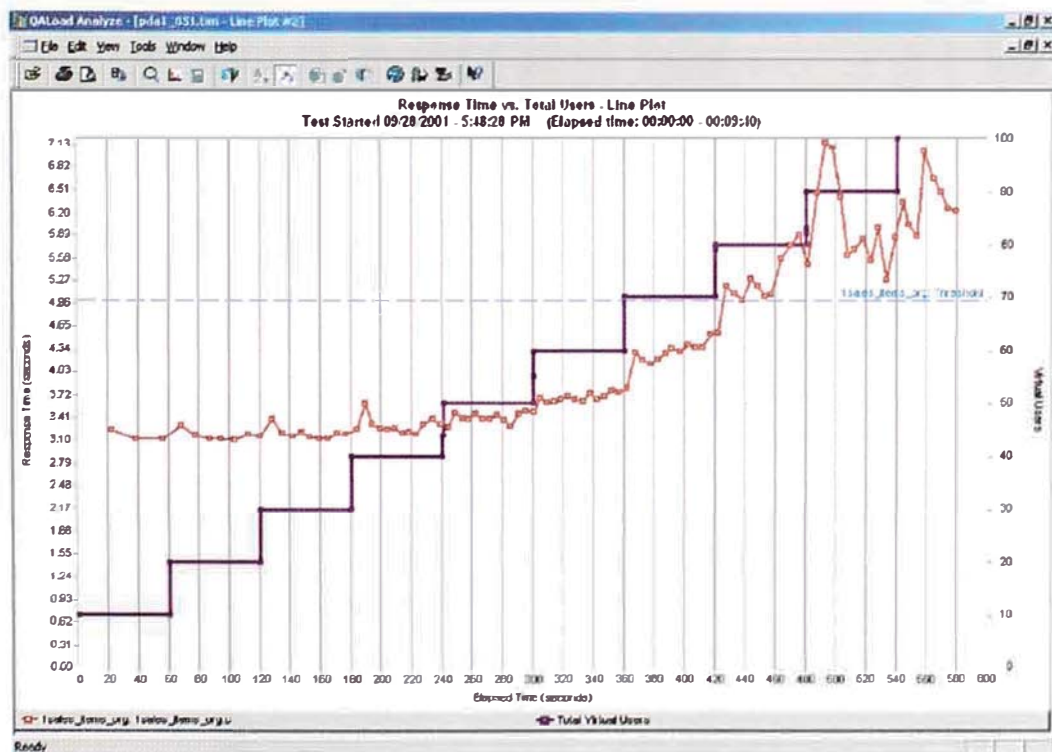
Administración de Requerimientos



Pruebas de Carga y Análisis



Análisis de Performance



ANEXO 4

CHECK LIST DE PRUEBAS GENERALES

N°	Sistema	Acción a realizar	Aplicó	Observación / Motivo
1	Todos	Pase a producción incluye documentación F:2.	Si	SIN DETALLE DE REGLAS
2	Todos	Pase a producción incluye Informe de Prueba	Si	SIN DETALLE DE CASOS PROBADOS
3	Todos	Se validó la existencia de directorios 4gx.pbl, dat, log en sunatId, etc. según se especifica en el instructivo del pase a producción.	Si	
4	Todos	Copiar mediante ftp los archivos fuentes especificados en el instructivo y verificar si alguno(s) no están en el directorio especificado.	Si	
5	Todos	Verificar que la lista de archivos con extensión .cat de los programas 4gx que se especifican en el instructivo del pase a producción esté completa. En el rediseño validar con el directorio de \$f: inclusive con los nuevos archivos cats remitidos a producción.	Si	
6	Todos	Validar el documento F:2 versus el cuadro de cumplimiento del MDSI.	Si	NO CUMPLE.
7	Rsirat,Sirat	Verificar mediante inspección de código los archivos 4gx.pbl y java remitidos en el pase a producción las diferencias existentes con las versiones en el directorio de fuentes de producción (\$F) y comprobar si los cambios obedecen únicamente a las especificaciones de la definición.	No	Se realizó diferencias de código de cdu25*4gx con antigua versión.
8	Todos	Verificar mediante inspección de código el archivo pNNNNNN.4gx y comprobar si los cambios obedecen únicamente a las especificaciones de la definición.	Si	
9	rsirat	Verificar mediante inspección de código el archivo pNNNNNN.4gx segmento de código con respecto a la siguiente información : - Grupo que envía el pase - Fecha de Salida del pase de desarrollo - Fecha y hora de salida del pase de calidad. - Si el pase incluye aplicativo Power en rediseño verificar que se incluya en el pNNNNNN.4gx el nombre de RSIRAT.E:XE.	Si	Aplica solo para pases de rediseño rsirat El pte. pase es del grupo SIRAT.
10	rsirat	Mediante inspección de código sobre el archivo pNNNNNN.4gx y de instalarse en el rediseño centralizado, comprobar que el programa se ejecute para el código de dependencia cod_dep = "0000".	Si	No se instaló en el rediseño centralizado.
11	Todos	Mediante inspección de código sobre el archivo pNNNNNN.4gx, comprobar que las altas o bajas de algunas opciones del menú correspondan únicamente lo que se especifica en el F:2.	No	No se envía nuevas opciones del menú
12	Rsirat,sirat	Verificar mediante inspección de código que los archivos SPL contengan en el archivo con extensión .sql una sentencia similar a: EXECUTE PROCEDURE sgs88 con spl ins		No aplica en este pase

