

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS FACULTAD I



TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS

Título: Guía para asegurar la calidad en el sistema de gestión
de estaciones Nova - Xerberos

Clasificación: Investigación

Autor:

Yurisdán López Villa

Tutora: Ing. Yusleydi Fernández del Monte
Co – Tutor: Ing. Sergio Jesús García de la Puente

Ciudad de la Habana, Enero de 2012

Resumen

Los objetivos medulares en todo proceso de software son que el producto desarrollado esté acorde en su totalidad con las expectativas de los usuarios y producir el mismo con la mejor calidad posible. En la actualidad los problemas más severos a nivel mundial en cuanto a este tópico se basan en la poca experiencia, el escaso manejo de la documentación y la carencia de herramientas para realizar el trabajo, además de la insuficiente utilización de buenas prácticas que faciliten la labor del equipo. Todo lo antes expuesto trae como consecuencia resultados ineficientes e incumplimiento de los requerimientos establecidos, produciendo finalmente la insatisfacción del cliente. Debido a que no existen métodos o guías para el aseguramiento de la calidad del sistema para gestionar estaciones Nova – Xerberos es que se ha decidido llevar a cabo la presente investigación que se propone elaborar la guía inexistente antes mencionada. Para desarrollar la investigación se hizo uso de métodos científicos como: estadísticos (Delphi), teóricos (Analítico-Sintético, Histórico-Lógico) y empíricos (Observación); además se utilizó como técnica de recopilación de información la encuesta. Estos métodos y la posterior evaluación de la investigación por especialistas demostraron la necesidad de incorporar la propuesta como documentación al desarrollo del software antes mencionado y dicha guía puede constituir una nueva posibilidad para el aprendizaje de todos aquellos que de una forma u otra contribuyan a garantizar la calidad del software.

Palabras Claves: Aseguramiento de la Calidad de Software (ACS), Calidad, Guía y Sistemas de gestión de estaciones.

Tabla de contenido

<i>Introducción</i>	5
Capítulo 1: Fundamentación teórica de la gestión de la calidad enfocada al desarrollo del sistema de gestión de estaciones Nova - Xerberos	9
1.1 Concepto de Calidad	9
1.1.1 Calidad del Software	9
1.2 Nova - Xerberos y otros sistemas para gestionar estaciones	10
1.2.1 Sistemas de gestión y distribución de software	11
1.2.1.1 Landscape	11
1.2.1.2 Cfengine	11
1.2.1.3 Puppet	12
1.2.1.4 Spacewalk	12
1.2.1.5 Dameware Nt Utilities	13
1.2.1.6 Secospace TSM	13
1.2.1.7 Plataforma de software distribuido (T-arenal)	14
1.2.1.8 COSMO	15
1.3 Garantía de calidad del software	16
1.4 Aseguramiento de la Calidad del Software	17
1.4.1 Actividades para el aseguramiento de la calidad del software	18
Métricas del software	18
Verificación y Validación	19
Gestión de la configuración del software	19
Plan de aseguramiento de la calidad	20
1.5 Control de Calidad de Software	20
1.6 Factores de calidad del software	21
1.6.1 Modelo McCall	21
1.6.2 Modelo de Boehm	24
1.7 Modelos de Calidad de Software	25
1.7.1 Modelo SPICE (ISO/IEC 15504)	25
1.7.2 Modelo CMMI	26
1.8 Normas o estándares utilizados para gestionar la calidad de software	29
1.8.1 ISO 9001	30
1.8.2 ISO 9003	31
1.8.3 ISO 1028	31
1.9 Proceso de desarrollo de software	33
1.9.1 Proceso de desarrollo del sistema para gestionar estaciones Nova - Xerberos	35
1.10 Metodologías de desarrollo de software	36
1.10.1 SXP	36
1.10.2 Proceso Unificado de Rational (RUP)	38
1.11 Proceso de Mejoras	40
1.12 Resumen del Capítulo	41
Capítulo #2. Guía para asegurar la calidad en el sistema de gestión de	

estaciones Nova - Xerberos	42
2.1 Introducción de la Guía	42
2.1.1 Presentación	42
2.1.2 Estructura	42
2.1.3 Propósito	43
2.1.4 Alcance	43
2.1.5 Características principales	43
2.2 Descripción de la guía	43
2.2.1 Proceso de desarrollo definido en la guía	43
2.2.2 Componentes de la guía	46
2.2.2.1 Procesos de ACS aplicados al Personal	47
2.2.2.2 Procesos de ACS aplicadas a los elementos de configuración del proyecto	48
2.2.2.3 Procesos de ACS aplicadas al proceso de desarrollo	49
2.2.2.4 Procesos de ACS aplicadas al producto	54
2.2.2.5 Procesos de ACS para factores de calidad críticos en Nova - Xerberos.	57
2.2.2.6 Procesos de ACS para la arquitectura del Nova - Xerberos.	62
2.3 Plan de Implementación	65
Capítulo # 3. Evaluación de la guía para asegurar la calidad en el sistema de gestión de estaciones Nova - Xerberos	66
3.1 Evaluación por el método Delphi	66
3.2 Fase inicial	67
3.2.1 Formulación del problema	67
3.2.2 Selección del grupo de expertos	67
3.3 Fase exploratoria	69
3.3.1 Elaboración del cuestionario	69
3.4 Fase final	70
3.4.1 Cálculo de concordancia entre los expertos	70
3.4.3 Resultado final	74
<i>Conclusiones</i>	75
<i>Recomendaciones</i>	75
<i>Referencias Bibliográficas</i>	76
<i>Bibliografía</i>	78
<i>Anexos</i>	79

Introducción

Durante todo el decurso de la humanidad se han identificado disímiles necesidades. A todas ellas el hombre se ha propuesto darle respuestas con el mejor grado de solidez posible para así evitar volver a incurrir en lo que ya supuso estuvo solucionado, esto es a lo que se llama un trabajo “hecho con calidad”. Mundialmente en los medios de comunicación, particularmente en la informática, la calidad tiene una presencia vital debido al papel fundamental que juega en la competitividad de las empresas, pues los productos y servicios brindados deben adaptarse a las necesidades, gustos y exigencias de los potenciales clientes para mantenerse en el mercado. Cada aplicación hecha debe llevar intrínseca propiedades o cualidades como la seguridad, la disponibilidad, el rendimiento óptimo, la total integridad de los datos, la facilidad de mantenimiento y de ser usada, características principales mediante las cuales se puede asegurar la calidad del software.

Cuba de igual forma se ha propuesto que el software que en el país se desarrolle goce de altos niveles de calidad y principalmente la Universidad de las Ciencias Informáticas (UCI) donde existe CALISOFT, una dirección dedicada exclusivamente a la investigación y aplicación de métricas y estándares que ayuden en la gestión interna en temas de calidad de software (para visitar su portal mediante la web acceda a: <http://calisoft.uci.cu>). Desafortunadamente, la mayoría de los proyectos que se llevan a cabo en este centro de altos estudios no poseen un alto nivel en cuanto a esta temática se refiere, es el caso del proyecto productivo Nova – Xerberos. El sistema Nova - Xerberos tiene como finalidad principal la gestión de todos los eventos y configuraciones llevadas a cabo en una Computadora Personal (PC) o portátil, pudiendo hacer todo esto de forma centralizada mediante el establecimiento de la comunicación cliente-servidor. Además Nova - Xerberos permite resguardar información sensible, gestionar y configurar tanto discos duros como software y hardware así como perfiles de usuarios y también como funcionalidad de gran importancia tiene la protección de los sistemas de archivos contra acceso ilegal o daños maliciosos.

El propósito de las actividades de garantía de la calidad es tener un proceso de desarrollo sin errores y obtener gracias a ello la total aceptación de los clientes pero esta meta puede que no sea satisfactoriamente alcanzada pues el equipo de trabajo no cuenta con una guía para realizar todo este proceso de desarrollo de la forma óptima, evidenciándose como **situación problemática** de la investigación la inexistencia de una guía para administrar la calidad de software enfocada al desarrollo del sistema para gestionar estaciones Nova - Xerberos, sumando a esto que en el proyecto tampoco se tienen en cuenta, en su totalidad, los factores que determinan la calidad de software, no se utilizan métricas para poder cuantificar diferentes

parámetros durante todo el ciclo de vida del software y existe una ineficiente gestión de los errores cometidos, la convergencia de todos estos desaciertos puede conllevar a que no se puedan diseñar a tiempo las acciones correctivas necesarias para eliminar los problemas detectados.

De lo anteriormente planteado se llega al siguiente **problema científico**: ¿Cómo administrar la calidad en el desarrollo del sistema de gestión de estaciones Nova - Xerberos?

El **objeto de estudio** que encierra el problema a tratar es: Procesos de gestión de la calidad de software.

El **Campo de acción** que se define es: el proceso de gestión de la calidad de software en el desarrollo de la aplicación para gestionar estaciones Nova - Xerberos.

Para dar solución al problema identificado se determina como **objetivo general** de la investigación: Elaborar una guía que permita administrar la calidad de software en el desarrollo del sistema para gestionar estaciones Nova - Xerberos. Para cumplimentarlo se desglosa en los siguientes objetivos específicos:

Objetivos específicos

- Sistematizar sobre las tendencias actuales de la gestión de la calidad de software.
- Elaborar una guía que sirva de modelo en la administración de la calidad en el desarrollo del sistema para gestionar estaciones Nova - Xerberos.
- Evaluar la guía propuesta para asegurar la calidad del software en el desarrollo del sistema para gestionar estaciones Nova - Xerberos.

Tareas de la investigación

- Identificación de los principales conceptos relacionados con las actividades de garantía de la calidad de software para adquirir habilidades en los aspectos elementales de ésta área de conocimiento.
- Sistematización de las principales tendencias que existen en la gestión de la calidad de software para determinar las principales actividades que se realizan en función de garantizar la calidad de software.
- Análisis de normas, modelos y estándares de calidad que se utilizan en el desarrollo de software para determinar cómo se llevan a cabo los procesos de gestión de la calidad en sistemas que brinden la posibilidad de gestionar de forma centralizada toda la información y dispositivos en una computadora personal (PC).
- Elaboración de una guía que incluya los procesos de gestión de la calidad de software que se deben ejecutar en el desarrollo del sistema para gestionar estaciones Nova - Xerberos.

- Evaluación de la guía propuesta en el proyecto Nova - Xerberos para determinar las posibles ventajas que implicarían su uso.

Teniendo como **idea a defender**: La utilización de una guía que permita el aseguramiento de la calidad pudiera garantizar que el software producido en el proyecto Nova - Xerberos posea todos los parámetros de calidad de software necesarios.

Métodos Científicos

Para el desarrollo de ésta investigación se utilizan varios métodos científicos que la guían y facilitan, ellos son:

- Estadísticos

- **Delphi**: Es un proceso repetitivo. Su funcionamiento se basa en la elaboración de un cuestionario que ha de ser contestado por los expertos. Una vez recibida la información, se vuelve a realizar otro cuestionario basado en el anterior para ser contestado nuevamente [1]. Finalmente el responsable de ésta investigación pudo elaborar sus conclusiones a partir de la explotación estadística de los datos obtenidos, lo que contribuyó a evaluar la guía propuesta teniendo en cuenta la opinión de especialistas en el tema.

- Teóricos

- **Analítico-Sintético**: Permite estudiar de modo general el tema planteado para luego descomponerlo e interiorizar cada aspecto estudiado sintetizando lo esencial en función de lo investigado. Se utiliza en la investigación para instruirse acerca de cuáles son los procesos de gestión de la calidad que se persiguen en los distintos desarrollos de software, enfatizando en la manera en que esto se lleva a cabo en las aplicaciones que permiten gestionar estaciones de forma centralizada.
- **Histórico-Lógico**: El objetivo principal de este método es estudiar y analizar el marco histórico y estado del arte de las características del tema de la investigación. Se utiliza para estar al tanto del comportamiento, desarrollo, y estado actual de los aspectos de la calidad de software a nivel mundial y en la UCI, aprendiendo de las nuevas experiencias y teorías que han surgido a lo largo del tiempo.

- Empíricos

- **Observación**: El método de la observación es considerado el instrumento universal de la investigación y consiste en la percepción directa del objeto de la realidad. Es un reconocimiento de la realidad objetiva a través de la obtención de datos cuya fuente no se limita a los datos fácticos, sino que estarán también los datos tendenciales y los datos teóricos que son interpretados desde la cultura y desde las concepciones del autor de esta investigación. Este método es uno de los que mejores resultados brinda pues se constata en el escenario real de los eventos todo lo referente a la utilización y manejo del software de

gestión de estaciones en cuestión así como el conocimiento práctico que tienen los miembros del equipo de desarrollo.

Técnica de recopilación de información

- **Encuesta:** Con la utilización de éste método se adquiere información sobre problemas referidos al proyecto productivo en cuestión, entrevistando a cada miembro para conocer tanto aspectos teóricos como prácticos llevados a cabo por cada uno de ellos en particular. Se hace uso de un cuestionario previamente elaborado, con la idea de conocer la opinión y valoración de la persona seleccionada. Esta técnica es de vital importancia porque entre otras cosas va a permitir llevar a cabo la evaluación de la solución propuesta en la investigación.

Población y Muestra

La población está conformada por los sistemas de gestión de estaciones, y la muestra por el sistema de gestión de estaciones Nova Xerberos.

Posibles resultados:

- Obtener una guía que brinde un estándar para administrar la calidad del software en aplicaciones de gestión de estaciones.

El trabajo de diploma consta de tres capítulos organizados de la siguiente manera:

Capítulo # 1. Fundamentación teórica de la gestión de la calidad enfocada al desarrollo del sistema de gestión de estaciones Nova - Xerberos.

En este capítulo se dan a conocer los conceptos fundamentales para el entendimiento de la investigación, estudiando los principales aspectos que determinan la gestión de la calidad de software para sintetizarlos y obtener las buenas prácticas que se pueden incorporar al desarrollo del sistema para gestionar estaciones Nova - Xerberos.

Capítulo # 2. Guía para asegurar la calidad en el sistema de gestión de estaciones Nova – Xerberos.

En este capítulo se propone la guía que puede ayudar a obtener un elevado aseguramiento de la calidad en el sistema para gestionar estaciones Nova – Xerberos. Mediante la planificación del proceso de desarrollo, la realización de actividades y los procesos a seguir e indicando en cada caso las acciones correctivas, la idea es obtener una guía que pueda contribuir a que el proyecto logre la tan necesaria mejora continua.

Capítulo # 3. Evaluación de la guía para asegurar la calidad en el sistema de gestión de estaciones Nova – Xerberos.

En este capítulo se evalúa la propuesta de solución teniendo en cuenta el criterio de especialistas en el tema, ocurriendo todo esto en el escenario real en que se trabaja.

Capítulo 1: Fundamentación teórica de la gestión de la calidad enfocada al desarrollo del sistema de gestión de estaciones Nova - Xerberos

Durante el proceso de desarrollo de todo producto informático es necesario que se realicen actividades y acciones que ayuden a garantizar la calidad del software y con ello la satisfacción plena de los interesados.

Es por todo esto que en el presente capítulo se abordarán temas relacionados con:

¿Qué se define como calidad?, ¿Qué es entonces calidad de software?, ¿Qué es el software gestor de estaciones Nova - Xerberos?, ¿Qué se entiende por aseguramiento de la calidad?, ¿Cómo afrontar los procesos de calidad de software? ¿Qué es una norma?, ¿Qué es una metodología?, ¿Qué es un estándar?

1.1 Concepto de Calidad

Son variados los conceptos que se refieren a lo que es calidad de forma general, importantes figuras y organizaciones se dedican a profundizar en éste tópico, definiéndolo de las siguientes maneras:

La ISO (International Standard Organization), define La Calidad como la ausencia de deficiencias:

“Es la totalidad de aspectos y características de un producto o servicio que se refieren a su capacidad para satisfacer necesidades dadas en la adecuación de sus objetivo”. [2]

1.1.1 Calidad del Software

El Instituto de Ingeniería de Software (SEI) en su modelo CMM define la calidad como: [3]

- El grado en el cual un sistema, componente o proceso cumple con los requisitos especificados.
- El grado en el cual el sistema, componente o proceso cumple con las expectativas del cliente o usuario.

La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad y que por consiguiente se puede concluir que para decir que algo posee la calidad necesaria debe estar exento de errores, tanto funcionales como de composición y que se adecua a características, propiedades y atributos determinados de un producto o servicio.

La Calidad del Software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia.

Según Roger Pressman, una de las principales figuras a nivel internacional en temas de Ingeniería de Software, calidad del software es la *“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". [4]

La ISO/IEC (International Standard Organization) la define como: "la totalidad de rasgos y atributos de un producto de software que le apoyan en su capacidad de satisfacer sus necesidades explícitas e implícitas". [5]

Por lo que se puede afirmar que un software tiene todos los parámetros de calidad cuando logra satisfacer las necesidades y expectativas de los clientes pues cumple con todos los requisitos y características indicadas por ellos.

1.2 Nova – Xerberos y otros sistemas para gestionar estaciones

Nova – Xerberos es un software gestor de estaciones desarrollado en la Universidad de las Ciencias Informáticas que permite configurar de forma centralizada y remota cada una de las terminales que se encuentren disponibles para ello. Mediante este programa informático se puede brindar el servicio de soporte y mantenimiento a cada estación de trabajo de forma conjunta que se encuentren conectadas como clientes a otra configurada como servidor puesto que éste software se basa en la arquitectura cliente/servidor (*Figura 1*), donde los clientes son los equipos que se van a instalar, gestionar o auditar, y el servidor quien provee a los clientes las imágenes de sistemas operativos. [6]

Dentro de las principales funcionalidades que posee el software están:

- Realizar la gestión y clonación de imágenes de sistemas operativos.
- Realizar la gestión y configuración de discos duros.
- Realizar reportes de inventario de todo el software y hardware instalado en el terminal cliente.
- Detección de amenazas de seguridad como uso indebido de los recursos de la red, el acceso ilegal a las estaciones o terminales y posibles daños maliciosos o la fuga de información sensible.
- Control del acceso de cada usuario al terminal cliente por perfiles.
- Conocer en detalle qué está haciendo cada persona en su terminal de acuerdo al perfil accedido.
- Auditar acciones, procesos y recursos en las máquinas y enviar informes de violación de seguridad al servidor para la toma de medidas.
- Encriptación del sistema de archivos.
- Resguardo de Información sensible.

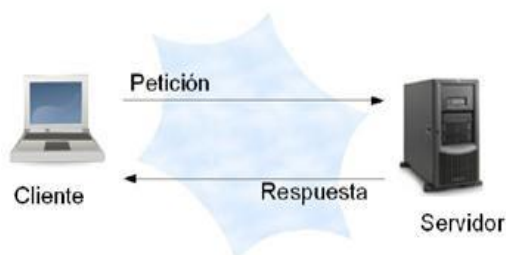


Figura 1. Arquitectura cliente – servidor

1.2.1 Sistemas de gestión y distribución de software

A continuación se hace referencia a algunos de los sistemas que permiten realizar la gestión de estaciones clientes y que son usados actualmente a nivel mundial. [7]

1.2.1.1 Landscape

Landscape es la herramienta desarrollada por Canonical, para la monitorización y administración centralizada de equipos que tengan como sistema operativo (SO) Ubuntu. Consta de una estructura cliente servidor, y toda la configuración y gestión de los equipos remotos se realiza mediante una interfaz web alojada en el servidor Landscape. Al estar diseñada específicamente para Ubuntu, se integra de manera muy fácil con las peculiaridades del SO pero el punto más vulnerable de su implantación es su costo en dólares (USD). [8]

Entre otras muchas funcionalidades, permite:

- **Instalación y gestión de paquetes.**
- **Gestión centralizada de configuraciones.**
- **Contar con un repositorio de paquetes centralizado.**
- **Ejecución de scripts en máquinas remotas.**
- **Monitorización de sistemas.**
- **Administrar máquinas apagadas o por su término en inglés “off-line”.**
- **Gestión de usuarios, grupos y tareas programadas.**

1.2.1.2 Cfengine

Cfengine es una herramienta de gestión centralizada de equipos que utiliza un lenguaje de alto nivel basado en políticas de configuración. Es software libre bajo licencia GPL (General Public License) y su autor, Mark Burgess, empezó a desarrollarlo en 1993 en la Universidad de Oslo con el objetivo de crear un framework portable para gestionar las diferentes variantes de UNIX basándose en sus propias tesis sobre la convergencia de sistemas. El software posee una estructura cliente-servidor y está principalmente diseñado para la administración y gestión de ficheros de configuración realizando la administración a través de consola. Mediante este sistema también se pueden gestionar puntos de montaje, trabajos cron y paquetes instalados. Al igual que Landscape, soporta la gestión de equipos desconectados. Empresas como Facebook, Google y NASA lo utilizan para controlar infraestructuras con más de 100.000 servidores. [9]

Dentro de las principales funcionalidades que posee el software están:

- **Comprobar y establecer permisos y propietario/grupo en ficheros/directorios:** permite que los directorios necesarios tengan la configuración de seguridad recomendada en UNIX.

- **Gestión de paquetes software RPM y DPKG:** cuenta con un módulo con el que podemos realizar tareas básicas relacionadas con la gestión de paquetes de software: instalar, borrar, actualizar, etc. abstrayendo el sistema operativo sobre el que se ejecuta el agente (Linux, FreeBSD, etc.) y el gestor de paquetes utilizado (RPM, DPKG, etc.)
- **Gestión de procesos y servicios del sistema:** las políticas en Cfengine para gestionar servicios se ejecutan al arrancar la máquina, esto permite comprobar que un determinado proceso se está ejecutando (monitorización) y que no tenga más de un determinado número de hijos.
- **Edición de ficheros:** es uno de los módulos más potentes de Cfengine que permite realizar cualquier tipo edición sobre ficheros: reemplazar una cadena, comentarla, eliminar una línea, agregar un texto, etc.

1.2.1.3 Puppet

Puppet es una herramienta profesional utilizada para la gestión centralizada de equipos. Fue desarrollada por Puppet Labs con Ruby on Rails 2.3 y se basa en la arquitectura cliente-servidor (Ver Figura 1). Este software fue diseñado principalmente para la administración y gestión de ficheros de configuración. La configuración se realiza íntegramente desde consola y desde un servidor central que gestiona las configuraciones pues los clientes se conectan periódicamente y envían informes al servidor, ésta comunicación se realiza sobre REST (antes era XML-RPC). Utiliza un lenguaje de alto nivel (aunque también permite el trabajo a bajo nivel, permitiendo así un balance de especificación muy grande entre las configuraciones de los servidores). Esta novedosa herramienta es multiplataforma pues funciona en computadoras que tengan como SO RedHat, Fedora, Gentoo, FreeBSD, Mac OS X, Microsoft Windows (para este SO está un poco limitado) y la usan la red social Twitter y la reconocida corporación Sun/Oracle (fruto de la unión de Sun con Oracle). Puppet es en la actualidad una alternativa a Cfengine. [10]

Puppet permite:

- **Gestionar configuraciones y ficheros.**
- **Gestionar usuarios y grupos.**
- **Gestionar paquetes.**
- **Administrar recursos tales como: user, file, package, service y cron.**
- **La integración con repositorios de datos como el subversion.**

1.2.1.4 Spacewalk

Spacewalk es una solución libre de gestión para sistemas Linux/Solaris. Es el sistema en el que está basado "Red Hat Network Satellite", el sistema de gestión de servidores centralizado que utiliza RedHat. Funciona en sistemas que posean RedHat y Fedora y en distribuciones derivadas de éstas, tales como Centos, Scientific Linux, etc. El soporte para Debian está siendo desarrollado

actualmente, pero aún no es usable, ya que está en una fase experimental, es por esto que Spacewalk no es compatible con sistemas Debian ni con sus derivados como Ubuntu. [11]

Spacewalk permite:

- **Inventariado de hardware y software de sistemas.**
- **Instalación y actualización de software en los equipos.**
- **Monitorización.**
- **Despliegue de configuraciones unificadas.**
- **Gestionarlo todo desde una única interfaz gráfica.**

1.2.1.5 Dameware Nt Utilities

Dameware Nt Utilities (DNTU) es un sistema empresarial de gestión de estaciones para Windows NT/2000/XP/2003/Vista/7 que está diseñado para permitir a los administradores tener el control remoto sobre los ordenadores cliente. Proporciona una colección integrada de utilidades de administración, que se incorpora dentro de una interfaz centralizada para la administración remota de servidores con sistema operativo Windows y estaciones de trabajo. Este software es capaz de correr como una aplicación o como un servicio. Algunos de los usuarios más reconocidos a nivel mundial que utilizan este software son: AOL, AT&T, Coca Cola, EBay, ESPA, FOX, Good Year, HITACHI, IBM, HP, INTEL, McDonald's, Motorola, NIKON, Nvidia, PEPSI, SIEMENS, Sony y T-Mobile. [12]

Dentro de las principales funcionalidades que posee el software están:

- **Controlar remotamente el escritorio en forma segura** (debido a su alto nivel de encriptación).
- **Control de teclado, mouse e inactividad** (la cantidad de equipos a controlar es ilimitada).
- **Instalación remota de servicios.**
- **Modo solo observación.**
- **Autorización requerida.**
- **Auditoría de Logs.**
- **Filtro de IP y Proxys.**
- **Que es una aplicación multi-threaded:** Esto permite mejorar el rendimiento y la velocidad pues no se tiene que esperar a que una Vista se termine de cargar antes de comenzar otra tarea dentro del software.

1.2.1.6 Secospace TSM

Secospace Tivoli Storage Manager (TSM) es un software desarrollado por Huawei Technologies, empresa privada de alta tecnología que se especializa en el equipamiento de comunicaciones y provee soluciones de redes personalizadas para éste tipo de industria. Secospace TSM es una

herramienta modular que es capaz de hacer uso de diferentes sistemas existentes para el monitoreo, detección de intrusos, sistemas de control de puertos remotos, así como de asistencia técnica remota. [13]

Dentro de las principales funcionalidades que posee el software están:

- **Mantener terminales ilegales separadas o aislar y recuperar terminales inseguros:** usando la autenticación de identidad y las políticas de seguridad para usuarios de la terminal este sistema permite separar las estaciones ilegales que no cumplan con los requisitos.
- **Controlar el acceso de todos los empleados y clientes en general:** bloquea o pone en cuarentena a usuarios inválidos en la red, combate además virus, gusanos y softwares espías.
- **Supervisar las auditorías y los comportamientos de las terminales de la red empresarial:** monitorea y audita el comportamiento de los usuarios para asegurarse que se está haciendo un uso racional de los recursos de la red, mejorando el tráfico en la red y estableciendo acciones preventivas ante ataques maliciosos.
- **Administrar el software y hardware activo:** realizando además auditorías a los clientes, tales como el monitoreo de los medios de almacenamiento USB y los procesos del sistema.
- **Administración centralizada y flexible basada en niveles y permisos:** permite la gestión desde determinados lugares a administradores con diferentes permisos, permitiéndoles la administración de sus computadoras asignadas.
- **Soporte flexible y fácil despliegue para conocer los requerimientos de todo el entorno de red:** para que el Security Access Control Gateway (SACG) sea conectado al router en modo directo o bypass.

1.2.1.7 Plataforma de software distribuido (T-arenal)

T-arenal es un producto informático desarrollado al igual que Nova - Xerberos en la Universidad de las Ciencias Informáticas. Fue desarrollado en el año 2008 por el Ing. Longendri Aguilera Mendoza. [14] Ha sido utilizado para dar solución a varios problemas de la bioinformática, y brinda un modelo de programación de alto nivel basado en el paradigma de la POO. T-arenal ofrece una alternativa de cómputo distribuido que aglutina en un solo conjunto varias estaciones de trabajo sin intentar eliminar o pretender aminorar las amplias posibilidades de aplicación de los modelos paralelos, sino de complementar todos los medios disponibles en una gran “supercomputadora virtual” o sea que todas las computadoras de una red existente funcionen como un todo virtualmente. T-arenal es multiplataforma (para sistemas operativos tanto GNU/Linux como Windows) ya que fue implementado en lenguaje Java. Está dividida en tres componentes esenciales: servidor central, servidor de peticiones y cliente.

Principales funcionalidades del software:

- **Gestionar cuentas por usuario o por grupos.**
- **Gestionar Elementos de Cómputos por rangos de IP.**
- **Generar reportes de datos gráficamente.**
- **Gestionar clientes, problemas y soluciones.**
- **Gestionar Ejecuciones (por estado de los procesos: en espera, ejecutándose, finalizado, en cola, pausado).**
- **Gestionar Actualizaciones de Clientes.**
- **Brindar información del servidor central.**
- **Gestionar Servidores de Peticiones.**
- **Realizar reportes estadísticos (gráficamente).**
- **Brindar opciones de configuración.**

1.2.1.8 COSMO

COSMO es un sistema informático que permite la gestión de estaciones. Fue desarrollado por la Ing. Elizabeth Santana Beoto perteneciente al Instituto Superior Politécnico José Antonio Echeverría (ISPJAE) en el año 2010 y presentado para optar por el título de máster en ciencias. Fue desarrollado primeramente con el objetivo de gestionar los laboratorios o agrupaciones de computadoras en el ISPJAE pero posteriormente ha sido utilizado en otros organismos de la nación como por ejemplo en la UCI. Cosmo está basado en la arquitectura Cliente – Servidor y la misma está implementada de forma modular [15]. En el servidor se encuentran módulos servidores que interactúan con diferentes orígenes de información (base de datos, servidor de aplicaciones, páginas Web, etc.), que a su vez se comunican con los módulos homólogos en el cliente. Cosmo funciona sobre Windows Server 2003/XP o superior y en cualquier distribución de Linux. El sistema Cosmo está estructurado por tres partes fundamentales: servidor Cosmo, cliente Cosmo, administración Web como se muestra en la figura.

Principales funcionalidades del software:

- **Controlar las horas de inicio y fin de sesión.**
- **Chequear la autenticación de los usuarios y la concurrencia.**
- **Verificar conectividad (chequear red para verificar si está conectado el cable de red).**
- **Aplicar la lista de programas prohibidos y evitar su ejecución.**
- **Determinar la cantidad de usuarios promedio que usan los laboratorios o agrupaciones de computadoras.**
- **Determinar la cantidad e identidad de los usuarios que están usando las computadoras en el instante actual.**
- **Obtener las trazas de uso de una computadora o de un usuario.**

- **Obtener la lista de procesos activos en los clientes y los programas más usados.**
- **Crear módulos para la gestión del estado, composición, responsable y operadores de los laboratorios.**
- **Generar reportes de Logs (datos de usuarios, números de IP accedidos, filtros de fecha y hora).**

El análisis profundo de todos estos sistemas nacionales e internacionales posibilita que en capítulos posteriores donde se confeccionen los criterios de evaluación de los diferentes factores de calidad para el sistema Nova – Xerberos se pueda tener una base de conocimiento amplia y establecer los parámetros de evaluación correctos en función de obtener correctitud, completitud, eficiencia, seguridad y fiabilidad en el sistema. Para ello es que se ha estudiado a fondo tanto las funcionalidades como las características operacionales de los sistemas antes mencionados, obteniendo así una serie de elementos que ayuden a comparar las oportunidades en el mercado, apoyando por tanto, el aumento de la competitividad del producto.

1.3 Garantía de calidad del software

La garantía de calidad abarca todas aquellas actividades o prácticas que se realizan con el objetivo de asegurar un cierto nivel de calidad en el producto desarrollado. [16]

Es una disciplina de la ingeniería de software que se especializa en la aplicación de procesos de calidad a lo largo del proyecto de software.

Las áreas que caen bajo la responsabilidad de la garantía de la calidad se pueden sintetizar en:

- 1.- **Las metas y objetivos:** Se debe asegurar que las metas de la organización en primer lugar, y los objetivos del usuario en segundo lugar se están satisfaciendo, y que no existen conflictos entre ellos, o entre los objetivos de diferentes usuarios.
- 2.- **Los métodos:** Se debe asegurar que las actividades de desarrollo de software siguen los procedimientos establecidos, se ajustan a los estándares seleccionados, están de acuerdo con las políticas de la organización y se ejecutan según las guías de trabajo y recomendaciones disponibles.
- 3.- **Rendimiento:** Se debe asegurar que se optimiza la utilización del hardware y software en los productos desarrollados, que son económicos (se desarrollan con el menor coste posible), eficientes (sacan el máximo partido posible a los recursos utilizados) y efectivos (alcanzan el resultado deseado con la menor cantidad posible de recursos, tiempo y esfuerzo).

Las principales tareas del grupo de garantía de calidad, por lo tanto, son:

- 1.- **Planificación de la calidad:** Consiste en seleccionar, clasificar y ponderar las propiedades de calidad que se van a establecer como requisitos, con respecto al producto y con respecto

al proceso. Se elegirán también los mecanismos de control de calidad a utilizar para medir y evaluar estas características y se determinarán las metas a alcanzar.

2.- **Supervisión de la calidad:** Consiste en supervisar y corregir, si es necesario, el trabajo que se está realizando (según los resultados obtenidos con las actividades de control de calidad), con el objetivo de llegar a satisfacer los requisitos establecidos.

3.- **Construcción de la calidad:** La constituyen las actividades constructivas que son aquellas que sirven para “construir” la calidad, es decir, son actividades preventivas cuyo objetivo es evitar la introducción de errores mediante la puesta en práctica de ciertos principios, métodos, formalismos y herramientas.

Por ejemplo: La formación del personal y la motivación, la aplicación de principios, técnicas y herramientas de Ingeniería de Software y la aplicación de modelos de proceso o planes.

La siguiente gráfica muestra la evolución de la garantía de la calidad en el desarrollo de proyectos de software, desde el primer eslabón que son las inspecciones internas por el departamento de calidad, pasando por el autocontrol, las actividades y buenas prácticas para asegurar la calidad hasta lo que idealizan o desean perseguir todos los que laboran con la finalidad de obtener un producto eficiente, fácil de mantener, interoperable, multiplataforma, fácil de usar y sobre todo corregible que es la gestión de la calidad o también conocido como calidad total.

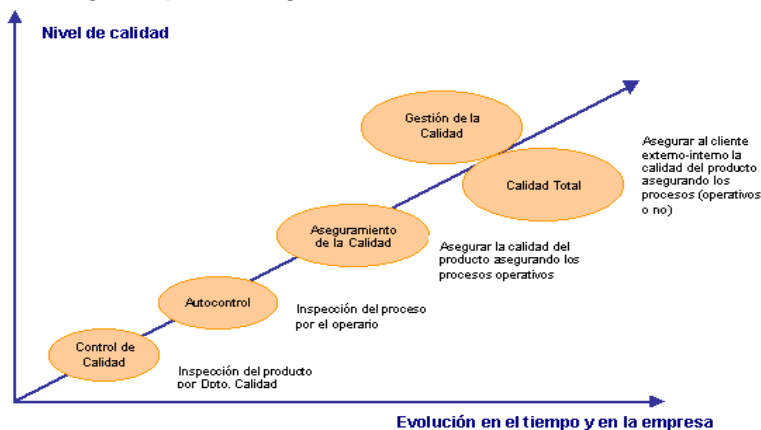


Figura 3. Evolución de la garantía de la calidad del software.

1.4 Aseguramiento de la Calidad del Software

El Aseguramiento de la Calidad del Software está dado por el conjunto de actividades planificadas y sistemáticas que se necesitan para lograr que el producto sea funcional, operacional y basado en las especificaciones previas de los clientes [17]. Ésta planificación debe hacerse antes de comenzar el proceso de desarrollo del software. Pressman define que para Asegurar la Calidad se deben hacer evaluaciones, auditorías y revisiones en las diferentes etapas del desarrollo del proyecto, se deben aplicar estándares, mecanismos de medida (métricas), métodos y herramientas para el análisis, el diseño, la programación y prueba, la documentación y el control de software. Se conoce mundialmente este proceso como Quality Assurance (QA, por sus siglas en inglés).

1.4.1 Actividades para el aseguramiento de la calidad del software

A continuación se relacionan un conjunto de actividades que forman parte de las buenas prácticas que deben tener en cuenta los equipos de trabajo a lo largo de todo el proceso de desarrollo para el aseguramiento de la calidad en proyectos de software.

- Utilizar métricas de software para el control y la planificación del proyecto.
- Verificar y validar el software a lo largo de todo el ciclo de vida.
- Realizar de pruebas de "caja negra", de "volumen y stress" y procesos de revisión e inspección.
- Realizar la gestión de la configuración del software.
- Garantizar una adecuada administración en el proyecto.
- Lograr administrar y erradicar los riesgos potenciales.
- Registrar lo que no se ajuste a los requisitos.
- Asegurar la documentación de las desviaciones.
- Revisar paulatinamente la descripción del proceso para asegurar:
 - Ajuste a la política de la empresa.
 - Cumplimiento de estándares internos y externos.

Métricas del software

Una *métrica* es una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado, definición que brinda el glosario de términos de la IEEE (Institute of Electrical and Electronics Engineers). [20] Las métricas nos ayudan a entender tanto el proceso técnico que se utiliza para desarrollar un producto, como el propio producto, nos permiten cuantificar y por consiguiente gestionar de forma más efectiva los recursos materiales o personales.

Hay varias razones por las cuales se mide un producto.

- 1.- Para indicar la calidad del mismo.
- 2.- Para evaluar la productividad de las personas que desarrollan el producto.
- 3.- Para evaluar los beneficios en términos de productividad y de calidad, derivados del uso de nuevos métodos y herramientas de la ingeniería de software.
- 4.- Para establecer una línea que sirva de base para la estimación de factibilidad.
- 5.- Para ayudar a justificar el uso de nuevas herramientas o de formación adicional.

La UCI se encuentra inmersa en un proceso de mejora en su versión 3.3 (Ver acápite 1.11), proceso que ayuda a la institución a llevar a cabo un desarrollo de software centralizado en función de obtener el 2do nivel de CMMI con todas las características que lleva embebidas este nivel, para ahondar en esta temática se recomienda consultar los *Procesos y Guías* que define dicho programa referente a ésta temática de las métricas, específicamente el documento **IPP-3550-**

En Nova – Xerberos, para garantizar el aseguramiento de la calidad, se utilizan *métricas* definidas por este proceso, que trata de ser genérico para toda la institución universitaria y pretende ayudar a que el desarrollo de software en este centro de altos estudios sea de primer nivel. Para ello se rigen por lo establecido en el documento **MA 0326_Definición de Medidas y Procedimientos de Recolección y Almacenaje.pdf**, en este documento se define primeramente una cantidad determinada de *indicadores* y luego cómo se obtiene cada uno de ellos, su medida base, la fuente, el método de medición, la escala, la unidad de medida, el límite de los datos, detalles de la definición operacional, origen y dueño de la información a medir, cómo se recolecta, cuándo y con qué frecuencia se recolecta, el responsable de la recolección, dónde y cómo se almacena el indicador, quién tiene acceso al lugar de almacenamiento (como medida de seguridad) y el responsable de almacenaje del indicador.

Verificación y Validación

Dentro de las actividades de verificación y validación ligadas al control de la calidad se incluyen las pruebas y procesos de revisión y auditorías.

Las revisiones según el estándar ANSI/IEEE Std. 1028:1988 se definen como: evaluación de un elemento para determinar diferencias con los resultados planeados y recomendar mejoras.

Las auditorías según el estándar ISO 19011:2002 se definen como: proceso sistemático, independiente y documentado para evaluar el estado actual de manera objetiva con el fin de determinar la extensión en que se cumplen los criterios de auditoría.



Figura 2. Calidad en el ciclo de vida del software

Gestión de la configuración del software

La gestión de configuración del software (GCS) es una actividad de protección que se realiza a lo largo de todo el desarrollo de software. Considerando que durante el proceso de desarrollo de software el cambio puede producirse en cualquier momento, las actividades que plantea la GCS

permiten, identificar, controlar y garantizar que el cambio se realiza adecuadamente e informar del mismo a todos los interesados. [22]

- **Actividades de la GCS:**

- ❖ **Identificación:** Con esta actividad se trata de adecuar la documentación a estándares de documentación existentes.
- ❖ **Control de cambios:** Esta actividad consiste en la evaluación y registro de todos los cambios que se realicen en cuanto a configuración de software.
- ❖ **Auditorías de configuraciones:** Junto a las revisiones técnicas formales, esta actividad ayuda a lograr que el cambio se implemente correctamente.
- ❖ **Generación de informes:** Esta actividad es muy útil para la posterior revisión de los resultados arrojados por las revisiones y auditorías ya realizadas.

Para todo lo relacionado con la gestión y la configuración del software Nova – Xerberos hace uso de lo establecido en el Programa de Mejoras, utilizando los estándares de configuración que se definen en el documento **5602_Estándares de Configuración.pdf** donde se explica cuáles son los elementos configurables y cuáles no; los lineamientos para el control de versiones para el producto, la documentación y las líneas bases; la estructura del repositorio de los datos y el proceso de respaldo del contenido de la gestión de la configuración.

Plan de aseguramiento de la calidad

Este documento se realiza para adaptar las pautas marcadas por los sistemas de calidad a cada proyecto, se debe generar un plan específico para cada uno y es denominado plan de aseguramiento de la calidad, es conformado por el grupo de aseguramiento de la calidad del proyecto y sirve como guía para el desarrollo de las actividades de gestión de la calidad.

1.5 Control de Calidad de Software

Software Quality Assurance (SQA por sus siglas en inglés) se trata de la garantía de calidad del software: es el mecanismo que permite visualizar los procesos de ingeniería de software y verificar los métodos, estándares y procedimientos utilizados para asegurar la calidad del software y mediante el cual se pueden tratar y dar seguimiento a los problemas detectados.

Una vez seleccionados los índices de calidad, debe establecerse el proceso de control, que requiere los siguientes pasos: [17]

- Definir el software que va a ser controlado: clasificación por tipo, esfera de aplicación, complejidad, etc., de acuerdo con los estándares establecidos para el desarrollo del software.
- Seleccionar una medida que pueda ser aplicada al objeto de control pues para cada clase de software es necesario definir los indicadores y sus magnitudes.

- Crear o determinar los métodos de valoración de los indicadores: métodos manuales como cuestionarios o encuestas estándares para la medición de criterios periciales y herramientas automatizadas para medir los criterios de cálculo.
- Definir las regulaciones organizativas para realizar el control: quiénes participan en el control de la calidad, cuándo se realiza, qué documentos deben ser revisados y elaborados, etc.

Principales mecanismos utilizados para verificar los procesos de desarrollo:

- ❖ **Calidad de desarrollo:** Este mecanismo es el que guía la realización de los test unitarios, de integración y de regresión, permite detectar anomalías tempranamente y brinda buenas prácticas para las auditorías del código del software.
- ❖ **Bug Tracking público:** es un sistema de control y seguimiento de errores que permite sanear totalmente las aplicaciones y otros proyectos, asegurando que queden libres de fallos y como indica el nombre del mecanismo se hace con el público, para encontrar la mayor cantidad de defectos posibles. Es por ello que mediante la utilización de este mecanismo:
 - Se torna fácil para los usuarios apuntar a un defecto en específico encontrado.
 - El usuario puede informar que algo falla.
- ❖ **Ciclos de testing:** Este tipo de mecanismo es más bien destinado a la subsanación de todos los errores posibles del software antes de que salga al mercado y es por ello que:
 - Se programan "entregas" al Departamento de Calidad previas a la versión que se va a entregar al cliente.
 - Cada entrega tiene un objetivo claro y si la misma no alcanza resultados concretos pues se puede volver a repetir el proceso de pruebas.

1.6 Factores de calidad del software

Se conocen como factores de calidad del software a los factores que determinan cualidades tanto internas como externas que debe poseer un software. [18]

A continuación se realiza un compendio de los principales modelos y normas que rigen los factores de calidad del software. Luego del estudio minucioso de ellos se podrá definir cuáles de los factores serán críticos o indispensables para el proyecto que es objeto de estudio, ellos son: los modelos McCall [23] y Boehm [24].

1.6.1 Modelo McCall

El modelo de McCall fue presentado en 1977. Se focaliza en el producto final, identificando atributos claves desde el punto de vista del usuario, estos atributos se denominan factores de

calidad y son principalmente atributos externos aunque están presentes también algunos atributos internos.