		('2005', '000265', sgs25_obt_interes') al final de cada Procedure con el valor respectivo del año y número de pase.		
13	Todos	Verificar de la relación de programas fuentes enviados en el pase, que ninguno de ellos se encuentre en etapa de pruebas de calidad en otro pase a producción. De ser así las versiones deben ser sincronizadas.	Si	Se verificó en el listado de archivo excel.
14	Todos	Los programas modificados durante las pruebas serán renombrados con la extensión .N donde N es un número correlativo y deberán ser comparados los cambios realizados. Verificar si los cambios obedecen únicamente a las especificaciones de la definición.	Si	Esto sirve para verificar que los nuevos cambios realizados sean únicamente para las especificaciones comprendidas en el F2.
15	Todos	Preparar el plan de pruebas	Si	
16	Rsirat,sirat	En el rsirat, copiar los fuentes al \$P, hprico sunat0h:/sqa/WORK/PASE para compilarlos.	Si	
17	Rsirat,sirat	Luego de compilar los ejecutables copiarlos al directorio \$B sunat0h:/sqa/fte/rprico/bin	Si	
18	Rsirat,sirat	Después de finalizado las pruebas copiar los archivos fuentes 4gx, spl al directorio \$F sunat0h:/sqa/fte/rprico/4gx y \$SPL respectivamente.	Si	
19	Todos	Si la implantación del pase es de un nuevo sistema, el pase debería contemplar un proceso de control de errores que identifique y comunique automáticamente mediante correo electrónico a ATUS las ocurrencias de las inconsistencias sucedidas. El Tivoli es una alternativa.	No	No es nuevo sistema.
20	Todos	Para la ejecución de procesos Batch de información masiva y cuyo tiempo de ejecución estimado sean de días, verificar que el proceso administre las caídas del programa, de tal forma que pueda continuar la ejecución desde el último registro procesado + 1.	No	No hay proceso masivo..
21	Sirat,rsirat	Si en la relación de cat existen programas del módulo dp , se requerirá precisar en el instructivo del pase a producción : "Ejecutar el makefile de sidcot". No se debe consignar la lista de programas dpxxx.cat en el instructivo.	No	
22	rsirat	Verificar que el programa pNNNNNN.4gx actualice la fecha de salida del pase de la División de Control de Calidad.	Si	
23	Rsirat, Sirat	Verificar que de modificarse los siguientes programas : ins01.spl, sgs03.spl sgs17.spl, sgs25.spl, sgs33.spl , debe precisarse en el instructivo del pase a producción que deben remitirse a todas las dependencias Prico y Mepeco co Sirat instalado incluido SunatCC y ser ejecutado en las Bases de Datos rsirat, sirat y recauda inclusive. De la misma manera si se actualizan los siguientes parámetros : 035,286,050,129 de la tabla t0lparam.	No	

ANEXO 5
REPORTE DE PRUEBAS ESPECÍFICA

Regla de Negocio	Descripción
Incorporar nuevo menú la Boleta de Pago de Documentos Valorados (F. 1052, 1252) en el proceso de reimputación de Pagos	Modificación del proceso de Reimputación (cdu25.4gx) debido a que se está incorporando el nuevo motivo de Reimputación Por Documentos Valorados .Este nuevo motivo permitirá reimputar los formularios 1052 y 1252 con fecha de pago a partir del 01/07/1999, validando que sólo puede reimputarse a deudas destino cuyo tributo sea considerado ingreso del Tesoro Público.