Este modelo organiza los factores en tres ejes o puntos de vista desde los cuales el usuario puede contemplar la calidad de un producto, basándose en once factores de calidad organizados en torno a los tres ejes y a su vez cada factor se desglosa en otros criterios:

Puntos De Vista O Ejes	Factor	Criterios
OPERACIÓN DEL PRODUCTO	Facilidad de uso	<ul style="list-style-type: none"> - Facilidad de operación: Atributos del software que determinan la facilidad de operación del software. - Facilidad de comunicación: Atributos del software que proporcionan entradas y salidas fácilmente asimilables. - Facilidad de aprendizaje: Atributos del software que facilitan la familiarización inicial del usuario con el software y la transición del modo actual de operación. - Formación: El grado en que el software ayuda para permitir que nuevos usuarios apliquen el sistema.
	Integridad	<ul style="list-style-type: none"> - Control de accesos: Atributos del software que proporcionan control de acceso al software y los datos que maneja. - Facilidad de auditoría: Atributos del software que facilitan la auditoría de los accesos al software. - Seguridad: La disponibilidad de mecanismos que controlen o protejan los programas o los datos.
	Corrección	<ul style="list-style-type: none"> - Compleitud: Atributos del software que proporcionan la implementación completa de todas las funciones requeridas. - Consistencia: Atributos del software que proporcionan uniformidad en las técnicas y notaciones de diseño e implementación. - Trazabilidad o rastreabilidad: Atributos del software que proporcionan una traza desde los requisitos a la implementación con respecto a un entorno operativo concreto.
	Fiabilidad	<ul style="list-style-type: none"> - Precisión: Atributos del software que proporcionan el grado de precisión requerido en los cálculos y los resultados. - Tolerancia a fallos: Atributos del software que posibilitan la continuidad del funcionamiento bajo condiciones no usuales. - Modularidad: Atributos del software que proporcionan una estructura de módulos altamente independientes. - Simplicidad: Atributos del software que posibilitan la implementación de

		funciones de la forma más comprensible posible.
	Eficiencia	<ul style="list-style-type: none"> - Eficiencia en ejecución: Atributos del software que minimizan el tiempo de procesamiento. - Eficiencia en almacenamiento: Atributos del software que minimizan el espacio de almacenamiento necesario.
REVISIÓN DEL PRODUCTO	Facilidad de mantenimiento	<ul style="list-style-type: none"> - Modularidad, Simplicidad, Consistencia. - Concisión: Atributos del software que posibilitan la implementación de una función con la menor cantidad de códigos posible. - Auto descripción: Atributos del software que proporcionan explicaciones sobre la implementación de las funciones.
	Facilidad de prueba	<ul style="list-style-type: none"> - Modularidad, Simplicidad, Auto descripción. - Instrumentación: Atributos del software que posibilitan la observación del comportamiento del software durante su ejecución para facilitar las mediciones del uso o la identificación de errores.
	Flexibilidad	<ul style="list-style-type: none"> - Auto descripción. - Capacidad de expansión: Atributos del software que posibilitan la expansión del software en cuanto a capacidades funcionales y datos. - Generalidad: Atributos del software que proporcionan amplitud a las funciones implementadas. - Modularidad.
TRANSICIÓN DEL PRODUCTO	Reusabilidad	<ul style="list-style-type: none"> - Auto descripción, Generalidad, Modularidad. - Independencia entre sistema y software: Atributos del software que determinan su dependencia del entorno operativo. - Independencia del hardware: Atributos del software que determinan su dependencia del hardware.
	Interoperabilidad	<ul style="list-style-type: none"> - Modularidad. - Compatibilidad de comunicaciones: Atributos del software que posibilitan el uso de protocolos de comunicación e interfaces estándar. - Compatibilidad de datos: Atributos del software que posibilitan el uso de representaciones de datos estándar. - Estandarización en los datos: El uso de estructuras de datos y de tipos estándar a lo largo de todo el programa.

Tabla 1. Factores para determinar la calidad de un software (Modelo McCall).

Para aplicar este modelo es necesario especificar los requisitos de calidad del producto software teniendo en cuenta:

- Las características particulares del producto que se está diseñando.
- La determinación de las etapas del ciclo de vida donde es necesario evaluar cada factor de calidad para conocer en cuáles se dejan sentir más los efectos de una calidad pobre con respecto a cada uno de los factores.

1.6.2 Modelo de Boehm

Este modelo fue presentado por su creador, Barry Boehm en 1978. Introduce características de alto nivel, características de nivel intermedio y características primitivas, cada una de las cuales contribuye al nivel general de calidad. Se identifican en él características de calidad desde el punto de vista del usuario. Existe una gran similitud entre este modelo y el de McCall pues ambos tratan las características de alto nivel (representan requerimientos generales de usabilidad) pero en este se incluyen características de desempeño del hardware, omitidas en el de McCall.

Boehm planteaba como características principales en los sistemas (*alto nivel*):

- **utilidad per-se** cuán usable, confiable y eficiente es el producto en sí mismo.
- **mantenibilidad** cuán fácil es modificarlo, entenderlo y retestearlo.
- **utilidad general** si puede seguir usándose si se cambia el ambiente.

En este modelo Boehm divide la calidad en tres características: usos principales, componentes intermedios y componentes primitivos y explica que todos en su conjunto van a estar regidos por métricas para verificar su cumplimiento.

En el proyecto productivo Nova - Xerberos se aplican varios de los factores de calidad expuestos en el acápite anterior y teniendo en cuenta el tipo de software que se desarrolla y sus características los factores críticos de calidad que definen el software son: la **corrección**, la **eficiencia**, la **integridad** y la **fiabilidad**. El factor corrección se ve presente en 3 de sus atributos en Nova – Xerberos; *completitud* (porque se busca un desarrollo completo de todas las funcionalidades), *consistencia* (porque se trabaja en función de obtener uniformidad entre técnicas y notaciones de diseño e implementación) y *trazabilidad* (porque existe trazabilidad desde los requisitos a la implementación para lograr un correcto entorno operativo). El factor eficiencia se ve presente en cuanto a lograr eficiencia en la *ejecución* del software para elevar la velocidad del procesamiento de los datos y eficiencia en el *almacenamiento* pues se busca minimizar el espacio que se requiera para almacenar datos (reportes, instaladores, etc). El factor integridad se puede decir que es medular en este tipo de software de gestión debido a su capacidad de operar con estaciones (todo a través de la red), la gestión de recursos, usuarios y las acciones auditables que se pueden realizar desde él, si no se tiene en cuenta ni el *control de accesos* (de personal no autorizado) ni la *seguridad* (con mecanismos que controlen o protejan el software y los datos) pues el sistema que se desarrolla no posee los parámetros de calidad de software requeridos. El

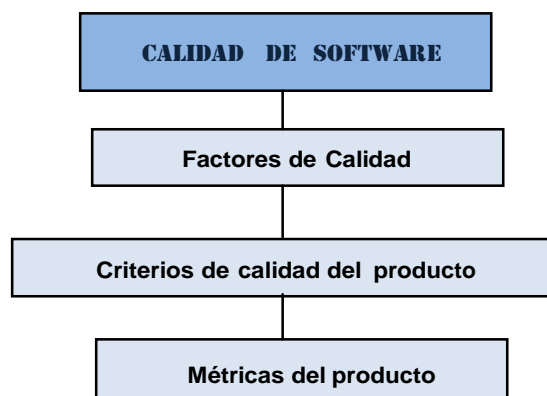
factor fiabilidad se puede hacer palpable en Nova – Xerberos por la *precisión* (vista por la exactitud que posee el software en la realización de los cálculos y operaciones que ejecutan las diferentes funcionalidades) y la *modularidad* con que está definido el software (está diseñado por módulos que se integran como un todo pero, pensados para ser implementados por separado sin interferir unos con otros ni hacer muy engorroso el proceso de su implementación).

1.7 Modelos de Calidad de Software

Los modelos de calidad de software son los que sirven de guía para el desarrollo gestionado del proceso de desarrollo del software, ofreciendo una definición más operacional para la elaboración del producto.

En los modelos de calidad, la calidad se define de forma jerárquica. Es un concepto que se deriva de un conjunto de sub-conceptos, cada uno se va a evaluar a través de un conjunto de indicadores o métricas. [25]

Tienen una estructura, por lo general, en tres niveles:



La ventaja de los modelos de calidad es que la calidad se convierte en algo concreto, que se puede definir, que se puede medir y, sobre todo, que se puede planificar.

Los modelos de calidad ayudan también a comprender las relaciones que existen entre diferentes características de un producto software. A continuación se detallan los modelos SPICE (ISO/IEC 15504) y CMMI.

1.7.1 Modelo SPICE (ISO/IEC 15504)

Es uno de los modelos más correctos para la mejora y evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software. Valora la arquitectura que se define en los procesos y prácticas aconsejables. Conocido como Software Process Improvement and Capability Determination (SPICE, por sus siglas en inglés). [2]

Características del Modelo ISO/IEC 15504:

- ❖ Establece un marco y los requisitos para cualquier proceso de evaluación y proporciona requisitos para los modelos a ser utilizados.
- ❖ Es aplicable para la evaluación y mejora de procesos y ayuda en la determinación de capacidades del equipo de trabajo.
- ❖ Proporciona en su parte 5 un Modelo de evaluación para los procesos del ciclo de vida del software definidos en el estándar ISO/IEC 12207 que define los procesos del ciclo de vida del desarrollo, mantenimiento y operación de los sistemas de software.

Niveles del Modelo SPICE ISO/IEC 15504:

El modelo fue aprobado en 1998, contempla la evaluación y mejora por niveles de madurez, es decir, diferentes estados en los que puede encontrarse una organización en función de la calidad de sus procesos. Estos niveles de madurez van desde el nivel más básico ó 0, hasta el más maduro el 5. Cada uno de los niveles tiene un conjunto de procesos asociados que están definidos en la Norma ISO/IEC 12207 y son específicos para el desarrollo del software. Este modelo es similar al de CMMI que también se utiliza para la mejora de procesos y determinación de la capacidad de madurez sólo que CMMI está orientado directamente a organizaciones y SPICE se enfoca en los procesos de las organizaciones.

Niveles de madurez de SPICE:

- **0: no realizada:** no hay productos de trabajo identificables.
- **1: realizada informalmente:** planificación y seguimiento dependientes del conocimiento individual. Productos de trabajo identificables.
- **2: planificada:** verificada de acuerdo a los procedimientos especificados.
- **3: bien definida:** procesos bien definidos y documentados
- **4: controlada cuantitativamente:** medidas detalladas de realización, predicción, etc. Productos de trabajo evaluados cuantitativamente.
- **5: mejorada continuamente:** objetivos cuantitativos de eficiencia basados en los objetivos de negocio.

1.7.2 Modelo CMMI

Capability Maturity Model Integration (CMMI, por sus siglas en inglés), es un modelo de referencia para el crecimiento de capacidades y madurez enfocado en procesos de administración, ingeniería de sistemas y de software enmarcado en la mejora continua de los procesos de la organización, empresa o lugar en particular donde se utilice.

Fue desarrollado por el Carnegie Mellon SEI (Software Engineering Institute), creado por el DoD

(Department of Defense) de los Estados Unidos [26]. El proceso de mejora que se lleva a cabo en la UCI constituye la adaptación de este modelo a los procesos de los proyectos de la universidad.

Estructura de CMMI:

Este modelo está estructuralmente compuesto por 5 niveles de madurez como evidencia la siguiente figura.



Figura 4. Niveles de madurez de CMMI

- **Inicial o Nivel 1:** Este es el nivel en donde se encuentran las empresas que no tienen procesos o que los mismos son impredecibles y pobremente controlados. Los presupuestos se disparan, no es posible entregar el proyecto en fechas. Los resultados dependen de la competencia y esfuerzos de las personas. Hay carencia de procedimientos formales y mecanismos de administración y control. El estado y desarrollo del proyecto es completamente incipiente o nulo.

- **Repetible o Nivel 2:** En este nivel existen procesos definidos y administrados. La principal diferencia entre este nivel y el anterior es que el proyecto es gestionado y controlado durante el desarrollo del mismo. El desarrollo no es tan pobre y se puede saber el estado del proyecto en todo momento. Los recursos utilizados son los adecuados y se asignan responsabilidades.

El éxito de los resultados alcanzados se puede repetir. Los procesos que hay que implantar para alcanzar este nivel son:

- Gestión de requisitos.
- Planificación de proyectos.
- Seguimiento y control de proyectos.
- Gestión de proveedores.
- Aseguramiento de la calidad.
- Gestión de la configuración.

- **Definido o Nivel 3:** Alcanzar este nivel significa que la forma de desarrollar proyectos (gestión e ingeniería de software) está establecida, documentada y que existen métricas (obtención de datos precisos) para poder lograr objetivos concretos. Se identifican y se resuelven las necesidades de capacitación. En esta fase la organización dispone de un conjunto de procesos estándares, que cada proyecto particular puede adaptar en función de sus necesidades. Los procesos que hay que implantar para alcanzar este nivel son:

- Desarrollo de requisitos.
- Solución Técnica.
- Integración del producto.
- Verificación.
- Validación.
- Desarrollo y mejora de los procesos de la organización.
- Definición de los procesos de la organización.
- Planificación de la formación.
- Gestión de riesgos.
- Análisis y resolución de toma de decisiones.

La mayoría de las empresas que llegan al nivel 3 se detienen en él, debido a que es un nivel que proporciona muchos beneficios y no ven la necesidad de ir más allá porque tienen cubiertas la mayoría de sus necesidades.

- **Cuantitativamente Gestionado o Nivel 4:** En este nivel lo que hay es que establecer y mantener objetivos cuantitativos para el proceso, los cuales estén orientados a la calidad y al desempeño del proceso y basados en las necesidades del cliente y en los objetivos del negocio. Es recomendable también estabilizar el desempeño de uno o más subprocesos para determinar la habilidad general del proceso. Los procesos que hay que implantar para alcanzar este nivel son:

- Gestión cuantitativa de proyectos.
- Mejora de los procesos de la organización.

- **Optimizado o Nivel 5:** En este nivel es necesario asegurar la mejora continua del proceso al satisfacer los objetivos del negocio relevantes a la organización y corregir los pequeños problemas que puedan surgir.

Los procesos que hay que implantar para alcanzar este nivel son:

- Innovación organizacional.
- Análisis y resolución de las causas.

Es válido señalar que este modelo es uno de los más fortuitos a la hora de verlo como posible candidato a utilizar por el amplio espectro de áreas de proceso que ocupa pero por tener un ciclo de vida tan largo es que puede ser muy costosa su puesta en marcha tanto económicamente como en el desgaste del personal que lo llevará a cabo pues requiere como mínimo de 10 personas para su realización.

Aseguramiento de la calidad del proceso y producto en CMMI

Entre las áreas de proceso, en la categoría de soporte para el nivel 2 de CMMI se encuentra PPQA (Process and Product Quality Assurance), en español, Aseguramiento de la calidad de Producto y Proceso. Esta es el área de proceso principal para el aseguramiento de la calidad del software dentro de CMMI.

Las metas de esta área de proceso son:

- Evaluar objetivamente la ejecución de los procesos, los elementos de trabajo y servicios en contraste a los procesos, estándares y procedimientos definidos.
- Proporcionar información a las personas que están usando los procesos y a la gerencia del proyecto, de los resultados de las actividades del aseguramiento de la calidad.
- Asegurar que las no conformidades son tratadas.
- Identificar y documentar no conformidades.

El siguiente es el listado de funcionalidades típicas para satisfacer esta área de proceso:

- Completar las plantillas de chequeo del cumplimiento de los procesos definidos en la organización dentro del sistema de control de documentación.
- Generar la convocatoria de una auditoría de calidad (personal participante y objetivos) dentro de la agenda de eventos compartida.
- Dejar registro de la ejecución de la auditoría de calidad dentro del sistema de control de documentación (por ejemplo rellorando las plantillas de chequeo predefinidas).
- Registrar las no conformidades dentro del sistema de gestión de incidencias.
- Generación automática de informes de auditorías en base al rellorado de las plantillas de chequeo y notificación a las personas implicadas.
- Posibilidad de generar un cuadro de mando que visualice las no conformidades encontradas en las auditorías de calidad de cada proyecto
- Registro del tiempo incurrido en aseguramiento de calidad, así como una estimación del ahorro de tiempo obtenido al encontrar no conformidades con anticipo, de manera que se pueda analizar si el tiempo en aseguramiento de la calidad (QA) es una buena inversión para la pequeña empresa.

La UCI (como institución rectora) al estar inmersa en la aplicación de este modelo (para alcanzar el nivel 2 de madurez) pues entonces es que Nova – Xerberos se está rigiendo también por lo planteado en este modelo pues Nova – Xerberos es uno de los tantos proyectos de software de esa institución.

1.8 Normas o estándares utilizados para gestionar la calidad de software

Para poder realizar un proceso de desarrollo de software correcto, se hace necesaria la utilización de una serie de pautas y patrones que se deben seguir para implementar un correcto sistema de

gestión y aseguramiento de la calidad. La organización internacional de estándares, ISO (International Standardization Organization, por sus siglas en inglés), ha producido una serie de estándares para esta finalidad conocidos colectivamente como ISO 9000. El propósito de la ISO es promover el desarrollo de la estandarización y de las actividades para facilitar el intercambio internacional de mercancías y de servicios, y para desarrollar la cooperación en actividad intelectual, científica, tecnológica y económica. En este sentido los estándares internacionales ISO constituyen un instrumento importante para alcanzar las metas descritas. Estas normas surgen luego de llevar a cabo un proceso de normalización que no es más que un documento que establece las condiciones mínimas que debe reunir un producto o servicio para que sirva al uso al que está destinado. [19]

Las normas que se relacionan debajo son las que servirán de sustento para posteriormente analizar cuáles se tendrán en cuenta en el proyecto Nova – Xerberos.

A continuación se describen detalladamente las principales normas ISO que influyen en la administración de la calidad ellas son: ISO 9001, ISO 9003, ISO 9004, ISO 1028 e ISO/IEC 12207.

1.8.1 ISO 9001

El objetivo de esta norma es especificar los **requisitos** para un sistema de gestión de la calidad que pueden utilizarse para su aplicación interna por las organizaciones, para certificación o con fines contractuales. [2]

Permite realizar certificaciones externas de la calidad, mediante cuatro áreas fundamentales, que son responsabilidad en la gestión, gestión de recursos, realización de productos y servicios y medición, análisis y mejora. Esta norma es la única capaz de certificar y está estrechamente relacionada con las actividades de planificar, realizar o hacer, verificar y actuar; acorde con el diseño, la producción, la instalación y el servicio post-venta. Los beneficios que consiguen las empresas al implementar un sistema de calidad según las normas ISO 9000 son considerables, pues permiten obtener una mayor satisfacción de los clientes por la confianza en los productos y servicios que brindan.

- **Requisitos generales**

- La organización debe establecer, documentar, implementar y mantener un sistema de gestión de la calidad y mejorar continuamente su eficacia de acuerdo con los requisitos de esta.
- La organización debe gestionar sus procesos de acuerdo con los requisitos de la norma.
- En los casos en que la organización opte por contratar externamente cualquier proceso que afecte la conformidad del producto con los requisitos, la organización debe asegurarse de controlar tales procesos. El control sobre dichos procesos contratados externamente debe estar identificado dentro del sistema de gestión de la calidad.

1.8.2 ISO 9003

La norma ISO 9003 [2] está destinada a sistemas de calidad, específicamente en la elaboración de modelos para aseguramiento de la calidad en la **inspección final** y **pruebas**. Este es el modelo contractual para los sistemas de calidad que no incluyen diseño o producción. ISO 9003 contiene cerca de la mitad de requerimientos de ISO 9001 y modifica algunos requerimientos para adecuarse a la inspección y la aplicación de prueba final. ISO 9003 requiere el desarrollo de un manual de calidad y procedimientos documentados que definan la organización y operación del sistema de calidad. Es responsabilidad de una compañía crear y mantener estos documentos, de modo tal que sean relevantes y apropiados a la operación específica del negocio.

1.8.3 ISO 1028

El propósito de la norma 1028 [2] es definir las **revisiones** sistemáticas aplicables a la adquisición de software, el suministro, el desarrollo, operación y mantenimiento. Esta norma describe cómo llevar a cabo una revisión.

Las revisiones pretenden:

- Señalar la necesidad de mejora.
- Confirmar las partes que no hay que mejorar.
- Conseguir mejorar la calidad

Define 5 tipos de revisiones:

- **Revisiones de gestión:** sirven para controlar el progreso y detectar inconsistencias de los planes con la programación y los requisitos.
- **Revisiones técnicas:** revisan la documentación producida a lo largo del proyecto.
- **Inspecciones:** revisiones que involucran al autor de un producto.
- **Walkthrough (recorrido):** inspecciones conducidas únicamente por miembros del grupo de desarrollo que examinan una parte específica del producto.
- **Auditorías:** evaluaciones independientes dirigidas a evitar el fraude o mal uso de las aplicaciones informáticas sobre el cumplimiento de estándares, planes, procedimientos, etc.

Es en esta norma donde aparecen las revisiones técnicas formales (rtf), es una actividad de garantía de calidad del software llevada a cabo por los ingenieros del software y otros. Las rtf se enfocan principalmente en comprobar hasta qué punto el producto está bien elaborado técnicamente.

Los que hagan uso de esta norma deberán especificar dónde y cuándo se aplica esta norma y cualquier desviación prevista de la misma.

1.8.4 ISO/IEC 12207

ISO/IEC 12207 [2] surge a principios de la década de los noventa. Establece **un proceso de ciclo de vida** para el software. Esta norma tiene como objetivo principal proporcionar una estructura común para que compradores, proveedores, desarrolladores, personal de mantenimiento, operadores, gestores y técnicos involucrados en el desarrollo de software usen un lenguaje común. Es aplicable a la adquisición de sistemas, servicios, al suministro, desarrollo, operación y mantenimiento a productos de software. Este lenguaje común se establece en forma de procesos bien definidos. Puede ser adaptada a las necesidades de cualquiera que la use.