Programa o Menú	Opción : Reimputación de Pagos – Registro de reimputación a pedido.	Respuesta real del Sistema
Caso de prueba	Respuesta esperada del Sistema	Respuesta real del Sistema
Visualizar nueva opción de Reimputación para documentos valorados con Boletas 1052,1252 en el menú en opción del menú sirat : Reimputación de pagos – Registro de reimputación a pedido	<p>Para ingresar a verificar nueva opción se ingresó :</p> <p>Formulario de Pago : 1662</p> <p>Nro. de Orden : <u>3238022</u></p> <p><u>HPRICO</u></p>	<p>OK ; Se muestra nueva opción del SIRAT:</p> <p>8. Por Pagos con Documentos Valorados (1052, 1252) .</p>
	Indicar en instructivo del pase la nueva opción en el menú del SIRAT a comunicarse a los usuarios.	KO! No hay presiones al respecto.
Se utilizó los siguientes datos : Formulario de Pago : 1662 Nro. de Orden : <u>3238022</u>	NO SE PRECISA MODIFICAR Ninguna Opción del menú de la versión anterior.	<p>Se modificó opción 4 :</p> <p>Antes : 4. Por Documento Valorado</p> <p>Versión en pruebas : 4. Por Documento Valorado - REFT,RESIT,SEAP</p> <p>KO! Debería precisarse en el Instructivo del Pase la modificación del texto de la opción 4.</p>

ANEXO 6

FORMATO DE PLAN DE PRUEBAS

Identificador	Descripción		
	Características a probarse del Software		
	•		
	Características a no ser probadas		
	•		
	Actividades	Tiempo estimado	
	•	•	
	Técnicas de Pruebas		
Unitarias	<input type="checkbox"/>	Funcionales <input type="checkbox"/> Caja Blanca <input type="checkbox"/>	Regresión <input type="checkbox"/> Valor límite <input type="checkbox"/>
Integración	<input type="checkbox"/>	Funcional <input type="checkbox"/> Caja blanca <input type="checkbox"/> Regresión <input type="checkbox"/>	Carga y Stress <input type="checkbox"/> Valor límite <input type="checkbox"/>
Sistema	<input type="checkbox"/>	Funcional <input type="checkbox"/> Caja blanca <input type="checkbox"/> Regresión <input type="checkbox"/>	Carga y Stress <input type="checkbox"/> Valor límite <input type="checkbox"/> Documentación <input type="checkbox"/>
Aceptación	<input type="checkbox"/>	Implantación <input type="checkbox"/>	
	Herramientas a utilizar		
	•		
	Criterio de Aprobación		
	•		
	Criterio de Suspensión		
	•		
	Entregables		
	• Plan de pruebas		

<ul style="list-style-type: none"> • Especificación de casos de pruebas • Log de pruebas • Reporte de incidencias 	
Necesidades de Entorno	
Hardware	<ul style="list-style-type: none"> • •
Software Base	<ul style="list-style-type: none"> • •
Comunicaciones	<ul style="list-style-type: none"> • •
Seguridad	<ul style="list-style-type: none"> • •
Otros	<ul style="list-style-type: none"> •
Participantes	
Staff necesario para entrenamiento	
Cronograma de pruebas	
<i>Adjuntar cronograma</i>	
Riesgos	Contingencias
<ul style="list-style-type: none"> • • 	<ul style="list-style-type: none"> • •
Aprobación	
_____	_____
Coordinador de Calidad	Jefe de División
Fecha : / /	

ANEXO 7
FORMATO DE CASOS DE PRUEBA

Identificador :		Versión :	
Responsable (s) :		Breve Descripción :	
Objetivo			
•			
Entorno de Configuración			
Hardware	•		
	•		
Software Base	•		
	•		
Comunicaciones	•		
	•		
Seguridad	•		
Otros	•		
Pre-condición		Post-condición	
•		•	
Acción :			
Datos de entrada	Resultados esperados	Resultados obtenidos	
Acción :			
Datos de entrada	Resultados esperados	Resultados obtenidos	
Revisión			
<hr style="width: 50%; margin: 0 auto;"/>		<hr style="width: 50%; margin: 0 auto;"/>	
Coordinador de Calidad		Jefe de División	
Fecha : / /			