Se basa en dos principios fundamentales: *modularidad y responsabilidad*. Con la modularidad se pretende conseguir procesos con un mínimo acoplamiento y una máxima cohesión. En cuanto a la responsabilidad, se busca establecer un responsable para cada proceso, facilitando la aplicación del estándar en proyectos en los que pueden existir distintas personas u organizaciones involucradas.

❖ **Procesos de soporte.**

- **Documentación:**
 - Define las actividades para el registro de la información producida por un proceso del ciclo de vida.
- **Gestión de la configuración:**
 - Define las actividades de la gestión de la configuración.
- **Aseguramiento de calidad:**
 - Define las actividades para asegurar, de una manera objetiva, que los productos software y los procesos son conformes a sus requerimientos especificados y se ajustan a sus planes establecidos. Revisión Conjunta, Auditoría, Verificación y Validación pueden ser utilizadas como técnicas de Aseguramiento de la Calidad.
- **Verificación:**
 - Define las actividades (para el adquiriente, proveedor o una parte independiente) para verificar hasta un nivel de detalle dependiente del proyecto del software, el producto o sistema).
- **Validación:**
 - Define las actividades (para el adquiriente, proveedor o una parte independiente) para validar el software.
- **Revisión conjunta:**
 - Define las actividades para evaluar el estado de una acción ejecutada o no. Este proceso puede ser empleado por cualquiera de las dos partes, donde una de las partes (la revisora) revisa a la otra parte (la parte revisada), de una manera conjunta.
- **Auditoría:**

- Define las actividades para determinar la conformidad con los requerimientos, planes y contrato. Este proceso puede ser empleado por dos partes cualesquiera, donde una parte (la auditora) audita el software o actividades de otra parte (la auditada).

○ **Resolución de problemas:**

- Define las actividades para analizar y eliminar los problemas (incluyendo las no conformidades) que sean descubiertos durante la ejecución del proceso de desarrollo, operación, mantenimiento u otros procesos, cualquiera que sea su naturaleza o causa.

1.9 Proceso de desarrollo de software

Un Proceso de Desarrollo de Software (PDS) es la definición del conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto, a modo de plantilla que explica los pasos necesarios para terminar el proyecto. Un proceso define quién, está haciendo, qué, cuándo y cómo para alcanzar un determinado objetivo. [27]

A pesar de la variedad de propuestas de procesos de software, existe un conjunto de actividades fundamentales que se encuentran presentes en todos ellos, éstas son:

- 1. Especificación de software:** Se debe definir la funcionalidad y restricciones operacionales que debe cumplir el software.
- 2. Diseño del sistema:** se deben aplicar ciertas técnicas y principios con el propósito de definir el modelado de un sistema o un proceso, con suficientes detalles como para permitir su interpretación y realización física. Esta es la disciplina o actividad que describe más claramente a los desarrolladores lo que el sistema tiene que hacer y cómo lo debe hacer.
- 3. Implementación:** la implementación puede ser la parte más obvia del trabajo de ingeniería de software, pero no necesariamente es la que demanda mayor trabajo ni la más complicada. La complejidad y la duración de esta actividad está íntimamente relacionada al o a los lenguajes de programación utilizados, así como al diseño previamente realizado.
- 4. Validación:** el software debe validarse, para asegurar que trabaje como fue diseñado y que cumpla con lo que quiere el cliente.
- 5. Evolución:** el software debe evolucionar, para adaptarse a las necesidades del cliente.

Además de estas actividades fundamentales, Roger Pressman menciona un conjunto de “actividades protectoras”, también conocidas como “sombrija” que se aplican a lo largo de todo el proceso del software. Ellas se señalan a continuación:

- Seguimiento y control de proyecto de software.
- Revisiones técnicas formales.

- Garantía de calidad del software.
- Gestión de configuración del software.
 - 1 - Preparación y producción de documentos.
 - 2 - Gestión de reutilización.
 - 3 - Mediciones.
 - 4 - Gestión de riesgos.

Elementos que caracterizan un proceso de desarrollo de software.

- ✓ **Un marco común del proceso**, definiendo un pequeño número de actividades del marco de trabajo que son aplicables a todos los proyectos de software, con independencia del tamaño o complejidad.
- ✓ **Un conjunto de tareas**, cada uno es una colección de tareas de ingeniería del software, hitos de proyectos, entregas y productos de trabajo del software, y puntos de garantía de calidad, que permiten que las actividades del marco de trabajo se adapten a las características del proyecto de software y a los requisitos del equipo del proyecto.
- ✓ **Las actividades de protección**, tales como garantía de calidad del software, gestión de configuración del software y medición. Son independientes de cualquier actividad del marco de trabajo y aparecen durante todo el proceso.

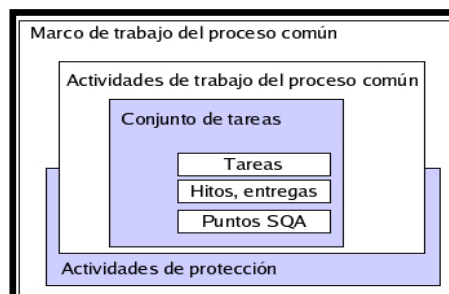


Figura 5. Estructura de los procesos de desarrollo de software

Cuando se desea realizar una gestión adecuada y eficaz del proceso de desarrollo es necesario que se pongan en funcionamiento cuatro elementos muy importantes: *personas*, *proyecto*, *procesos* y *producto*.

Las **personas** dentro de un proceso de desarrollo las constituyen seres humanos, que desempeñan diferentes roles, que pueden ser, por ejemplo: arquitecto, desarrollador, ingeniero de pruebas, analista, líder de proyecto, etc.

El **proyecto** es el elemento organizativo a través del cual se gestiona el desarrollo de software.

Los **procesos** los constituyen el conjunto de actividades necesarias para transformar los requisitos de usuario en un sistema.

El **producto** lo conforman los artefactos que se crean durante la vida del proyecto, como los modelos, el código fuente, los ejecutables y la documentación que es generada.

En la figura que se muestra a continuación se representa un esquema que ilustra el vínculo que existe entre éstos 4 elementos y todo el conjunto de actividades, participantes, elementos y recursos que influyen en la realización de un software.

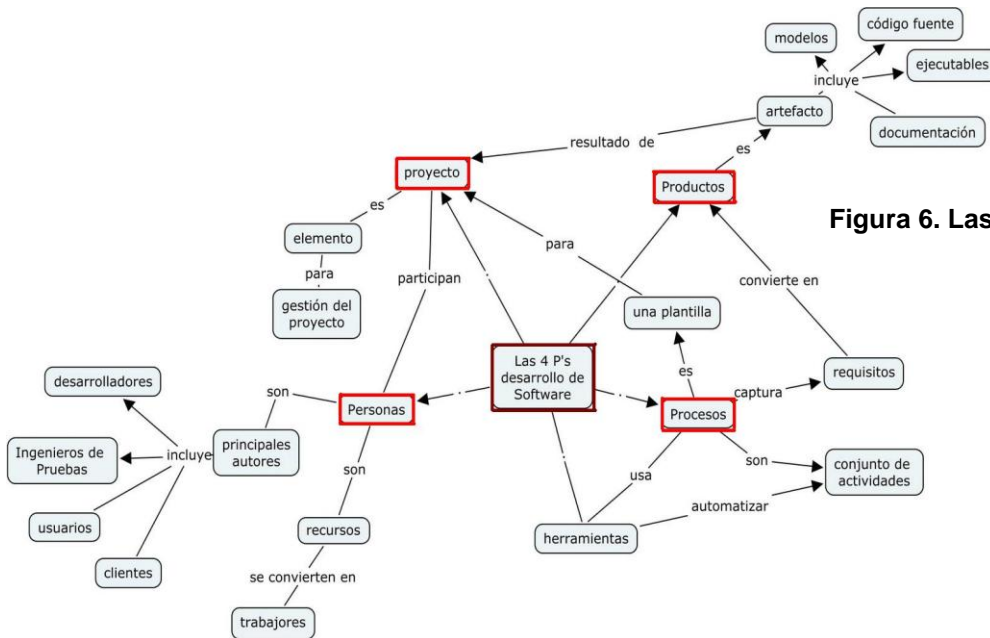


Figura 6. Las 4 P del desarrollo del software.

1.9.1 Proceso de desarrollo del sistema para gestionar estaciones Nova – Xerberos

Para definir el proceso de desarrollo del sistema para gestionar estaciones Nova - Xerberos se tuvieron en cuenta todos los aspectos que se mencionan en el acápite anterior pues constituyen la base para desarrollar un producto con las características operativas necesarias.

Es válido señalar que no existe un proceso definido para el desarrollo de este tipo de software, por tanto, el proceso que se define en esta guía es teórico.

La siguiente gráfica resume los procesos a realizar durante el ciclo de vida del software y la vinculación entre cada uno de ellos.

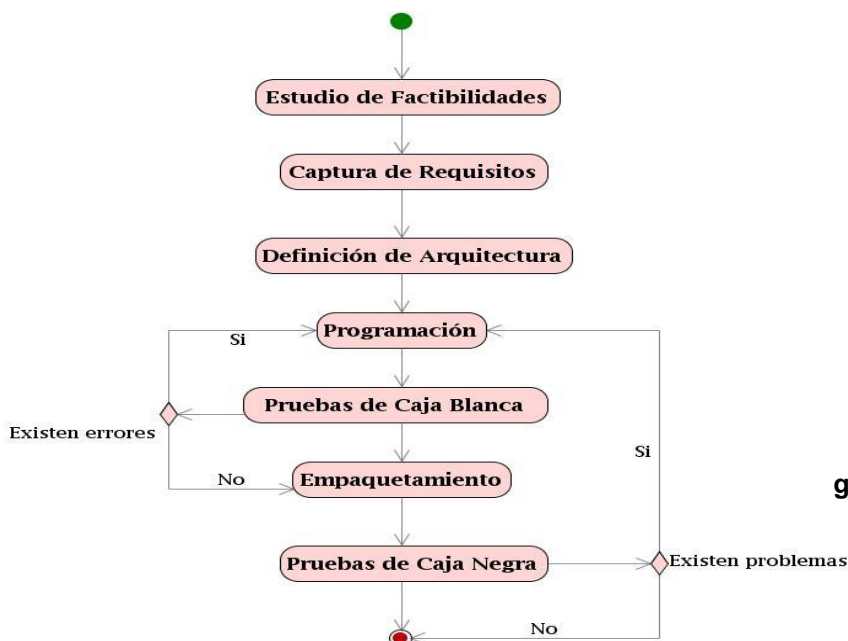


Figura 7. PDS del sistema para gestionar estaciones Nova – Xerberos

1.10 Metodologías de desarrollo de software

En un proyecto de desarrollo de software la metodología define quién debe hacer qué, Cuándo y cómo debe hacerlo. Una metodología es un proceso. No existe una metodología de software universal. Las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigen que el proceso sea configurable. Debido a esto, en la actualidad existen varias clasificaciones para las mismas, entre las que se pueden encontrar a las Metodologías Estructuradas, Orientadas a Objetos, Tradicionales o Pesadas y Ágiles. [4]

Estas últimas están más orientadas a la generación de código con ciclos rápidos, se dirigen a equipos de desarrollo pequeños donde los clientes y desarrolladores trabajan continuamente en constante comunicación. Además el modelo de desarrollo ágil es fácil de aprender, bien documentado y adaptable. A continuación se revisan brevemente algunas de las metodologías.

A continuación se hace un resumen de las metodologías a las que se hace referencia:

1.10.1 SXP

Desde septiembre del año 2006 un equipo de ingenieros de la UCI elaboró una metodología ágil compuesta por sus referentes internacionales SCRUM y XP, denominada SXP [28].

SCRUM es una forma de gestionar un equipo de manera que trabaje de forma eficiente y de tener siempre medidos los progresos. La metódica del trabajo y la necesidad de obtener una versión funcional después de cada iteración, ayuda a la obtención de un software de calidad superior.

XP más bien es una metodología encaminada para el desarrollo; consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

SXP consta de 4 fases principales:

- **Planificación-Definición:** donde se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto.
- **Desarrollo:** es donde se realiza la implementación del sistema hasta que esté listo para ser entregado.
- **Entrega:** puesta en marcha del sistema.
- **Mantenimiento:** es donde se realiza el soporte para el cliente.

En cada una de estas fases se realizan numerosas actividades tales como el levantamiento de requisitos, la priorización de la lista de reserva del producto (LRP), definición de las historias de usuario (HU), diseño, implementación, pruebas, entre otras; de donde se generan artefactos para documentar todo el proceso. Las entregas son frecuentes, y existe una refactorización continua, lo que permite mejorar el diseño cada vez que se le añade una nueva funcionalidad. La garantía de calidad, es una parte esencial en el proceso de SXP. La realización

de pruebas y la publicación de los resultados debe ser lo más rápido posible, para que los desarrolladores puedan realizar con la mayor rapidez los cambios que sean necesarios. A las pruebas de aceptación también se les conoce con el nombre de pruebas de funcionalidad, y constituyen la garantía de que los requerimientos fijados por los usuarios han sido reflejados en el sistema.

SXP está especialmente indicada para proyectos de pequeños equipos de trabajo, rápido cambio de requisitos o requisitos imprecisos, muy cambiantes, donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad. Ayuda a que trabajen todos juntos, en la misma dirección, con un objetivo claro, permitiendo además seguir de forma clara el avance de las tareas a realizar, de forma que los jefes pueden ver día a día cómo progresa el trabajo.

En esta metodología el aseguramiento de la calidad se puede evidenciar en toda la serie de actividades que se realizan con el fin de desarrollar software de la mejor manera posible. Cada una de las fases tiene elementos que pueden brindarle al equipo de desarrollo y al software un nivel de calidad realmente notable, a continuación se especifican dichas elementos:

- La metodología orienta la realización de una planificación detallada que debe ser seguida en cada iteración. Esta planificación debe plasmar alcances, estimaciones, prioridades, realización de reuniones, fechas de entrega de versiones del sistema (entregas pequeñas) y consecuencias de desviaciones.
- En la metodología se define con exactitud la realización del cronograma de producción, las tareas de ingeniería, lo referente a cómo realizar la implementación, la realización de juntas de seguimiento a las actividades donde se define: ¿Qué se hizo? ¿Qué se hará? Y ¿Qué obstaculiza el desarrollo?
- La metodología orienta el control a la consistencia de los artefactos generados.
- La lista de reserva del producto puede ser modificada (siempre que se esté al final de una iteración o sprint), esto, al tener la metodología un desarrollo en espiral, fortalece la mejora continua de los procesos.
- Se tiene en cuenta en todo momento (desde el principio hasta el final del proceso de desarrollo) la valoración de los riesgos, apoyando esto a la gestión de la calidad total del software y los mismos son debidamente orientados a documentar en la plantilla Lista de Riesgos.
- Se orienta a que la programación de las HU sea en parejas pues así se evitaría que al no estar disponible algún programador, su compañero pueda asumir las tareas y no ocurran desviaciones en el proceso por ese motivo, esto traería como ventaja que la tasa de errores del producto final sea baja, el tamaño del código menor y más óptimos los diseños. Otra de las ventajas es que se realiza el trabajo de una persona en casi la mitad del tiempo

y mejor.

- La metodología aboga por la refactorización del código que es una actividad constante de reestructuración del código con el objetivo de remover duplicación del mismo, mejorar su legibilidad, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios.

El proyecto que es motivo de ésta investigación utiliza la metodología *SXP* debido al reducido número de integrantes de su equipo de trabajo y las competencias que poseen sus miembros según las experiencias personales, y, por otra parte, porque es adaptable en cuanto a los roles que define dicha metodología. Desde su creación la han ido aplicando y ya han adquirido cierta práctica en el trabajo con la misma pero de manera coyuntural aluden a la metodología *RUP* debido a que *SXP* no cuenta con parámetros establecidos para el Aseguramiento de la Calidad y por su amplio conocimiento tanto por estudiantes como profesores y especialistas, específicamente se apoyan de los flujos de desarrollo en el de *Pruebas* y en los de soporte: el *Ambiente*, la *Administración de Proyecto* y el de la *Administración de la Configuración y Cambios*.

1.10.2 Proceso Unificado de Rational (RUP)

RUP es un proceso de ingeniería de software planteado por Kruchten (1996) cuyo objetivo es producir software de alta calidad, es decir, que cumpla con los requerimientos de los usuarios dentro de una planificación y presupuesto establecido [30]. Esta metodología abarca todo el ciclo de vida y desarrollo de software.

El Proceso Unificado es un proceso de desarrollo de software cuyo ciclo de vida se caracteriza por ser dirigido por casos de uso, centrado en arquitectura, iterativo e incremental. Consta de cuatro fases o etapas:

- **Comienzo o Inicio:** Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema.
- **Elaboración:** Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen. A pesar de que se desarrolla a profundidad una parte del sistema, las decisiones sobre la arquitectura se hacen sobre la base de la comprensión del sistema completo y los requerimientos (funcionales y no funcionales) identificados de acuerdo al alcance definido.
- **Construcción:** Se basa en la elaboración de un producto totalmente operativo y en la elaboración del manual de usuario. Construir el producto, la arquitectura y los planes, hasta que el producto está listo para ser enviado a la comunidad de usuarios. En esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.
- **Transición:** Se realiza la instalación del producto en el cliente y se procede al entrenamiento de los usuarios. Puede implicar reparación de errores.

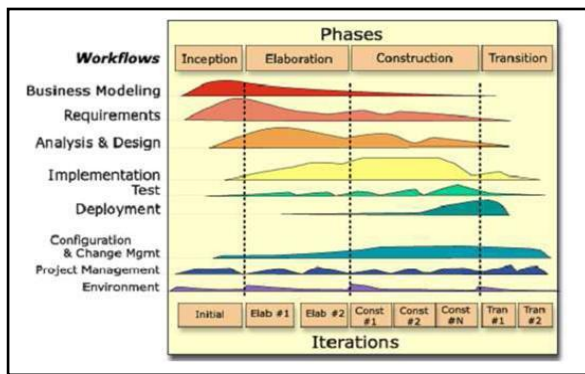


Figura 8. Fases de la metodología RUP

Además de las fases, el ciclo de vida contiene flujos de trabajos, los cuales se dividen en flujos de trabajo de desarrollo y flujos de trabajo de soporte.

Flujos de trabajo de desarrollo:

- **Modelamiento del negocio:** Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- **Requerimientos:** Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- **Análisis y diseño:** Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- **Implementación:** Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- **Prueba (Testeo):** Busca los defectos a lo largo del ciclo de vida.
- **Despliegue:** Se basa en la configuración de hardware y software necesaria, para generar un ambiente adecuado en donde el sistema se alberga.

Flujos de trabajo de soporte:

- **Administración del proyecto:** Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- **Administración de configuración y cambios:** Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.
- **Ambiente:** Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

De los 6 Flujos de trabajo de desarrollo el encargado de evaluar la *calidad del producto* es el de Prueba, esta evaluación no se realiza para aceptar o rechazar el producto al final del proceso de desarrollo, sino que debe ir integrado en todo el ciclo de vida. Además durante las pruebas se llevan a cabo otras actividades como:

- Encontrar y documentar defectos en la calidad del software.
- Generalmente asesora sobre la calidad del software percibida.
- Provee la validación de los artefactos realizados en el diseño y especificación de requisitos por medio de demostraciones concretas.
- Verificar las funciones del producto de software según lo diseñado.
- Verificar que los requisitos tengan su apropiada implementación.

1.11 Proceso de Mejoras

La Universidad de las Ciencias Informáticas (UCI) es un centro productivo, cuya misión es producir software y servicios informáticos a partir de la vinculación estudio – trabajo como modelo de formación. Es considerada la mayor organización productora de software en el país [31].

En la actualidad el centro está acometiendo un proyecto de mejora de sus procesos basado en el modelo CMMI (Capability Maturity Model Integration) y con la contratación de los servicios de consultoría del SIE Center (Software Industry Excellence Center) del Tecnológico de Monterrey.

El servicio de consultoría que ofrece el SIE Center permite:

- Ayudar la UCI a revisar su estrategia de mejora de procesos de software, para asegurar que su organización está basada en procesos y con un programa de mejora continua alineado con sus objetivos de negocio.
- Ayudar la UCI a establecer las bases y fundamentos para seguir mejorando sus procesos y fortalecer su cultura de calidad en el desarrollo de software.
- Alinear los procesos de desarrollo de software con los principios y requisitos del modelo CMMI, estableciendo planes de mejora con los que la organización oriente sus procesos hacia la consecución de sus metas.

El proceso de mejora está encaminado a que la Universidad alcance una certificación internacional del nivel 2 del modelo CMMI. Hecho que la convertiría en la primera empresa cubana certificada con este modelo.

El proyecto Nova – Xerberos está implementando, en todos sus procesos, este plan de mejoras debido a que toda la institución (la UCI) se encuentra desarrollando este plan en su versión 3.3 y por las ventajas que implica para el equipo de trabajo tenerlo como paradigma, ya en iteraciones anteriores del producto (en fase de desarrollo aún) se desplegó este plan en su versión 2.2.

Áreas de proceso del Proceso de Mejora

El proceso de mejora se ideó con el objetivo de establecer las políticas organizacionales y constituir una guía genérica para los proyectos productivos de la UCI. Para ello se dividió en las posibles áreas de proceso o subprocesos que pudieran desarrollar cualquier proyecto, definiéndose las mismas de la siguiente manera:

- ❖ **REQM:** Ésta área se encarga de la Administración de Requisitos.
- ❖ **PP:** Ésta área se encarga de la Planeación de Proyecto.
- ❖ **PMC:** Ésta área se encarga del Monitoreo y Control del Proyecto.
- ❖ **PPQA:** Ésta área se encarga del Aseguramiento de la Calidad a Proceso y Producto.
- ❖ **CM:** Ésta área se encarga de la Administración de la Configuración.
- ❖ **MA:** Ésta área se encarga de la Medición y Análisis.
- ❖ **SAM:** Ésta área se encarga de la Administración de Acuerdo con Proveedores.

De éstas áreas las que atañen al Aseguramiento de la Calidad son *PPQA*, *CM* y *MA*. Para encontrar toda la información referente a las políticas es necesario recurrir al documento **0523_Políticas.pdf**, perteneciente a la documentación del citado proceso.