ANEXO 8

FORMATO DE REPORTE DE INCIDENCIAS

Identificador		Pase a Producción	
Responsable (s) :			
Incidencias encontradas			
• Hardware <input type="checkbox"/>	• Seguridad <input type="checkbox"/>	• Software Base <input type="checkbox"/>	• Coordinaciones <input type="checkbox"/>
• Comunicaciones <input type="checkbox"/>	• Otros <input type="checkbox"/>		
Fecha	Descripción Incidencia	Impacto	
/ /			
/ /			
/ /			

ANEXO 9

FORMATO DE INFORME DE VERIFICACIÓN

Identificador		Pase a Producción
Responsable (s) :		
N°	Breve descripción	Comentarios
1	Verificación de consistencia de requerimientos	
1.1	Requerimientos consistentes	
1.1.1		
1.2	Requerimientos ambiguos	
1.2.1		
1.3	Requerimientos inconsistentes	
1.3.1		
2	Verificación de modelos de requerimientos	
2.1	Modelos correctos	
2.1.1		
2.2	Modelos ambiguos	
2.2.1		
2.3	Modelos incorrectos	
2.3.1		
3	Verificación de la precisión del SW	
3.1		
4	Verificación de la seguridad del SW	
4.2		
5	Verificación del comportamiento en el tiempo	

5.1		
6	Verificación de utilización de recursos	
6.1		
7	Verificación de concurrencias	
7.1		
8	Verificación de analizabilidad o del modelo de tecnología	
8.1	Modelos correctos	
8.1.1		
8.2	Modelos ambiguos	
8.2.1		
8.3	Modelos incorrectos	
8.3.1		
9	Verificación de la cambiabilidad	
9.1		
10	Verificación de la testeabilidad	
10.1		
11	Verificación de estándares de programación	
11.1		
12	Verificación de entendibilidad	
12.1		
13	Verificación de atractividad	
13.1		
14	Verificación del cumplimiento de usabilidad	
14.1		
15	Verificación de la madurez	
15.1		

16	Verificación de la tolerancia de fallos	
16.1		
17	Verificación de recuperabilidad de los datos	
17.1		
18	Verificación de la portabilidad	
18.1		

Evaluación

N°	Fecha	Evaluación	Comentario
1			
2			
Revisión			
_____		_____	
Coordinador de Calidad		Jefe de División	
Fecha : / /			

Nota : Considerar como puntaje de aprobación la calificación mayor a 5.0

Adjuntar Matriz de calificación del software.

ANEXO 10

ESPECIFICACIÓN DE REQUISITOS MINIMOS DE LA HERRAMIENTA A ADQUIRIR

Planificación y Gestión del Proceso de Pruebas Pruebas de Caja Negra	Integración del proceso de Planificación de Pruebas, Ejecución y Gestión de los defectos detectados, a través de un sistema tipo workflow configurable
	Permitir la participación de todos los involucrados en el proceso de pruebas: Analista de Negocio, Analistas de Pruebas, Jefes de Proyecto y Desarrolladores
	Soporte para la ejecución de pruebas manual de las aplicaciones.
	Organización de los casos de pruebas en carpeta o directorios.
	Reutilización de los Casos de Pruebas registrados en la planificación de un nuevo Plan de Pruebas
	Almacenamiento de scripts a través de un repositorio
	Definición de los datos de pruebas en formato de tipo excel.
	Generación de reportes de gestión, indicadores del estado de los diversos proyectos de testing. Permitiendo conocer como se viene realizando el proceso de pruebas de cada uno de ellos.
	Definición de diversos perfiles de acceso al proceso de pruebas.
	Soportar la ejecución de pruebas en modo local o remoto en paralelo.
	Infraestructura que soporta una disponibilidad del producto 24x7
	Almacenamiento de la información del proyecto de pruebas, como plan de pruebas, casos de pruebas, datos de pruebas y defectos en base de datos en un repositorio centralizado.
	Permitir el registro de enlaces o documentos externos utilizados como referencias en el proceso de planificación de las pruebas (plan de pruebas)
	Permitir establecer relaciones de secuenciación en la ejecución de los proyectos de pruebas, es decir, permitir establecer la dependencia entre proyectos de pruebas e tal forma que no se inicie la prueba de un proyecto si no se ha concluido las pruebas del proyecto que lo antecede.
	Permitir configurar el formato del plan de pruebas a fin de poder adicionar estados y/o atributos de control.
	Permitir registrar las mediciones en forma cuantificada para cada criterio de calidad definido en el plan de Pruebas
	Permitir hacer una evaluación objetiva de los resultados de las pruebas en base a los criterios de calidad definidos en el proyecto de pruebas.
	Para la automatización de las pruebas, debe soportar los siguientes tipos aplicaciones: Aplicaciones web, Aplicaciones cliente/servidor y aplicaciones basadas en terminales (emuladores).
	Generar en forma automática el informe de pruebas final
	La plataforma de software debe soportar los siguientes sistemas para la ejecución de los scripts de testing: Windows (XP, Professional, 2000, NT 4.0).
Soporte a la creación de testing automático con los métodos de grabación y codificación	
Posibilidad de ingresar puntos de verificación en el testing automatizado	
Utilización de un repositorio central para los objetos	