Los **artefactos** de éste programa (obtenidos como resultado de las áreas *PPQA*, *CM* y *MA*) que se definen para el aseguramiento de la calidad son:

- 0306_Plantilla de Solicitud de Mejora de Proceso o Producto
- 0312_Plantilla de Registro de Evaluaciones del proyecto
- 0314_Plantilla de Notificación de Escalabilidad
- 5107_Glosario de Términos
- 0316_Solicitud de Cambios para Mejora
- 0317_Solicitud de Cambios para Error
- 0330_Lista de Verificación para Auditorías a la Configuración
- 0320_Plantilla de Recolección Manual de Medidas
- 0321_Lista de Verificación de la Integridad de los Datos
- 0322_Plantilla de Reporte de las causas de la no realización del análisis

1.12 Resumen del Capítulo

En el siguiente mapa conceptual se hace un resumen de las principales temáticas abordadas en todo el capítulo, se evidencia la estrecha relación que existe entre cada una de las definiciones brindadas y lo que se necesita para lograr administrar la calidad en proyectos de sistemas para gestionar estaciones.

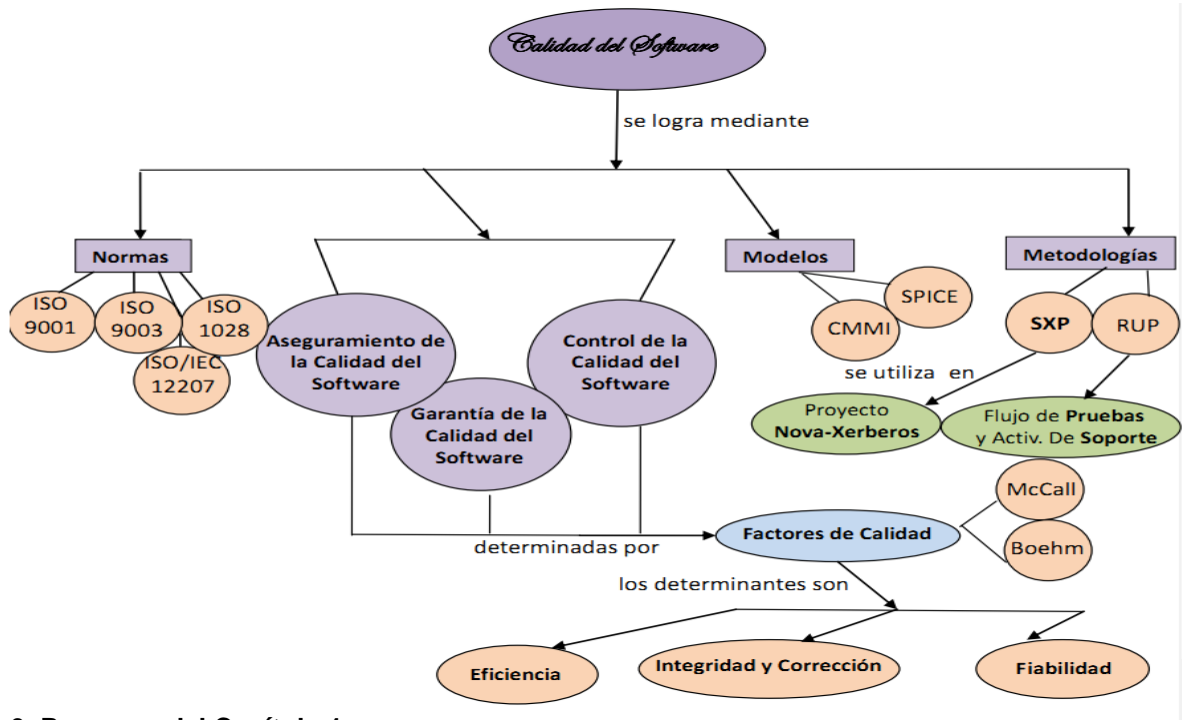


Figura 9. Resumen del Capítulo 1

Capítulo #2. Guía para asegurar la calidad en el sistema de gestión de estaciones Nova - Xerberos

En el presente capítulo se presenta una guía para gestionar la calidad en el desarrollo del software para administrar estaciones Nova - Xerberos. Para esto se lleva a cabo un análisis de lo investigado en el capítulo anterior teniendo como resultado un conjunto de buenas prácticas a tener en cuenta para obtener el producto Nova - Xerberos.

2.1 Introducción de la Guía

2.1.1 Presentación

La presente guía constituye un documento donde se hace un compendio de las principales acciones a tener en cuenta para utilizar las mejores prácticas a la hora de gestionar la calidad. Se crea con el objetivo de asegurar la calidad en todo el proceso de desarrollo de Nova - Xerberos y de esta manera obtener un alto nivel de productividad y mejores vías para garantizar la satisfacción del o los usuarios finales. A continuación se detalla gráficamente cómo está estructurada la guía.

2.1.2 Estructura

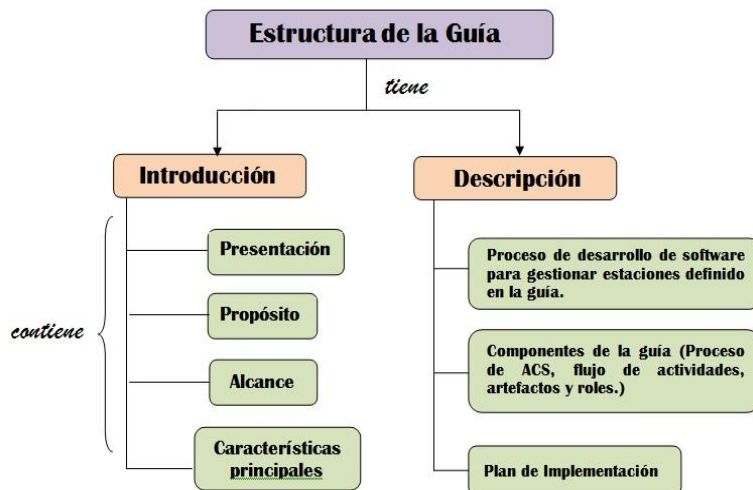


Figura 10. Estructura de la guía

2.1.3 Propósito

La guía tiene como propósito orientar a una mejor administración de la calidad de software durante el desarrollo del sistema para gestionar estaciones Nova - Xerberos.

2.1.4 Alcance

La guía puede ser utilizada en el proceso de desarrollo de cualquier software que tenga como finalidad gestionar estaciones, para ello el software debe contemplar subprocesos al menos similares a los que se definen en la misma.

2.1.5 Características principales

La guía se obtuvo luego de un encumbrado estudio de normas, modelos y estándares de calidad tanto internacionales como nacionales, así como metodologías que se utilizan en el desarrollo de software para determinar cómo se llevan a cabo los procesos de gestión de la calidad en sistemas que brinden la posibilidad de administrar estaciones de forma centralizada.

2.2 Descripción de la guía

La guía que se propone, basándose en el desarrollo evolutivo de la metodología SXP, intenta servir de paradigma en la gestión del proceso de desarrollo del sistema para gestionar estaciones Nova - Xerberos, conteniendo el flujo de actividades a seguir en cada momento así como los roles y los artefactos que intervienen durante todo el ciclo de vida del software. La guía también indica cómo desarrollar las tareas de ACS que se definen en el programa de mejora que está implementando la UCI como institución productora de software, la manera de evaluar las actividades de ACS en correspondencia con la metodología SXP y cómo implementar acciones correctivas en función del seguimiento y la mejora continua de los errores detectados.

2.2.1 Proceso de desarrollo definido en la guía

En el acápite 1.9.1 se definió cómo está estructurado el PDS del proyecto productivo que es motivo de esta investigación. Cada uno de los subprocesos que ocurren durante el proceso de desarrollo del software tiene un peso fundamental, es por ello que se hace el siguiente resumen de las misiones de cada subproceso por separado para su mejor entendimiento.

Estudio de factibilidades: es en este subproceso donde se proporcionan directrices prácticas para la planificación de las actividades, donde se realiza la selección y capacitación del personal, se crea la infraestructura para gestionar los posibles riesgos tanto amenazas como oportunidades y donde se hace un estudio del estado del arte para determinar posibles patrones a seguir en el desarrollo y cuáles serán las herramientas y tecnologías a utilizar. En esta fase también se realiza la asignación de responsabilidades.

Captura de requisitos: es en este subproceso donde se establecen y mantienen acuerdos entre

el equipo de trabajo y los interesados (clientes) sobre lo que el sistema debe hacer y donde se define cómo será la interfaz del sistema enfocada en las necesidades y especificaciones de éstos. También en esta fase es donde se provee a los desarrolladores de las características y condiciones que debe cumplir el software para dar paso a la implementación por parte de ellos, se crean las listas de reserva del producto y las historias de usuario. En este subproceso es necesario seguir las siguientes pautas:

- Verificar que los requisitos estén redactados en infinitivo (terminen en ar, er o ir).
- Verificar que cada requisito sea comprensible para todo tipo de lectores (clientes, usuarios, desarrolladores, equipo de pruebas, gestores, etc.)
- Verificar que cada requisito sea posible de probar, conciso (fácil de leer y entender) y consistente (no es contradictorio con otro requisito).
- Verificar que la especificación de los requisitos no sea ambigua (esto no ocurre si todo requisito posee una sola interpretación).
- Verificar que sea un requisito realizable o implementable en dependencia de los recursos disponibles.
- Realizar la gestión de los requisitos independientemente del diseño.
- Verificar que existe la matriz de trazabilidad entre los requisitos desde el punto de vista jerárquico.
- Verificar que exista trazabilidad entre requisitos y componentes del diseño (que sea posible precisar qué requisitos son implementados por qué componentes del diseño).
- Verificar que si se modifica algún requisito no se afecte demasiado la concepción general del sistema.
- Verificar que los requisitos se encuentren electrónicamente almacenados para su control de versiones del documento de requisitos, puede ser en un archivo de texto, en una base de datos o con una herramienta de gestión de requisitos

Definición de Arquitectura: es en este subproceso donde el equipo de trabajo define la arquitectura candidata para dar solución a lo pactado con el cliente y en función de cómo será posteriormente el proceso de desarrollo. La arquitectura basada en componentes es una aproximación del desarrollo de software que describe, construye y utiliza técnicas de software para la elaboración de sistemas abiertos y distribuidos mediante el ensamblaje de partes del código reutilizables. Un componente puede presentarse en forma de código fuente o código objeto; puede estar escrito en un lenguaje funcional, procedural o en un lenguaje orientado a objetos; y puede ser tan simple como un botón de la interfaz o tan complejo como un subsistema. La característica principal de este tipo de arquitectura es la imbricación que puede tener un componente con otro como se muestra en la figura 11.

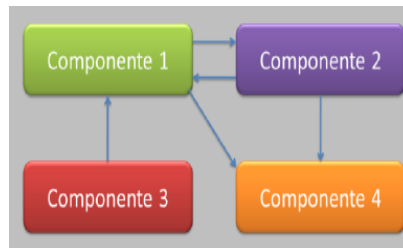


Figura 11. Comportamiento de la arquitectura basada en componentes.

En Nova – Xerberos se utiliza la arquitectura basada en componentes pues los plug-ins o add-ons, como también se les conoce, son aplicaciones que se relacionan con otras aplicaciones para aportarle funcionalidades nuevas al software, muy específicas y que pueden ser añadidas, incluso, de forma dinámica (con el software ejecutándose) y tienen la ventaja de que se pueden ir creando a medida que van surgiendo necesidades durante la implementación del sistema.

Los principios fundamentales cuando se diseña un componente es que estos deben ser:

- **Sin contexto específico** pues son diseñados para operar en diferentes ambientes y contextos. Información específica como el estado de los datos deben ser pasadas al componente en vez de incluirlos o permitir al componente acceder a ellos.
- **Extensibles** ya que un componente puede ser extendido desde un componente existente para crear un nuevo comportamiento.
- **Encapsulado** porque los componentes exponen interfaces que permiten al programa usar su funcionalidad sin revelar detalles internos, detalles del proceso o estado.
- **Independientes** pues los componentes están diseñados para tener una dependencia mínima con otros. Por lo tanto los componentes pueden ser instalados en el ambiente adecuado sin afectar otros componentes o sistemas.
- **Reusables** pues son usualmente diseñados para ser utilizados en disímiles escenarios por diferentes aplicaciones, sin embargo, algunos componentes pueden ser diseñados para tareas específicas.

Además por su ventaja en cuanto a reutilización de los componentes es que fue escogida por el proyecto este tipo de arquitectura.

Programación: es en este subproceso (luego de haber realizado los subprocesos anteriores pertenecientes a la fase de Planificación-Definición de la metodología y luego de haber realizado las especificaciones de las HU y el modelado de todo el diseño) donde se comienza a implementar el software. Para lograr este objetivo el equipo de desarrollo utiliza como lenguaje de programación C++ (lenguaje principal porque además se utilizan Bash y Perl), utilizando el estándar definido por la ISO/IEC 14882 de 1998 y como entorno de desarrollo integrado (IDE, por sus siglas en inglés) el QtCreator. Se utiliza C++ por la cantidad de librerías que posee y por la eficiencia del código que se produce. También es usado C++ porque es el lenguaje de programación más usado para crear software de sistemas y porque para el trabajo con los sistemas de ficheros y archivos de datos como lo hace Nova – Xerberos se hacen necesarias las construcciones de este lenguaje a bajo nivel (aunque es factible también porque dispone de

tipificado a medio y alto nivel).

Pruebas de Caja Blanca: este subproceso se realiza para evaluar y verificar detalles procedimentales del software, detalles en la estructura y correctitud del código fuente. Para ello se escogen distintos valores de entrada y se compila el código para así ir viendo los posibles flujos de ejecución del programa y observar si se van obteniendo los valores de salida adecuados en función de las características de cada módulo. Es en esta etapa donde se comprueban los flujos de ejecución para cada unidad: funcionalidades, clases y módulos.

Empaquetamiento: este subproceso es fruto de la obtención de programas ejecutables, ficheros, bibliotecas y que es necesario conformar un paquete que los contenga. El empaquetado de software garantiza organización y evitar problemas con las dependencias (archivos) pues con el paquete se representa, a los ojos del cliente, el software como un todo único en un archivo ejecutable cuando en realidad están contenidos en él varios ficheros.

Pruebas de Caja Negra: es en este subproceso donde se valida, se verifica y se constata que el software (desde el punto de vista externo) cumple con las especificaciones (requisitos) del cliente. Este tipo de pruebas se realizan trabajando sobre la interfaz principal del software y mediante listas de chequeo que indican qué datos entrarle al sistema para chequear que las salidas sean las correctas pues a diferencia de las pruebas de caja blanca, las de caja negra se enfocan en el exterior del software (interfaz gráfica y sus funcionalidades) y no en la revisión del código fuente. Se trata de la clásica pregunta: ¿Qué es lo que hace? y no ¿Cómo lo hace?

2.2.2 Componentes de la guía

En el siguiente mapa de procesos se puede constatar cuáles son las ACS más significativas y el seguimiento que deben tener durante cada una de las fases con las que cuenta la metodología de desarrollo.

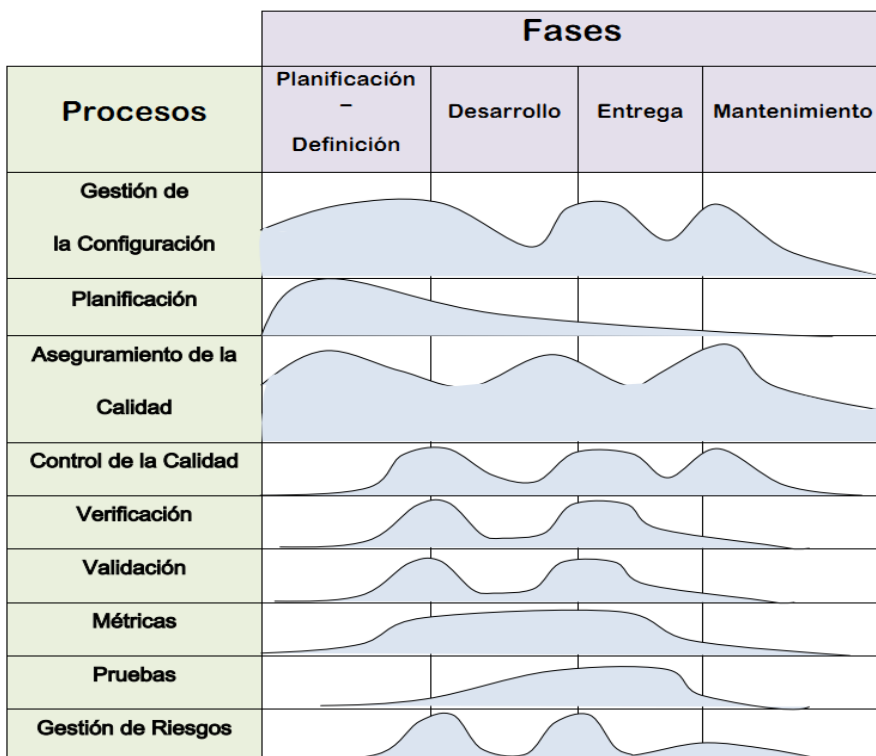


Figura 12. Mapa de procesos de ACS para la metodología de desarrollo SXP

Los componentes de la guía fueron definidos según lo planteado en los acápites 1.9.1 donde se hizo un estudio del proceso de desarrollo del sistema para gestionar estaciones Nova – Xerberos y el 2.2.1 donde se detallaron las principales actividades (por cada subproceso) que ocurren durante el ciclo de vida de este software. De esta forma se pudo decidir cuáles son las actividades más importantes de ACS que se acoplan con la metodología SXP, todas ellas producto de los procesos mencionados en los citados acápites.

Las tablas que se muestran a continuación precisan los parámetros para gestionar la calidad en función del proceso a seguir; con sus entradas, salidas y las actividades que componen el proceso, los roles y artefactos involucrados, los criterios de evaluación, las metas y cómo evaluar los resultados obtenidos (producto).

2.2.2.1 Procesos de ACS aplicados al Personal

Personal	
Nombre del proceso	Verificación de la formación individual de los miembros.
Meta	Verificar que se capacite al personal involucrado en el proyecto.
Actividades que componen el proceso	<ul style="list-style-type: none"> - Comprobación del nivel de conocimiento y motivación de las personas con que se cuenta para el diagnóstico de las competencias específicas y en función de esto trabajar en la asignación de tareas. - Comprobación de la gestión de cursos de capacitación y conferencias para asegurar que cada persona involucrada en el proceso esté lo suficientemente capacitada para desempeñar su rol. - Comprobación de los métodos de formación en función de la entrada de personal al proyecto. - Verificar que estén bien definidas las competencias por roles según lo planteado en el programa de mejora.
Criterios de evaluación	<p>Los criterios de evaluación los pueden proporcionar las siguientes preguntas:</p> <ul style="list-style-type: none"> • ¿Está cada persona capacitada para el rol que desempeña en el proyecto? Nivel de importancia (50%). • ¿Se capacita al personal del proyecto? Nivel de importancia (50%).
Roles	<ul style="list-style-type: none"> - Líder del proyecto. - Administrador de la calidad.
Artefactos de Entrada	<p>Según lo establecido en el Programa de Mejoras se tiene como entrada el artefacto:</p> <ul style="list-style-type: none"> - 0206_Plan de Desarrollo de Software. <p>El proyecto Nova – Xerberos al utilizar la metodología SXP tiene los artefactos de entrada siguientes:</p> <ul style="list-style-type: none"> - Plan de capacitación. - Plan de Aseguramiento de la calidad de software (NovaXerberos_CAL1_P-D_1.1.pdf)
Artefactos de Salida	<p>Según lo establecido en el Programa de Mejoras se tiene como salida el artefacto:</p> <ul style="list-style-type: none"> - 0206_Plan de Desarrollo de Software. <p>El proyecto Nova – Xerberos al utilizar la metodología SXP tiene los artefactos de salida siguientes:</p>

	<ul style="list-style-type: none"> - Plan de capacitación. - Plan de Aseguramiento de la calidad de software (NovaXerberos_CAL1_P-D_1.1.pdf)
Posible acción correctiva	Si no existe el acápite del Plan de capacitación del proyecto en el Plan de Desarrollo de Software (Plantilla: 0206_Plan de Desarrollo de Software), ni el Plan de capacitación y el Plan de Aseguramiento de la calidad de software (NovaXerberos_CAL1_P-D_1.1.pdf) por la metodología SXP o en caso contrario, si necesitan alguna modificación, como medida de acción correctiva indicar realizar la confección de los mismos en un tiempo prudencial y acorde al cronograma del proyecto y posteriormente realizar seguimiento y control a los errores detectados.

2.2.2.2 Procesos de ACS aplicadas a los elementos de configuración del proyecto

Elementos de configuración del proyecto	
Nombre del proceso	Auditoría de la gestión de la configuración en el proyecto.
Meta	<ul style="list-style-type: none"> - Comprobar si los cambios se han hecho de acuerdo a las políticas definidas por el plan de mejora en cuanto a la gestión de la configuración de software. - Comprobar si los cambios se han documentado debidamente.
Actividades que componen el proceso	<ul style="list-style-type: none"> - Auditorías a los elementos de configuración de software (ECS). - Revisión de los cambios realizados y el procedimiento para éstos.
Criterios de evaluación	<p>Los criterios de evaluación los pueden proporcionar las siguientes preguntas:</p> <ul style="list-style-type: none"> • ¿Se ha realizado una revisión técnica formal para comprobar la corrección de los ECS? Nivel de importancia (15%). • ¿Se han seguido adecuadamente los estándares de configuración del software definidos en el programa de mejora? (Plantilla: 5602_Estándares de Configuración.pdf) Nivel de importancia (20%) • ¿Los elementos de configuración están ubicados en la carpeta del repositorio establecida para ello? Nivel de importancia (15%) • ¿Los nombres de los ECS cumplen con la nomenclatura establecida en el Estándar de Configuración? Nivel de importancia (20%) • ¿Se han actualizado adecuadamente todos los ECS relacionados? Nivel de importancia (15%) • En caso de que se hayan incorporado modificaciones adicionales: ¿Se ha hecho el cambio especificado según lo definido en el proceso de mejora? (Plantilla: 0316_Solicitud de Cambios para Mejora o 0317_Solicitud de Cambios para Error) Nivel de importancia (15%)
Roles	<ul style="list-style-type: none"> - Líder del proyecto - Administrador de la calidad - Gestor de la configuración de software (por la metodología SXP sería el gerente.)
Artefactos de Entrada	Según lo establecido en el Programa de Mejoras se tienen como entrada los artefactos: <ul style="list-style-type: none"> - 0206_Plan de Desarrollo de Software.

	<ul style="list-style-type: none"> - 0209_Registro de Revisiones para el Compromiso al Plan. - 0316_Solicitud de Cambios para Mejora o 0317_Solicitud de Cambios para Error. - Repositorios. <p>El proyecto Nova – Xerberos al utilizar la metodología SXP tiene los artefactos de entrada siguientes:</p> <ul style="list-style-type: none"> - Plantilla de gestión de cambios. - Plan de Aseguramiento de la calidad de software (NovaXerberos_CAL1_P-D_1.1.pdf) - Plan de gestión de la configuración (NovaXerberos_GP2_P-D_1.1.pdf).
Artefactos de Salida	<p>Según lo establecido en el Programa de Mejoras se tienen como salida los artefactos:</p> <ul style="list-style-type: none"> - 0206_Plan de Desarrollo de Software. - 0209_Registro de Revisiones para el Compromiso al Plan. - 0316_Solicitud de Cambios para Mejora o 0317_Solicitud de Cambios para Error. - Repositorios. <p>El proyecto Nova – Xerberos al utilizar la metodología SXP tiene los artefactos de salida siguientes:</p> <ul style="list-style-type: none"> - Plantilla de gestión de cambios. - Plan de Aseguramiento de la calidad de software (NovaXerberos_CAL1_P-D_1.1.pdf) - Plan de gestión de la configuración (NovaXerberos_GP2_P-D_1.1.pdf).
Posible acción correctiva	<ul style="list-style-type: none"> - Si no están confeccionados los documentos: 0206_Plan de Desarrollo de Software, 0316_Solicitud de Cambios para Mejora o 0317_Solicitud de Cambios para Error, 0209_Registro de Revisiones para el Compromiso al Plan, Plantilla Lista de Riesgos, Plantilla de gestión de cambios, Plan de Aseguramiento de la calidad de software, Plan de gestión de la configuración, o en caso contrario, si necesitan alguna modificación, como medida de acción correctiva indicar realizar la confección de los mismos en un tiempo prudencial y acorde al cronograma del proyecto y posteriormente realizar seguimiento y control a los errores detectados. - Plasmar (en caso de que existan) las No Conformidades en la plantilla 0312_Plantilla de Registro de Evaluaciones del proyecto.

2.2.2.3 Procesos de ACS aplicadas al proceso de desarrollo

Verificación de requisitos	
Nombre del proceso	Verificación de la correctitud del levantamiento de requisitos.
Meta	Verificar que el levantamiento de requisitos se haya llevado a cabo según los estándares definidos en el programa de mejora (Plantilla: 0102_Lista de chequeo para detectar Inconsistencias en REQM)
Actividades que componen el proceso	<ul style="list-style-type: none"> - Verificar que el levantamiento de requisitos se lleve a cabo de acuerdo con las políticas definidas en el programa de mejora. (Plantilla: IPP-3510_2009 Libro de Proceso para la Administración de Requisitos.pdf). - Verificar que exista la descripción de los requisitos del cliente y los que haya considerado el proyecto, así como la Matriz de Trazabilidad entre los requisitos. (Plantilla: 0105_Reporte de Trazabilidad).
Criterios de evaluación	Los criterios de evaluación los pueden proporcionar las siguientes preguntas:

	<ul style="list-style-type: none"> • ¿Se encuentran definidas las necesidades de los involucrados y usuarios? Nivel de importancia (15%) • ¿Están identificados los proveedores que dan origen a los requisitos? Nivel de importancia (15%) • ¿Están especificados los requisitos del cliente? Nivel de importancia (20%) • ¿Se encuentra actualizada la Especificación de los Requisitos de Software (ERS) a partir de la lista de requisitos válidos? Nivel de importancia (20%) • ¿Se encuentran establecidas las relaciones de los requisitos con las Historias de Usuario (HU)? Nivel de importancia (15%) • ¿Se encuentra establecida la trazabilidad entre los requisitos y las HU? Nivel de importancia (15%)
Roles	<ul style="list-style-type: none"> - Líder del proyecto. - Administrador de la calidad. - Analista.
Artefactos de Entrada	<p>Según lo establecido en el Programa de Mejoras se tienen como entradas los artefactos:</p> <ul style="list-style-type: none"> - 0129_Descripción de requisitos. - 0119_Criterios para validar requisitos del producto - 0118_Criterios para validar requisitos del cliente - 0103_Criterios para definir proveedores válidos de requisitos - 0105_Reporte de Trazabilidad <p>El proyecto Nova – Xerberos al utilizar la metodología SXP tiene los artefactos de entrada siguientes:</p> <ul style="list-style-type: none"> - Plantilla de Lista de Reserva del producto. - Plantilla de Historias de Usuario (HU).
Artefactos de Salida	<p>Según lo establecido en el Programa de Mejoras se tienen como salidas los artefactos:</p> <ul style="list-style-type: none"> - 0129_Descripción de requisitos. - 0105_Reporte de Trazabilidad. <p>El proyecto Nova – Xerberos al utilizar la metodología SXP tiene los artefactos de salida siguientes:</p> <ul style="list-style-type: none"> - Plantilla de Lista de Reserva del producto. - Plantilla de Historias de Usuario (HU).
Posible acción correctiva	<ul style="list-style-type: none"> - Si no existen las plantillas: 0129_Descripción de requisitos, 0105_Reporte de Trazabilidad y 0132_Evaluación de Requisitos, Plantilla de Lista de Reserva del producto, Plantilla de Historias de Usuario, o en caso contrario, si necesitan alguna modificación, como medida de acción correctiva indicar realizar la confección de los mismos en un tiempo prudencial y acorde al cronograma del proyecto y posteriormente realizar seguimiento y control a los errores detectados. - Plasmar (en caso de que existan) las No Conformidades en la plantilla 0312_Plantilla de Registro de Evaluaciones del proyecto.
Programación	
Nombre del proceso	Verificar la correctitud del subproceso programación.

Meta	<p>Verificar que la programación del software se haya llevado a cabo según los estándares definidos en el programa de mejora para los lenguajes de programación que se utilizan en el proyecto.</p> <p>(Plantilla: 0120_52 Estándares de codificación para C++.doc)</p>
Actividades que componen el proceso	<ul style="list-style-type: none"> - Verificar que la implementación se lleve a cabo de acuerdo con las políticas definidas en el programa de mejora. (Plantilla: 0120_52 Estándares de codificación para C++.doc). - Verificar que el cronograma de producción se lleva a cabo y sin retrasos.
Criterios de evaluación	<p>Los criterios de evaluación los pueden proporcionar las siguientes preguntas:</p> <ul style="list-style-type: none"> • ¿Está el documento acorde con a la plantilla estándar del proyecto o del expediente de proyecto? Nivel de importancia (10%). • ¿Se ha hecho la implementación teniendo en cuenta el estilo arquitectónico definido para la aplicación? Nivel de importancia (10%). • ¿Se han enfocado las clases a las capacidades o condiciones del sistema? Nivel de importancia (5%). • ¿Se han identificado las responsabilidades (atributos y métodos) de las clases? Nivel de importancia (10%). • ¿Se ha verificado que no exista más de una clase con la misma responsabilidad? Nivel de importancia (5%). • ¿Se han determinado las relaciones entre las clases (asociación, agregación, composición, herencia)? Nivel de importancia (5%). • ¿Los atributos definen completamente el objetivo y alcance de la clase? Nivel de importancia (5%). • ¿Se han aplicado patrones para el mejor entendimiento de las clases? (GRASP, GoF) Nivel de importancia (10%). • ¿Existe un correcto estilo de programación tabulado en los códigos? . Nivel de importancia (10%). • ¿Existe un correcto espaciado en los códigos? Nivel de importancia (10%). • ¿Existe una correcta aplicación de bucles y estructuras de control en los códigos? Nivel de importancia (5%). • ¿A cada historia de usuario hay asignada una o varias tareas de ingeniería relacionadas con la codificación de la misma? Nivel de importancia (10%). • ¿Los programadores cumplen las tareas de implementación ajustados en tiempo con su planificación en el cronograma de producción? Nivel de importancia (5%).
Roles	<ul style="list-style-type: none"> - Administrador de Calidad - Programadores.
Artefactos de Entrada	<p>El proyecto Nova – Xerberos al utilizar la metodología SXP tiene los artefactos de entrada siguientes:</p> <ul style="list-style-type: none"> - Códigos fuente. - Plantilla Cronograma de producción.

	- Plantilla Plan de releases.
Artefacto de Salida	El proyecto Nova – Xerberos al utilizar la metodología SXP tiene los artefactos de salida siguientes: - Códigos fuente. - Plantilla Cronograma de producción. - Plantilla Plan de releases.
Posible acción correctiva	- Subir al repositorio los códigos fuente arreglados. - Plasmar (en caso de que existan) las No Conformidades en la plantilla 0312_Plantilla de Registro de Evaluaciones del proyecto. - Si no existen las plantillas: Cronograma de producción o el Plan de releases , o en caso contrario, si necesitan alguna modificación, como medida de acción correctiva indicar realizar la confección de los mismos en un tiempo prudencial y acorde al cronograma del proyecto y posteriormente realizar seguimiento y control a los errores detectados.
Empaquetamiento de Software	
Nombre del proceso	Verificación de la correctitud del empaquetamiento de software.
Meta	Verificar que el empaquetamiento de software se haya llevado a cabo según los estándares definidos internacionalmente (no se define nada relacionado en el programa de mejora) o predefinidos por los desarrolladores (lo define el estándar de codificación del proyecto).
Actividades que componen el proceso	- Verificar que el empaquetamiento de software se haya llevado a cabo según los estándares definidos internacionalmente. Ver Anexo 2 o predefinidos por los desarrolladores (lo define el estándar de codificación del proyecto en este caso el proyecto Matriz es Nova para consultar la forma de empaquetamiento visite http://comunidades.uci.cu/blogs/nova/?p=280). Para conocer las políticas de empaquetamiento de Ubuntu visitar: https://wiki.ubuntu.com/PackagingGuide/es/Complete .
Criterios de evaluación	Los criterios de evaluación los pueden proporcionar las siguientes preguntas: <ul style="list-style-type: none"> • ¿El software alcanza sus requisitos mínimos? Nivel de importancia (25%). • ¿Se puede garantizar que el software se desarrolla de acuerdo a ciertos estándares predefinidos tanto internacionales como predefinidos por los desarrolladores (lo define el estándar de codificación del proyecto)? Nivel de importancia (15%). • ¿Existe optimización del proceso de empaquetamiento? Nivel de importancia (20%). • ¿Los elementos del código están ubicados en la carpeta del repositorio establecida para ello? Nivel de importancia (20%). • ¿Logra el equipo de trabajo conseguir un paquete de manera uniforme? Nivel de importancia (10%). • ¿Una vez que el software es empaquetado e instalado en el sistema operativo funciona correctamente? Nivel de importancia (10%).

Roles	<ul style="list-style-type: none"> - Líder del proyecto. - Administrador de la calidad. - Programadores (pues no existe el rol de mantenedor de paquetes.)
Artefactos de Entrada	<p>Este proceso no depende para su realización de artefactos del PM ni de la metodología SXP, necesitándose:</p> <ul style="list-style-type: none"> - Códigos fuente. - Estándares. - Políticas de empaquetamiento.
Artefacto de Salida	<p>Este proceso no depende para su realización de artefactos del PM ni de la metodología SXP, necesitándose:</p> <ul style="list-style-type: none"> - Códigos fuente. - Estándares. - Políticas de empaquetamiento.
Posible acción correctiva	<ul style="list-style-type: none"> - Plasmar (en caso de que existan) las No Conformidades en la plantilla 0312_Plantilla de Registro de Evaluaciones del proyecto. - Indicar al equipo de trabajo el uso de políticas de empaquetamiento como medida en caso de presentar errores y posteriormente darle seguimiento a los errores detectados.

Pruebas

Nombre del proceso	Verificación de la correctitud del subproceso pruebas.
Meta	- Comprobar que las pruebas (tanto de Caja Blanca como de Caja Negra) realizadas al software se lleven a cabo según el plan de pruebas definido por el proyecto y el mismo esté acorde a lo definido en el programa de mejora.
Actividades que componen el proceso	<ul style="list-style-type: none"> - Verificar que esté creado el ambiente y las condiciones para llevar a cabo el proceso de pruebas. - Revisar si en los escenarios de prueba se tuvieron en cuenta el cumplimiento de los requisitos funcionales y no funcionales. - Verificar que en las pruebas se tuvieron en cuenta los siguientes factores de calidad del software: usabilidad, funcionalidad, corrección, integridad y fiabilidad. - Verificar que se pruebe la interfaz y sus componentes. - Verificar que se elabore un Informe de Pruebas, en el que se registren los resultados del proceso de prueba.
Criterios de evaluación	<ul style="list-style-type: none"> • Verificar que las pruebas se realicen en cada iteración del software. Nivel de importancia (35%) • Comprobar que los defectos se corrijan tan pronto como sean encontrados. Nivel de importancia (15%). • Comprobar que el producto software respeta los criterios o estándares aplicados en el proyecto y cumpla con los factores de calidad requeridos. Nivel de importancia (35%).

	<ul style="list-style-type: none"> • Verificar que se analicen las lecciones aprendidas para futuras liberaciones del proyecto y la mejora continua a la que se aspira para obtener un proceso de desarrollo totalmente gestionado. Nivel de importancia (15%).
Roles	<ul style="list-style-type: none"> - Departamento de pruebas de CALISOFT - Administrador de la calidad
Artefactos de Entrada	<p>Según lo establecido en el Programa de Mejoras se tienen como entradas los artefactos:</p> <ul style="list-style-type: none"> - 5306_Guía para Establecer Puntos de Monitoreo 2.0.pdf - 5307_Guía para Identificar Desviaciones Significativas 2.0.pdf - 5304_Guía para Generar y Asignar Acciones Correctivas.pdf - QA 5203_Guia para procesos y productos a evaluar v3.0.pdf <p>Según lo establecido en el Programa de Mejoras se tienen como entradas los artefactos:</p> <ul style="list-style-type: none"> - 0129_Descripción de requisitos. - 0205_Plan de Pruebas <p>El proyecto Nova – Xerberos al utilizar la metodología SXP tiene los artefactos de entrada siguientes:</p> <ul style="list-style-type: none"> - Plantilla caso de prueba de aceptación. - Casos de Prueba previamente elaborados por los probadores.
Artefactos de Salida	<p>Según lo establecido en el Programa de Mejoras se tienen como salidas los artefactos:</p> <ul style="list-style-type: none"> - 0129_Descripción de requisitos. - 0205_Plan de Pruebas <p>El proyecto Nova – Xerberos al utilizar la metodología SXP tiene los artefactos de salida siguientes:</p> <ul style="list-style-type: none"> - Plantilla caso de prueba de aceptación. - Casos de Prueba previamente elaborados por los probadores.
Posible acción correctiva	<ul style="list-style-type: none"> - Si no están confeccionados los documentos: 0205_Plan de Pruebas, 0129_Descripción de requisitos ni la Plantilla caso de prueba de aceptación, o en caso contrario, si necesitan alguna modificación, como medida de acción correctiva indicar realizar la confección de los mismos en un tiempo prudencial y acorde al cronograma del proyecto y posteriormente realizar seguimiento y control a los errores detectados. - Plasmar (en caso de que existan) las No Conformidades en la plantilla 0312_Plantilla de Registro de Evaluaciones del proyecto.

2.2.2.4 Procesos de ACS aplicadas al producto

Producto	
Nombre del proceso	Control de calidad al producto.
Meta	- Controlar que el producto cumpla con los estándares definidos por el proyecto y en el programa de mejora.

Actividades que componen el proceso	<ul style="list-style-type: none"> - Controlar que los defectos se corrijan tan pronto como sean encontrados. - Comprobar que el producto software respeta los criterios o estándares aplicados en el proyecto y cumpla con los factores de calidad requeridos.
Criterios de evaluación	<p>Los criterios de evaluación los pueden proporcionar las siguientes preguntas:</p> <ul style="list-style-type: none"> • ¿Se ha realizado una revisión técnica formal para comprobar la calidad del producto? Nivel de importancia (30%). • ¿Se ha seguido adecuadamente lo definido en el programa de mejora? Nivel de importancia (30%). • ¿El producto software respeta los criterios o estándares aplicados en el proyecto y cumple con los factores de calidad requeridos? Nivel de importancia (15%). • En caso de que se hayan incorporado modificaciones adicionales al software: ¿Se ha hecho el cambio especificado según lo definido en el proceso de mejora? (Plantilla: 0316_Solicitud de Cambios para Mejora o 0317_Solicitud de Cambios para Error) Nivel de importancia (25%).
Roles	<ul style="list-style-type: none"> - Líder del proyecto - Administrador de la calidad - Departamento de pruebas de CALISOFT.
Artefactos de Entrada	<ul style="list-style-type: none"> - 0129_Descripción de requisitos. - 0205_Plan de Pruebas - 0316_Solicitud de Cambios para Mejora o 0317_Solicitud de Cambios para Error. - 5306_Guía para Establecer Puntos de Monitoreo 2.0.pdf - 5307_Guía para Identificar Desviaciones Significativas 2.0.pdf - 5304_Guía para Generar y Asignar Acciones Correctivas.pdf - QA 5203_Guia para procesos y productos a evaluar v3.0.pdf - Producto. <p>El proyecto Nova – Xerberos al utilizar SXP tiene además otros artefactos de entrada para este proceso:</p> <ul style="list-style-type: none"> - Plantilla de Lista de Reserva del producto. - Plantilla de Historias de Usuario (HU). - Plantilla caso de prueba de aceptación.
Artefactos de Salida	<p>Según lo establecido en el Programa de Mejoras se tienen como salidas los artefactos:</p> <ul style="list-style-type: none"> - 0129_Descripción de requisitos. - 0205_Plan de Pruebas - 0316_Solicitud de Cambios para Mejora o 0317_Solicitud de Cambios para Error. - Producto. <p>El proyecto Nova – Xerberos al utilizar la metodología SXP tiene los artefactos de salida siguientes:</p> <ul style="list-style-type: none"> - Plantilla de Lista de Reserva del producto. - Plantilla de Historias de Usuario (HU). - Plantilla caso de prueba de aceptación.

Possible acción correctiva	<ul style="list-style-type: none"> - Si no están confeccionados los documentos: 0206_Plan de Desarrollo de Software, 0129_Descripción de requisitos, 0205_Plan de Pruebas, 0316_Solicitud de Cambios para Mejora o 0317_Solicitud de Cambios para Error, 0209_Registro de Revisiones para el Compromiso al Plan, Plantilla de Lista de Reserva del producto, Plantilla de Historias de Usuario (HU), Plantilla caso de prueba de aceptación, o en caso contrario, si necesitan alguna modificación, como medida de acción correctiva indicar realizar la confección de los mismos en un tiempo prudencial y acorde al cronograma del proyecto y posteriormente realizar seguimiento y control a los errores detectados. - Plasmar (en caso de que existan) las No Conformidades en la plantilla 0312_Plantilla de Registro de Evaluaciones del proyecto.
Producto	
Nombre del proceso	Validación del producto.
Meta	<ul style="list-style-type: none"> - Validar que el producto cumpla con los estándares definidos por el proyecto, con lo establecido en el programa de mejora y con lo definido por los clientes.
Actividades que componen el proceso	<ul style="list-style-type: none"> - Controlar que los defectos se corrijan tan pronto como sean encontrados. - Comprobar que el producto software respeta los criterios o estándares aplicados en el proyecto y cumpla con los factores de calidad requeridos. - Validar que se realicen pruebas de sistema. - Validar que se realicen pruebas de aceptación.
Criterios de evaluación	<p>Los criterios de evaluación los pueden proporcionar las siguientes preguntas:</p> <p>¿Se ha realizado una revisión técnica formal para comprobar la calidad del producto? Nivel de importancia (15%).</p> <p>¿Se ha seguido adecuadamente lo definido en el programa de mejora? Nivel de importancia (10%).</p> <p>¿El producto software respeta los criterios o estándares aplicados en el proyecto y cumple con los factores de calidad requeridos? Nivel de importancia (15%).</p> <p>¿Se hacen las pruebas requeridas durante el proceso de desarrollo de nova Xerberos? Nivel de importancia (15%)</p> <p>¿Se guardan los diseños de casos de prueba? Nivel de importancia (10%)</p> <p>¿Se guardan las no conformidades detectadas? Nivel de importancia (10%)</p> <p>¿Se le da seguimiento a solución de las no conformidades? Nivel de importancia (10%)</p> <p>En caso de que se hayan incorporado modificaciones adicionales al software: ¿Se ha hecho el cambio especificado según lo definido en el proceso de mejora? (Plantilla: 0316_Solicitud de Cambios para Mejora o 0317_Solicitud de Cambios para Error) Nivel de importancia (15%).</p>
Roles	<ul style="list-style-type: none"> - Líder del proyecto - Administrador de la calidad

Artefactos de Entrada	<p>Según lo establecido en el Programa de Mejoras se tienen como entradas los artefactos:</p> <ul style="list-style-type: none"> - 0129_Descripción de requisitos. - 0205_Plan de Pruebas - 0316_Solicitud de Cambios para Mejora o 0317_Solicitud de Cambios para Error. - 5306_Guía para Establecer Puntos de Monitoreo 2.0.pdf - 5307_Guía para Identificar Desviaciones Significativas 2.0.pdf - 5304_Guía para Generar y Asignar Acciones Correctivas.pdf - QA 5203_Guia para procesos y productos a evaluar v3.0.pdf - Producto. - Pruebas de aceptación por parte del cliente o los usuarios finales. <p>El proyecto Nova – Xerberos al utilizar la metodología SXP tiene los artefactos de entrada siguientes:</p> <ul style="list-style-type: none"> - Plantilla de Lista de Reserva del producto. - Plantilla de Historias de Usuario (HU). - Plantilla caso de prueba de aceptación.
Artefactos de Salida	<p>Según lo establecido en el Programa de Mejoras se tienen como salidas los artefactos:</p> <ul style="list-style-type: none"> - 0129_Descripción de requisitos. - 0205_Plan de Pruebas - 0316_Solicitud de Cambios para Mejora o 0317_Solicitud de Cambios para Error. - Producto. - Pruebas de aceptación por parte del cliente o los usuarios finales. <p>El proyecto Nova – Xerberos al utilizar la metodología SXP tiene los artefactos de salida siguientes:</p> <ul style="list-style-type: none"> - Plantilla de Lista de Reserva del producto. - Plantilla de Historias de Usuario (HU). - Plantilla caso de prueba de aceptación.
Posible acción correctiva	<ul style="list-style-type: none"> - Si no están confeccionados los documentos: 0206_Plan de Desarrollo de Software, 0129_Descripción de requisitos, 0205_Plan de Pruebas, 0316_Solicitud de Cambios para Mejora o 0317_Solicitud de Cambios para Error, 0209_Registro de Revisiones para el Compromiso al Plan, Plantilla de Lista de Reserva del producto, Plantilla de Historias de Usuario (HU), Plantilla caso de prueba de aceptación, o en caso contrario, si necesitan alguna modificación, como medida de acción correctiva indicar realizar la confección de los mismos en un tiempo prudencial y acorde al cronograma del proyecto y posteriormente realizar seguimiento y control a los errores detectados. - Plasmar (en caso de que existan) las No Conformidades en la plantilla 0312_Plantilla de Registro de Evaluaciones del proyecto.