Utilización de tecnología para generar y ejecutar scripts de test que sean adaptables al cambio de los objetos
Habilidad de crear casos de pruebas para validar datos dinámicos sin la necesidad de codificación
Soporte a compartir los scripts de testing con el grupo de desarrollo a través del repositorio para reproducir los errores rápidamente
Soporte a la definición de permisos y accesos en el repositorio
Disponer de una interfase gráfica para el scheduling de casos de pruebas.
Ejecución de testing cross-browser, soportando internet explorer (5.0 o superior) y Netscape (6.2 o superior)
Habilidad de realizar pausa y continuar de un script de testing automatizado.
Soporte a la generación de reportes automáticos con los resultados de tests y métricas estadísticas del proyecto
Soportar ejecución de test local y remoto en paralelo.
Soporte a relacionar casos de prueba a requerimientos, de modo de generar los casos de pruebas en base a los requerimientos originales de la solución
Posibilidad de editar los scripts de testing para agregar funcionalidad
Habilidad de configurar un workflow para el planeamiento de testing (planear, diseñar, implementar, ejecutar, evaluar)
Soporte al ingreso de información para definir los casos de pruebas en formato de tipo: Excel, Rational Rose (modelos de casos de uso), y <u>rational requisitepro.</u>
Posibilidad de reutilizar casos de pruebas en el planeamiento de testing
Posibilidad de generar registros de entrada para la utilización en la reproducción de scripts
Soporte a la ejecución de testing manual de las aplicaciones
Funcionalidad de transparencia para ver los pasos de testing y la aplicación a realizar el testing en forma simultánea
Soporte de ejecución de test local y remoto en paralelo
Relacionar casos de prueba a requerimientos.
Permitir el análisis de impacto de los cambios en los requerimientos
Soporte a la integración automática para el reporte y seguimiento de defectos.
Integración de los scripts de pruebas con los datos de pruebas, permitiendo variar los valores de los datos de pruebas, para realizar diversas pruebas con el mismo script.
Generación de los scripts en lenguajes visual basic, .net, java, o en su defecto un lenguaje entendible y/o parecido a los lenguajes mencionados.
Posibilidad de edición de los scripts de testing para modificar su funcionalidad.
Administración de los datos de pruebas, permitiendo crear, modificar o extraer datos desde una base de datos o archivos planos.
Conexión a base de datos, como Oracle, Informix, mysql, sqlserver
Soporte de Testing para los siguientes ambientes: aplicaciones Web, aplicaciones cliente/servidor y aplicaciones basadas en Terminales.
Generación de reportes automáticos con los resultados de las pruebas del proyecto de pruebas
Generación de reportes automáticos con los resultados de las mediciones realizadas a las métricas de calidad definidas en el proyecto de pruebas.