2.2.2.5 Procesos de ACS para factores de calidad críticos en Nova – Xerberos.

Corrección	
Nombre del proceso	Verificación del factor de calidad Corrección.

Meta	- Controlar que el producto cuente con el factor de calidad corrección.
Actividades que componen el proceso	<ul style="list-style-type: none"> - Comprobar que el producto software cuente con completitud, consistencia y trazabilidad. - Controlar que los defectos se corrijan tan pronto como sean encontrados.
Criterios de evaluación	<p>Los criterios de evaluación los pueden proporcionar las siguientes preguntas:</p> <ul style="list-style-type: none"> • ¿Se han definido técnicas y notaciones de diseño e implementación para la completitud del código? Nivel de importancia (25%). • ¿Se ha seguido lo establecido en los artefactos de la Ingeniería de Software: Lista de reserva del producto, Arquitectura del Software y Plantilla Modelo de Diseño? Nivel de importancia (10%). • ¿Se lleva la trazabilidad entre requisitos funcionales y las HU a las que pertenece en la Matriz de Trazabilidad? Nivel de importancia (20%). • ¿Los registros de errores en cuanto a desviaciones en la correctitud de las funcionalidades se tienen en cuenta? ¿Se documentan y se eliminan esos errores detectados? Nivel de importancia (10%). • ¿Se reúne el equipo de trabajo para chequear el desarrollo y completitud del software? Nivel de importancia (10%). • ¿Están implementadas todas las funcionalidades? Nivel de importancia (15%). • ¿Se verifican las funcionalidades y se determina la correctitud de cada una de ellas en cuanto a lo que debe hacer? Nivel de importancia (10%).
Roles	<ul style="list-style-type: none"> - Líder del proyecto. - Administrador de la calidad. - Resto del equipo de trabajo.
Artefactos de Entrada	<p>Según lo establecido en el Programa de Mejoras se tienen como entradas los artefactos:</p> <ul style="list-style-type: none"> - 0129_Descripción de requisitos. - 5306_Guía para Establecer Puntos de Monitoreo 2.0.pdf - 5307_Guía para Identificar Desviaciones Significativas 2.0.pdf - 5304_Guía para Generar y Asignar Acciones Correctivas.pdf - QA 5203_Guia para procesos y productos a evaluar v3.0.pdf - Producto. - Casos de prueba. <p>El proyecto Nova – Xerberos al utilizar la metodología SXP tiene los artefactos de entrada siguientes:</p> <ul style="list-style-type: none"> - Plantilla de Arquitectura del software. - Plantilla de Historias de Usuario (HU). - Plantilla Modelo de Diseño.

Artefactos de Salida	<p>Según lo establecido en el Programa de Mejoras se tienen como salidas los artefactos:</p> <ul style="list-style-type: none"> - 0129_Descripción de requisitos. - Producto. <p>El proyecto Nova – Xerberos al utilizar la metodología SXP tiene los artefactos de salida siguientes:</p> <ul style="list-style-type: none"> - Plantilla de Arquitectura del software. - Plantilla de Historias de Usuario (HU). - Plantilla Modelo de Diseño.
Posible acción correctiva	<ul style="list-style-type: none"> - Si no están confeccionados los documentos: Plantilla de Arquitectura del software, Plantilla de Historias de Usuario (HU), Plantilla Modelo de Diseño, o en caso contrario, si necesitan alguna modificación, como medida de acción correctiva indicar realizar la confección de los mismos en un tiempo prudencial y acorde al cronograma del proyecto y posteriormente realizar seguimiento y control a los errores detectados. - Plasmar (en caso de que existan) las No Conformidades en la plantilla 0312_Plantilla de Registro de Evaluaciones del proyecto.
EFICIENCIA	
Nombre del proceso	Verificación del factor de calidad Eficiencia.
Meta	<ul style="list-style-type: none"> - Controlar que el producto cuente con el factor de calidad eficiencia.
Actividades que componen el proceso	<ul style="list-style-type: none"> - Comprobar que el producto software cuente con eficiencia en ejecución, en almacenamiento y sea eficiente al enviar información entre servidor/clientes. - Controlar que los defectos se corrijan tan pronto como sean encontrados.
Criterios de evaluación	<p>Los criterios de evaluación los pueden proporcionar las siguientes preguntas:</p> <ul style="list-style-type: none"> • ¿Se utilizan herramientas automatizadas para la revisión de códigos y medir su eficiencia? Nivel de importancia (15%). • ¿Se le realizan diariamente compilaciones al producto para verificar su funcionamiento? Nivel de importancia (15%). • ¿Se hace uso por parte de los desarrolladores de técnicas de programación para lograr eficiencia en el software? Por ejemplo: Divide y vencerás, backtracking, chequeo de complejidad temporal y ciclomática, notaciones de complejidad asintótica, uso de estructuras de datos ligeras. Nivel de importancia (20%). • ¿Se cuenta con una base de datos para registrar los bugs y corregirlos antes de adicionar código nuevo? Nivel de importancia (10%). • ¿Se realiza la compresión de los datos generados por el software para evitar utilizar grandes volúmenes de almacenamiento? Nivel de importancia (15%). • ¿Se logra optimizar el ancho de banda para el envío de información entre cliente - servidor? Nivel de importancia (15%).

Roles	<ul style="list-style-type: none"> - Líder del proyecto - Administrador de la calidad - Desarrolladores.
Artefactos de Entrada	<p>Este factor de calidad no depende para su verificación de artefactos del PM ni de la metodología SXP, necesitándose:</p> <ul style="list-style-type: none"> - Producto. - Códigos fuente. - Casos de prueba.
Artefactos de Salida	<p>Este factor de calidad no depende para su verificación de artefactos del PM ni de la metodología SXP, generándose:</p> <ul style="list-style-type: none"> - Producto. - Códigos fuente.
Posible acción correctiva	<ul style="list-style-type: none"> - Si existen problemas o desviaciones (en dependencia con los criterios de evaluación) en los códigos o el producto, o en caso contrario, si necesitan alguna modificación, como medida de acción correctiva indicar realizar la confección de los mismos en un tiempo prudencial y acorde al cronograma del proyecto y posteriormente realizar seguimiento y control a los errores detectados. - Plasmar (en caso de que existan) las No Conformidades en la plantilla 0312_Plantilla de Registro de Evaluaciones del proyecto.

INTEGRIDAD

Nombre del proceso	Verificación del factor de calidad Integridad.
Meta	<ul style="list-style-type: none"> - Controlar que el producto cuente con el factor de calidad Integridad.
Actividades que componen el proceso	<ul style="list-style-type: none"> - Comprobar que el producto software cuente con seguridad, que sea fácil de auditar y que posea un estricto control de accesos. - Controlar que los defectos se corrijan tan pronto como sean encontrados.
Criterios de evaluación	<p>Los criterios de evaluación los pueden proporcionar las siguientes preguntas:</p> <ul style="list-style-type: none"> • ¿Tiene el software implementados mecanismos para detectar ataques, amenazas o riesgos de seguridad? Por ejemplo detección de intrusos, contra ataques de monitorización, de autenticación, de ingeniería inversa, de denegación de servicios o de modificación – daño, inyecciones SQL, por sólo citar algunos. Nivel de importancia (30%). • ¿Se hace uso de protocolos seguros para enviar y recibir datos? Por ejemplo: https, ssh, ssl, tls. Nivel de importancia (25%). • ¿La información y la documentación del software están guardadas en un lugar seguro y confiable? Por ejemplo: repositorios de datos donde el acceso sea por autenticación y sólo usuarios autorizados. Nivel de importancia (10%). • ¿Es fácil auditar el software? ¿Son intuitivos y específicos los nombres de las funcionalidades? Nivel de importancia (15%).

	<ul style="list-style-type: none"> ¿Se verifica el control de accesos al software? Por ejemplo mediante un nivel de acceso por roles, utilizando proveedores de identidades o con un sistema de acceso centralizado. Nivel de importancia (20%).
Roles	<ul style="list-style-type: none"> - Líder del proyecto - Administrador de la calidad - Desarrolladores.
Artefactos de Entrada	<p>Este factor de calidad no depende para su verificación de artefactos del PM ni de la metodología SXP, necesitándose:</p> <ul style="list-style-type: none"> - Producto. - Códigos fuente. - Casos de prueba.
Artefactos de Salida	<p>Este factor de calidad no depende para su verificación de artefactos del PM ni de la metodología SXP, generándose:</p> <ul style="list-style-type: none"> - Producto. - Códigos fuente.
Posible acción correctiva	<ul style="list-style-type: none"> - Si existen problemas o desviaciones (en dependencia con los criterios de evaluación) en los códigos o el producto, o en caso contrario, si necesitan alguna modificación, como medida de acción correctiva indicar realizar la confección de los mismos en un tiempo prudencial y acorde al cronograma del proyecto y posteriormente realizar seguimiento y control a los errores detectados. - Plasmar (en caso de que existan) las No Conformidades en la plantilla 0312_Plantilla de Registro de Evaluaciones del proyecto.

FIABILIDAD

Nombre del proceso	Verificación del factor de calidad Fiabilidad.
Meta	- Controlar que el producto cuente con el factor de calidad fiabilidad.
Actividades que componen el proceso	<ul style="list-style-type: none"> - Comprobar que el producto software cuente con precisión, sea tolerante a fallos, sea modular y posea simplicidad. - Controlar que los defectos se corrijan tan pronto como sean encontrados.
Criterios de evaluación	<p>Los criterios de evaluación los pueden proporcionar las siguientes preguntas:</p> <ul style="list-style-type: none"> • ¿Al introducirle datos de diversas índoles el software responde coherentemente ante ellos? ¿Es tolerante ante los fallos el software? Nivel de importancia (25%). • ¿Es escalable cada funcionalidad del software? De esta forma al introducirle nuevas modificaciones a alguna de ellas no se afectaría todo el sistema. Nivel de importancia (20%). • ¿Se emplea en el desarrollo un enfoque minimalista? Si esto ocurre se haría lo necesario, lo que el software debe hacer y no más de lo requerido. Nivel de importancia (10%). • ¿Son precisos los cálculos que realizan las funcionalidades que deban hacerlos? Nivel de importancia

	<p>(30%).</p> <ul style="list-style-type: none"> • ¿Está estructurado por módulos el sistema? Ello permitiría que al realizar cambios sobre alguno no se afecten los demás. Nivel de importancia (15%).
Roles	<ul style="list-style-type: none"> - Líder del proyecto - Administrador de la calidad - Desarrolladores.
Artefactos de Entrada	<p>Este factor de calidad no depende para su verificación de artefactos del PM ni de la metodología SXP, necesitándose:</p> <ul style="list-style-type: none"> - Producto. - Códigos fuente. - Casos de prueba.
Artefactos de Salida	<p>Este factor de calidad no depende para su verificación de artefactos del PM ni de la metodología SXP, generándose:</p> <ul style="list-style-type: none"> - Producto. - Códigos fuente.
Posible acción correctiva	<ul style="list-style-type: none"> - Si existen problemas o desviaciones (en dependencia con los criterios de evaluación) en los códigos o el producto, o en caso contrario, si necesitan alguna modificación, como medida de acción correctiva indicar realizar la confección de los mismos en un tiempo prudencial y acorde al cronograma del proyecto y posteriormente realizar seguimiento y control a los errores detectados. - Plasmar (en caso de que existan) las No Conformidades en la plantilla 0312_Plantilla de Registro de Evaluaciones del proyecto.

2.2.2.6 Procesos de ACS para la arquitectura del Nova - Xerberos.

Arquitectura	
Nombre del proceso	Verificación del subproceso definición de arquitectura.
Meta	Verificar que el subproceso definición de arquitectura se haga correctamente.
Actividades que componen el proceso	<ul style="list-style-type: none"> - Comprobación de la correctitud de la arquitectura seleccionada para el sistema. - Controlar que el equipo de trabajo conoce y domina la arquitectura seleccionada para el sistema.
Criterios de evaluación	<p>Los criterios de evaluación los pueden proporcionar las siguientes preguntas:</p> <ul style="list-style-type: none"> • ¿Es robusta la arquitectura seleccionada para el sistema? ¿Se ajusta a las necesidades del software? Nivel de importancia (10%). • ¿Están bien identificados los principales componentes de la arquitectura, sus interacciones y sus dependencias? Nivel de importancia (10%). • ¿Existe un escenario arquitectónico para cada requisito funcional así como para cada requisito no funcional? Nivel de importancia (20%).

- ¿Ha sido probada la integración de los componentes de la arquitectura? Nivel de importancia (10%).
 - ¿Está documentada la selección de la arquitectura con sus respectivas vistas? En Nova – Xerberos las vistas imprescindibles son: Vista de Datos, Vista de Presentación, Vista de Proceso, Vista de Sistema, Vista de Seguridad y la Vista de Infraestructura. Nivel de importancia (25%).
 - A continuación se describe por cada una de estas vistas los parámetros a verificar que no pueden faltar.
- ❖ **Vista de Proceso:**
1. Verificar que se caracterice el entorno según la vista.
 2. Conceptualizar los macro procesos del entorno corporativo.
 3. Verificar que la vista cuente con un mapa conceptual por procesos.
 4. Verificar que existe priorización de los procesos del negocio
- ❖ **Vista de Sistema:**
1. Verificar que existe priorización de los requisitos funcionales del sistema.
 2. Verificar que haya agrupamiento y priorización de los requisitos.
 3. Controlar que exista identificación de los paquetes y componentes principales.
 4. Verificar que exista representación de la dependencia entre paquetes.
 5. Chequear que exista la descripción de los escenarios y patrones de solución.
- ❖ **Vista de Presentación:**
1. Verificar que existe la descripción de las posibles pantallas o interfaces del sistema.
 2. Verificar que se definan áreas en pantalla: Establece las áreas que han sido identificadas como estándar en los análisis realizados de sistemas nacionales e internacionales y la descripción de las mismas.
 3. Verificar que se definan las características funcionales de los contenidos genéricos por áreas.
 4. Chequear que existe la descripción textual de los elementos de la estructura, sus características y su comportamiento.
- ❖ **Vista de Datos:**
1. Chequear que existe la descripción de los escenarios y los patrones de solución.
 2. Verificar que existe un diccionario de Datos.
 3. Verificar que existe el modelo de datos.
- ❖ **Vista de Seguridad:**
1. Verificar que estén definidas las metas de seguridad del producto desarrollado.
 2. Verificar que se haga uso de la defensa en profundidad.
 3. Chequear que se asume que los sistemas externos son inseguros.
- ❖ **Vista de Infraestructura:**

	<p>1. Chequear que se describa el diagrama de configuración de redes.</p> <p>2. Verificar que se describa el diagrama de configuración del software.</p> <p>3. Verificar que se describa el diagrama de configuración de hardware.</p> <ul style="list-style-type: none"> • ¿Se han hecho pruebas de aceptación o pruebas de concepto de arquitectura para cada subsistema? Nivel de importancia (25%). <p>A. Se calcula el valor del Tiempo Medio entre Fallas (MTBF) de cada subsistema durante las pruebas operacionales del sistema o en las pruebas de concepto de arquitectura para cada uno de los subsistemas.</p> $MTBF = \frac{T1 + T2 + \dots + Tn}{n}$ <p>T: tiempo en funcionamiento n: total de horas análisis</p> <p>B. Se calcula el valor del Tiempo Medio para Restaurar (MTTR) de cada subsistema durante las pruebas operacionales del sistema o en las pruebas de concepto de arquitectura para cada uno de los subsistemas.</p> <div style="border: 1px solid black; padding: 5px; display: inline-block;"> $MTTR = \frac{t1 + t2 + \dots + tn}{n}$ </div> <p>t: Tiempo de fallo del sistema n: total de horas análisis</p> <p>C. Se calcula la disponibilidad de cada uno de los subsistemas durante las pruebas de concepto de arquitectura.</p> $Disponibilidad = \frac{MTBF}{MTBF + MTTR}$ <p>D. Se calcula la disponibilidad del sistema global combinando los subsistemas a partir de las siguientes fórmulas</p> <ul style="list-style-type: none"> • Disponibilidad de sistema serial = $D1 * D2 * \dots * Dn$ • Disponibilidad de sistema paralelo = $1 - (1 - D1) * (1 - D2) * \dots * (1 - Dn)$ <p>Se utiliza para estos cálculos el diseño que se obtuvo en la vista de arquitectura del sistema.</p> <p>Se debe analizar el resultado de la disponibilidad y avanzar hacia un sistema desacoplado y realizar las adecuaciones necesarias para aumentar la disponibilidad de los sistemas.</p>
<p>Roles</p>	<ul style="list-style-type: none"> - Líder del proyecto - Administrador de la calidad.
<p>Artefactos de Entrada</p>	<p>Según lo establecido en el Programa de Mejoras se tienen como entradas los artefactos:</p> <ul style="list-style-type: none"> - 0120_0 Arquitectura de Software Guia Base.doc - 0120_1 Arquitectura Vista de Proceso.doc - 0120_2 Arquitectura Vista Sistema.doc - 0120_3 Arquitectura Vista de Presentación.doc - 0120_4 Arquitectura Vista de Datos.doc

	<ul style="list-style-type: none"> - 0120_7 Arquitectura Vista de Seguridad.doc - 0120_8 Arquitectura Vista de Infraestructura.doc <p>El proyecto Nova – Xerberos al utilizar la metodología SXP tiene el artefacto de entrada siguiente:</p> <ul style="list-style-type: none"> - Plantilla de Arquitectura de Software.
Artefactos de Salida	<p>Según lo establecido en el Programa de Mejoras se tienen como salidas los artefactos:</p> <ul style="list-style-type: none"> - 0120_0 Arquitectura de Software Guia Base.doc - 0120_1 Arquitectura Vista de Proceso.doc - 0120_2 Arquitectura Vista Sistema.doc - 0120_3 Arquitectura Vista de Presentación.doc - 0120_4 Arquitectura Vista de Datos.doc - 0120_7 Arquitectura Vista de Seguridad.doc - 0120_8 Arquitectura Vista de Infraestructura.doc <p>El proyecto Nova – Xerberos al utilizar la metodología SXP tiene el artefacto de salida siguiente:</p> <ul style="list-style-type: none"> - Plantilla de Arquitectura de Software.
Posible acción correctiva	<p>Si no existen los artefactos del proceso de mejora (Plantillas: 0120_0 Arquitectura de Software Guia Base.doc, 0120_1 Arquitectura Vista de Proceso.doc, 0120_2 Arquitectura Vista Sistema.doc, 0120_3 Arquitectura Vista de Presentación.doc, 0120_4 Arquitectura Vista de Datos.doc, 0120_7 Arquitectura Vista de Seguridad.doc, 0120_8 Arquitectura Vista de Infraestructura.doc), ni la Plantilla de Arquitectura de Software por la metodología SXP, o en caso contrario, si necesitan alguna modificación, como medida de acción correctiva indicar realizar la confección de los mismos en un tiempo prudencial y acorde al cronograma del proyecto y posteriormente realizar seguimiento y control a los errores detectados.</p> <ul style="list-style-type: none"> - Plasmar (en caso de que existan) las No Conformidades en la plantilla 0312_Plantilla de Registro de Evaluaciones del proyecto.

2.3 Plan de Implementación

Para aplicar la guía es necesario estudiar en profundidad cada una de las buenas prácticas y los procesos de ACS a los que se hace referencia en ella y por otra parte tener en cuenta que está diseñada para los elementos verificables dentro del proceso de desarrollo, como son: el personal, el producto y los elementos de configuración. La guía puede ser utilizada cada vez que el equipo de trabajo necesite verificar y validar que su proceso de desarrollo esté acorde con lo estipulado por la Universidad, sus directrices internas y lo definido en el proceso de mejora, pudiendo ocurrir esto inclusive de forma periódica. En caso que no exista en la planificación del proyecto su puesta en marcha, se recomienda que se realice cada 3 meses. Haciéndolo así se podrá identificar y corregir errores desde etapas tempranas, obtener software con altos parámetros de calidad y por consiguiente reducir esfuerzos en cuanto a planificación, formación del personal y costos monetarios de creación del software, así como de recursos. Es necesario también que las actividades de aseguramiento de la calidad sean agregadas al cronograma del proyecto.