	Posibilidad de editar los scripts de pruebas para adicionar funcionalidad o parametrizar variables de control, que permitan leer las fuentes de datos definidas para estos casos.
	Creación de objetos de pruebas
	Almacenar en un repositorio centralizado los diversos objetos creados
	Importación o utilización de objetos para la creación de casos de pruebas
	Permitir la vista de objetos recorridos de acuerdo a la secuencia de ejecución del script de pruebas
	Permitir la creación de puntos de control (checkpoints)
	Permitir contar con librerías de funciones y scripts tipo plantillas
	Generar reportes en formato word, html o xml.
	Soportar las pruebas funcionales para aplicaciones desarrolladas en HTML, DHTML, javascript, jsp
	Soportar las pruebas funcionales para aplicaciones desarrolladas en Visual Basic
	Soportar las pruebas funcionales para aplicaciones desarrolladas en Power Builder
	Soportar las pruebas funcionales para aplicaciones desarrolladas en Java
	Soportar las pruebas funcionales para aplicaciones desarrolladas en C++ / C
	Soportar las pruebas funcionales para aplicaciones desarrolladas en webservices
	Soportar las pruebas funcionales para aplicaciones desarrolladas en .Net
	Soportar las pruebas funcionales para aplicaciones desarrolladas en PHP
	Soportar las pruebas funcionales para aplicaciones desarrolladas en WML
	Soportar las pruebas funcionales para aplicaciones desarrolladas en Terminal Emulador
Ciclo del Error	Permite la administración y seguimiento de fallas.
	Manejo de formularios manejados por reglas y plantillas.
	Gestionar todo el ciclo del error con un cliente visual (No web)
	Notificación vía correo electrónico
	Soporte para configurar un workflow para la resolución del defecto encontrado.
	Permitir ingresar un defecto desde la interfaz de análisis de resultados de los casos de pruebas ejecutados.
	Disponer de información de auditoria detallada para el seguimiento de los defectos.
	Soporte a la realización de reportes y informes gráficos para la visualización de los pedidos de solución de errores.
	Permitir parametrizar una clasificación propia de los errores
	Permitir parametrizar los niveles de severidad de los errores
	Emisión de diversos reportes estadísticos de los errores
	Capacidad de exportar la información de los errores a formato excel
	Soporte a la integración automática para el reporte y seguimiento de defectos
	La plataforma de software debe soportar la administración de defectos encontradas anteriormente
	Soporte a la definición de un workflow para la resolución del defecto encontrado
	Se debe permitir ingresar un defecto desde la interfaz de análisis de resultados de los casos de pruebas ejecutados
	Disponer de información de auditoria detallada para el seguimiento de los defectos

	Habilidad de generar templates de información para la carga de errores
	Soporte a la utilización de permisos y roles dentro del workflow definido para la administración de defectos
	Soporte a la realización de reportes y informes gráficos para la visualización de los pedidos de solución de errores.
Pruebas de Stress o Carga	Administración de datos de pruebas
	Capacidad de parametrizar data de entrada a través de wizards
	Contar reglas para definir criterios para la creación de los usuarios virtuales, durante la ejecución de pruebas de stress.
	Analizar internamente el comportamiento de la aplicación en pruebas a través de sus capas: Cliente, Reglas de Negocio y Data
	Identificación de los cuellos de botellas en las aplicaciones
	Permitir identificar los puntos donde se deben realizar los ajustes de optimización de las aplicaciones
	Permitir detectar los problemas de concurrencia y de sincronización
	Se debe contar con consultas del histórico de tiempos y consumo de recursos por cada versión del aplicativo probado
	Debe soportar la plataforma tecnológica de SUNAT: Servidor de Aplicaciones weblogic, Base de Datos Informix, Base de Datos Oracle, Servidor de Aplicaciones Jboss, Servidor Apache, Servidor web Netscape, Apache.
	Debe soportar el Servidor de Aplicaciones WebSphere
	Soporte para pruebas de carga sobre windows
	Soporte para pruebas de carga sobre unix
	Soporte para pruebas de carga sobre linux
	Proporcionar un monitoreo sobre todas las capas del sistema: Sistema Operativo, Aplicación, Servidores, Componentes.
	Mostrar en tiempo real métricas sobre el monitoreo para el análisis del comportamiento del sistema mientras se realizan las pruebas de carga
	Permitir la visualización de las métricas, en tiempo real, de las métricas desde una consola centralizada para el análisis del comportamiento del sistema mientras se realizan las pruebas de carga
	Contar con un motor de análisis offline para la generación de reportes
	Permitir almacenar los datos de pruebas en archivos planos
	Permitir realizar análisis del tipo drill down (hacia adentro) en la revisión de los resultados de las pruebas
	Permitir analizar concurrentemente múltiples métricas de resultados
Características No Técnicas y de Valor	Capacidad de integración con al menos una herramienta reconocida de gestión de requerimientos del sistema que permita establecer una trazabilidad desde los casos de pruebas hacia los requerimientos (sincronizar cambios en los requerimientos con cambios en las pruebas)
	Capacidad de integración con al menos una herramienta reconocida de gestión del cambio que permita implantar un workflow de reporte y seguimiento de defectos.
	Capacidad de integración con al menos una herramienta reconocida de control de versiones que permita la integración del área de desarrollo con el área de QA.
	Soporte local tanto técnico como en educación a cargo de proveedores reconocidos
	El proveedor deberá contar con personal certificado otorgado por el fabricante de la herramienta.