Capítulo # 3. Evaluación de la guía para asegurar la calidad en el sistema de gestión de estaciones Nova – Xerberos

Una vez concluida la propuesta de la guía para administrar la calidad de software en el sistema de gestión de estaciones Nova – Xerberos se hace necesaria la evaluación y verificación de la misma. Este proceso es crucial para determinar cuán necesaria, factible y útil puede resultar para los miembros del equipo de desarrollo la puesta en ejecución de la guía. Con la finalidad de estimar y conocer datos concisos sobre el comportamiento de la propuesta de solución se determina utilizar como método de pronóstico cualitativo, uno de los más populares, el método Delphi mediante el criterio de expertos.

3.1 Evaluación por el método Delphi

El método Delphi consiste en la selección de un grupo de expertos a los que se les aplican cuestionarios sobre temas referidos al problema en análisis y a la solución propuesta. Mediante las respuestas éstos expresan su punto de vista y éstas se tabulan para obtener los resultados de forma estadística. [1]

El método suele dividirse en tres etapas o fases fundamentales:

- **Fase inicial:** Es donde ocurre la formulación del problema. Las preguntas deben ser precisas, cuantificables (versan por ejemplo sobre probabilidades de realización de hipótesis y/o acontecimientos, la mayoría de las veces sobre datos de realización de acontecimientos) e independientes (la supuesta realización de una de las cuestiones en una fecha determinada no influye sobre la realización de alguna otra cuestión).
- **Fase exploratoria:** Es en esta fase donde ocurre la selección de los expertos con toda la serie de parámetros que la misma requiere.
- **Fase final:** Es donde ocurre el desarrollo práctico y la explotación de los resultados.

Las características principales de Delphi son:

- ❖ **Anonimato:** se evidencia porque los expertos no se conocen y dan sus respuestas sin llegar a confrontaciones entre ellos y también impide la posibilidad de que un miembro del grupo sea influenciado por la reputación de otro o que cambie sus opiniones sin que eso suponga una pérdida de imagen.
- ❖ **Iteración:** se pueden hacer tantas rondas como sean necesarias para obtener datos concisos.

- ❖ **Retroalimentación controlada:** después de cada ronda de preguntas se tabulan las respuestas y se procesan de forma tal, que antes de la siguiente ronda los participantes pueden evaluar los resultados de la ronda anterior, así como las razones dadas para cada respuesta y su dispersión del promedio.
- ❖ **Respuesta estadística del grupo:** entre cada ronda de preguntas, la información obtenida se procesa por medio de técnicas estadístico-matemáticas, las que dotan al investigador de un instrumento objetivo y concreto en el cual pueden apoyarse para tomar una decisión final.

3.2 Fase inicial

Durante esta primera fase se delimita el contexto, el objetivo a alcanzar con la aplicación del método seleccionado, el diseño, los elementos básicos del trabajo y por último se produce la selección de los expertos que protagonizarán la evaluación de los resultados.

3.2.1 Formulación del problema

Primeramente se definen los atributos **(C)** a evaluar por los especialistas, éstos deben ser precisos, medibles e independientes. Los atributos que se tuvieron en cuenta fueron los siguientes:

- C1. Importancia de la creación y utilización de la guía para asegurar la calidad de software en el sistema de gestión de estaciones Nova – Xerberos.**
- C2. Alcance de la guía.**
- C3. Necesidad de recurrir a la guía para asegurar la calidad de software en el sistema de gestión de estaciones Nova – Xerberos.**
- C4. Complejidad en la aplicación de la guía.**
- C5. Efectividad de la guía.**
- C6. Adaptabilidad a los proyectos que desarrollan sistemas para gestionar estaciones.**
- C7. Nivel de importancia de los procesos de aseguramiento de calidad que ofrece la guía.**
- C8. Influencia de la guía en la obtención de buenas prácticas para el desarrollo del sistema Nova- Xerberos.**

Estos criterios constituyen la base para la confección del cuestionario que se les aplica a los especialistas.

3.2.2 Selección del grupo de expertos

Se realiza la selección de los especialistas bajo las siguientes condiciones:

- Debe ser graduado de nivel superior.
- Debe estar vinculado a la investigación y al desarrollo de sistemas informáticos.
- Debe poseer conocimiento acerca de la gestión de la calidad de software.

- Debe contar con un año de experiencia como mínimo en el tema.
- Debe tener noción y habilidades sobre los procesos de calidad del software.
- Debe conocer y saber aplicar buenas prácticas de gestión de calidad de software en proyectos de desarrollo de aplicaciones.
- No necesariamente un mismo especialista tiene que dominar todos los temas.

Luego de seleccionados los temas en los que deben tener conocimiento los especialistas se prosigue a realizar su elección. Para la evaluación de esta investigación se seleccionaron 7 especialistas.

Fue necesario determinar las competencias de cada uno de ellos, esto consiste en el nivel de calificación en la esfera de conocimiento tratada; en algunas ocasiones se tiende a pensar que la competencia está dada por el nivel científico o el cargo que ocupan, sin embargo no siempre esto determina el grado de conocimiento.

La competencia de los especialistas se determina por el coeficiente K, que se calcula haciendo uso de la siguiente fórmula matemática:

$$K = \frac{1}{2} (Kc + Ka)$$

Donde:

Kc: es el coeficiente de conocimiento del experto sobre el tema.

Ka: es el coeficiente de argumentación.

Kc, es el coeficiente de conocimiento o información que tiene el posible experto acerca del problema, calculado sobre la valoración del propio experto en una escala del 0 al 10, de esta forma, la evaluación "0" indica que el experto no tiene absolutamente ningún conocimiento de la problemática correspondiente, mientras que la evaluación "10" significa que el experto tiene pleno conocimiento de la misma. El experto debe marcar con una cruz (X) en la casilla que estime pertinente. Posteriormente este valor obtenido se multiplica por 0.1 para obtener el coeficiente en un rango de 0 a 1.

$$Kc = \text{criterio} * 0.1$$

El **Ka** es el coeficiente de argumentación o fundamentación de los criterios del experto. Para calcular este coeficiente, el experto debe clasificar en alto, medio o bajo según su consideración y cuáles fueron sus fuentes para la obtención del conocimiento. Cada nivel de clasificación posee un valor y la suma de los valores marcados por cada criterio será el coeficiente de argumentación (Ka) del candidato a experto.

$$Ka = \sum \text{valores TP}$$

Después de haber calculado el **Ka**, se evalúa cuáles de los candidatos pueden pasar a ser expertos. La selección final de los expertos tendrá lugar atendiendo a los siguientes criterios de interpretación del coeficiente de competencia (**K**):

- Si $0.8 \leq k \leq 1.0$, el coeficiente de competencia es alto.
- Si $0.5 \leq k < 0.8$, el coeficiente de competencia es medio.
- Si $k < 0.5$, el coeficiente de competencia es bajo.

Es recomendable incluir en el grupo de expertos a aquellos que posean un coeficiente de competencia medio o alto, mayor que 0.5.

En la siguiente tabla se muestran los resultados obtenidos en la encuesta de autovaloración realizada a los especialistas seleccionados.

Experto	Coeficiente de conocimiento Kc	Coeficiente de argumentación Ka	Coeficiente de competencia K	Grado de influencia del coeficiente de competencia.
1	0.8	0.9	0.85	Alto
2	0.9	0.8	0.85	Alto
3	0.7	0.9	0.80	Alto
4	0.8	0.8	0.80	Alto
5	0.9	0.8	0.85	Alto
6	0.8	0.9	0.85	Alto
7	1.0	1	1	Alto

Para la evaluación de la propuesta de la investigación los 7 especialistas cumplen con la salvedad de tener un coeficiente de conocimiento alto y fue por tal motivo que fueron seleccionados.

3.3 Fase exploratoria

En esta etapa las principales actividades a realizar son la elaboración y aplicación de los cuestionarios, el método puede tener tantas iteraciones como sean necesarias, pero en este caso, debido al alto ritmo de trabajo de la UCI se hace difícil y extenso el proceso de hacer más de una ronda de preguntas. Para la realización de estas actividades se deben tener presente la longitud y el tipo de pregunta, pues un planteamiento demasiado conciso provoca una excesiva variedad de interpretaciones y uno muy largo requiere administrar demasiados elementos de una sola vez, por lo que la forma en que se realiza el cuestionario es con planteamientos de mediana longitud.

3.3.1 Elaboración del cuestionario

Inicialmente se hace entrega de un resumen de la propuesta que se desea validar a todos los expertos, lo cual permite la comprensión de la solución por parte del panel posibilitando la resolución de la encuesta.

El cuestionario se centra en indicadores que debe cumplir la propuesta para que sea válida. Se

definen un conjunto de preguntas abiertas y cerradas que contribuyen a la obtención de evaluaciones y criterios de los expertos. Las preguntas abiertas contienen un espacio donde se les permite a los especialistas exponer su opinión personal de la propuesta realizada, lo cual es muy importante pues permite conocer sugerencias que pueden influir en el comportamiento de una determinada cuestión. Por otra parte las preguntas cerradas admiten registrar ciertos aspectos con el objetivo de mostrar gráficamente los resultados obtenidos en el cuestionario. La respuesta para este tipo de preguntas se representa en una escala dividida en 5, donde 1 constituye el nivel más bajo y el 5 el máximo nivel, sin importar si esta división está propuesta de forma cuantitativa (%) o cualitativa (muy alto, alto, medio, bajo, muy bajo). El cuestionario es el Anexo # 1 de la guía.

3.4 Fase final

En esta fase es donde se lleva a cabo el análisis estadístico de los datos que fueron obtenidos luego de aplicar el cuestionario a los especialistas. Es fundamental calcular el porcentaje de las respuestas ofrecidas por los especialistas, pues este valor va a indicar el nivel de importancia (en dependencia del porcentaje) que le concede cada especialista a cada uno de los atributos sometidos a evaluación.

3.4.1 Cálculo de concordancia entre los expertos

Una vez definido el grupo de especialistas y contando con los resultados de la encuesta respondida por ellos, se analizó sus criterios sobre la guía de buenas prácticas para gestionar la calidad de software en el proceso de desarrollo de Nova - Xerberos. Para realizar la explotación de los resultados y con el objetivo de aportarle un mayor peso a la validación, se decidió calcular el coeficiente de concordancia Kendall (W) entre los especialistas, este coeficiente posibilita conocer el grado de concordancia respecto a las temáticas preguntadas. Fue posible calcularlo utilizando el software estadístico Statistical Product and Service Solutions (SPSS).

Los valores del W deben oscilar entre 0 y 1 ($0 < W < 1$), mientras más próximos se encuentren los valores a 1, mayor será el nivel de concordancia entre los criterios de los expertos.

N	7
Kendall's W ^a	.960
Chi-Square	47.052
df	7
Asymp. Sig.	.000

Figura 19. Datos obtenidos mediante el Coeficiente de Kendall.

a. Kendall's Coefficient of Concordance

El valor obtenido por la herramienta posibilita asegurar que con un 96% de concordancia en sus

respuestas los especialistas avalan y están de acuerdo en que la guía es importante, eficiente y muy necesaria para el proyecto productivo Nova – Xerberos. Finalmente el autor de la investigación pudo determinar que la guía cumple con los objetivos para los que fue creada.

3.4.2 Análisis de resultados de la encuesta a los especialistas

A continuación se muestran los gráficos de los resultados obtenidos por cada uno de los criterios establecidos.

El siguiente gráfico muestra las respuestas obtenidas para la pregunta 1.



Gráfico 1: Importancia de la creación de la guía

La gráfica permite verificar que todos los especialistas le confieren a la guía una gran importancia y consideran que su aplicación y puesta en marcha va a incidir en la obtención de un sistema con altos parámetros de calidad. Se obtuvo un 100% de aceptación de la propuesta en esta pregunta por parte de los especialistas.

El siguiente gráfico muestra las respuestas obtenidas para la pregunta 2.



Gráfico 2. Alcance de la guía según las actividades de ACS que define.

Como evidencia la figura hay un alto grado de aceptación de la guía en cuanto a su alcance por parte los especialistas. Interpretando la gráfica se puede constatar que 3 (E1, E4, E7) de ellos piensan que la guía es lo suficientemente abarcadora en cuanto a los procesos de ACS y 4 (E2, E3, E5, E6) la encuentran a un 80 % de alcanzar esta meta. Se puede dilucidar entonces que los procesos definidos en la guía son los correctos para el software en cuestión y que la puesta en práctica de la propuesta dada va a ofrecer al sistema eficientes procesos de ACS.

El siguiente gráfico muestra las respuestas obtenidas para la pregunta 3.



Gráfico 3. Necesidad de aplicación de la guía.

Como se puede ver en la gráfica anterior los especialistas llegan al consenso de que es muy necesario poner cuanto antes en práctica la guía en Nova – Xerberos para de esa forma poder comenzar a aplicar verdaderos patrones de gestión de la calidad para que el sistema a desarrollar sea todo lo fiable, eficiente y seguro posible. Se obtuvo un 100% de aceptación de la propuesta en esta pregunta por parte de los especialistas.

El siguiente gráfico muestra las respuestas obtenidas para la pregunta 4.

¿Existirán dificultades para aplicarla?

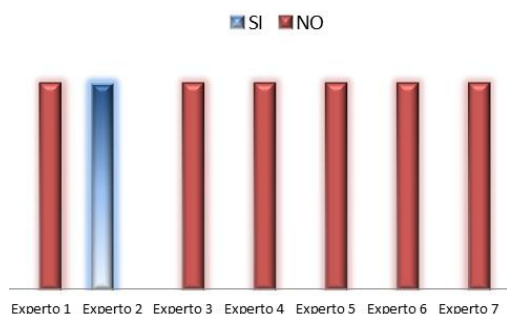


Gráfico 4. Dificultad en la aplicación en la guía.

Como se puede ver en la gráfica anterior 6 de los 7 especialistas consideran que no sea dificultosa la puesta en práctica de la guía en Nova – Xerberos puesto que la consideran intuitiva, comprensible y aseveran que están bien definidas las acciones a tomar y lo que hay que realizar para cada subproceso definido en la investigación.

El especialista que cree pueda existir algún problema plantea que puede ser por el escaso conocimiento sobre temas de calidad de software del equipo de desarrollo y el poco manejo de las temáticas establecidas en el programa de mejoras.

Se obtuvo un 85.7% de aceptación de la propuesta en esta pregunta por parte de los especialistas.

El siguiente gráfico muestra las respuestas obtenidas para la pregunta 5.

¿Será efectiva para Nova - Xerberos?

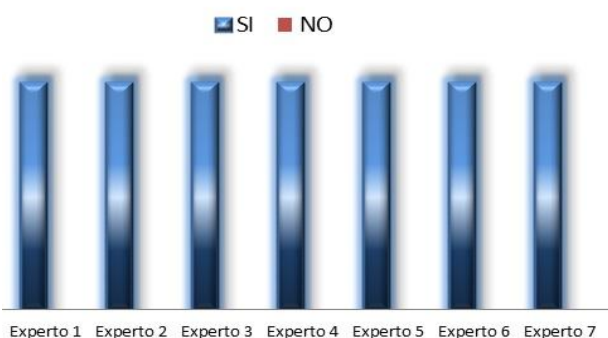


Gráfico 5. Efectividad en la aplicación en la guía.

Los especialistas con un 100% de concordancia piensan que al aplicar la guía la mejoría del sistema puede ser notable y que resulta realmente necesaria su puesta en marcha en el proyecto Nova - Xerberos. También se obtuvieron satisfactorias respuestas como:

- Con la utilización de una guía como esta los desarrolladores podrán tener el camino más guiado hacia la finalidad de obtener software con menor cantidad de errores y también se

debe establecer un cronograma para ajustar y enfocar los objetivos en una misma dirección.

- Al no existir nada que le dijera al equipo de trabajo cómo, con qué y mediante qué recursos hacer las cosas las personas podían desorganizarse y de la forma que se plantea en la guía todo se puede mirar desde una perspectiva más funcional, el sistema pudiera mejorarse significativamente e incluso en futuras modificaciones y versiones posteriores.

El siguiente gráfico muestra las respuestas obtenidas para la pregunta 6.

¿Será adaptable a proyectos similares?

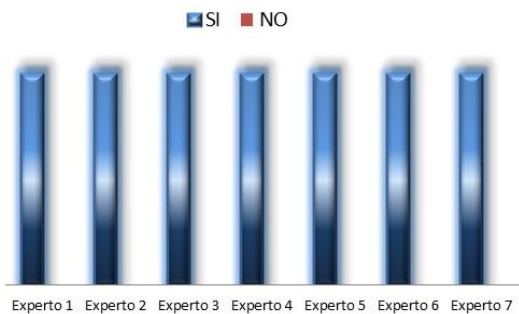


Gráfico 6. Adaptabilidad de la guía a otros sistemas similares.

Al hacerle la pregunta a los especialistas sobre si es posible que otros proyectos que desarrollen sistemas similares puedan adaptar lo planteado en la guía para sus procesos de desarrollo se pudo constatar que todos concuerdan en que sería totalmente posible e idóneo, inclusive si se decidiera llevar a cabo algo así en este centro de altos estudios, la UCI, o en el país pudiese la guía servir como un punto de partida determinante para no incurrir en errores antes cometidos. Se obtuvo un 100% de aceptación para esta pregunta.

Como se puede observar en la imagen debajo se grafica por cada encuestado cuáles de los procesos de ACS que se definen en la guía él como especialista piensa que es más o menos importante para el sistema Nova – Xerberos y cómo se explica en la investigación realizada en función de medir finalmente la calidad total del software. El criterio se emite en dependencia del grado de importancia del proceso (de una escala de 1 a 5; 1 es el mínimo y 5 es el máximo nivel de importancia).

El siguiente gráfico muestra las respuestas obtenidas para la pregunta 7.

Procesos de ACS definidos en la guía

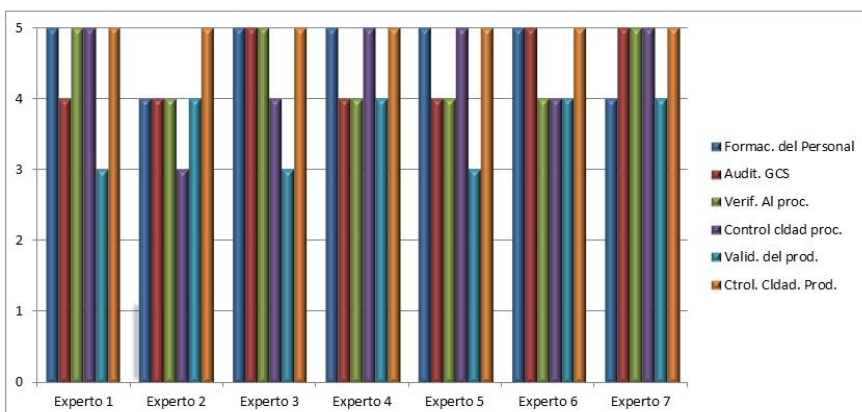


Gráfico 7. Procesos de ACS definidos en la guía.

Como se puede apreciar en la imagen superior los especialistas opinan que de los procesos que se detallan en la guía los que más importantes son y que mejor se explican en la solución de la investigación son los procesos de: *formación del personal, el control de la calidad al proceso de desarrollo y el control de la calidad al producto.*

El siguiente gráfico muestra las respuestas obtenidas para la pregunta 8.

¿En qué medida incrementará la guía la calidad de Nova - Xerberos?

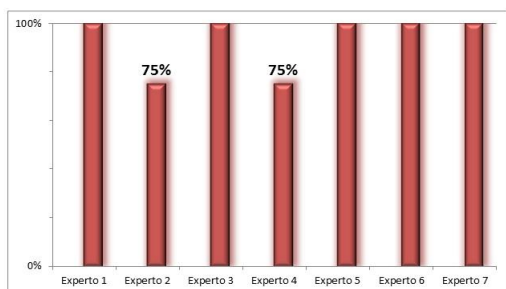


Gráfico 8. Procesos de ACS definidos en la guía.

Evidentemente los especialistas creen que si se aplica y se sigue a cabalidad lo plasmado en la guía el proyecto productivo puede alcanzar una gran mejoría. Alegan algunos que se logrará disciplina y cohesión en el equipo al estar rigiéndose por actividades descritas en un cronograma y bajo la orientación de cada paso a seguir.

El siguiente gráfico muestra las respuestas obtenidas para la pregunta 9.

¿En qué medida la guía cumple los objetivos para los que fue creada?

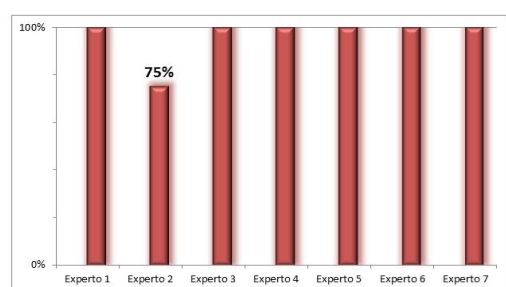
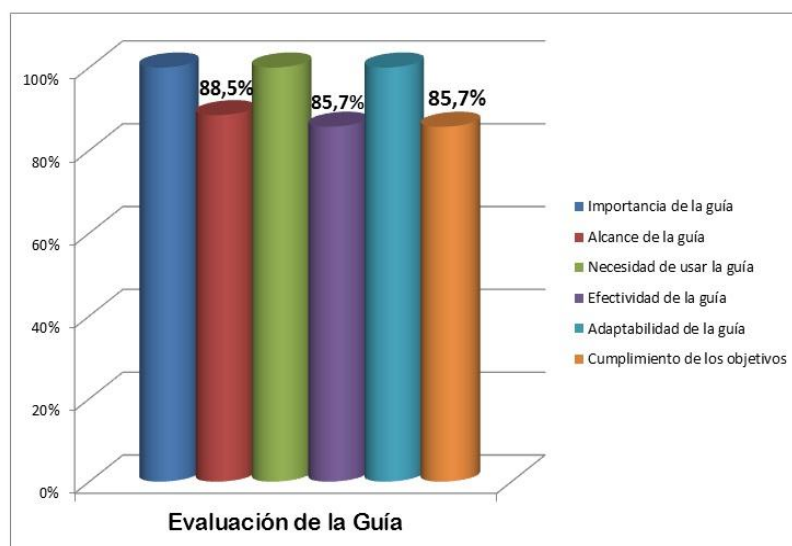


Gráfico 9. Cumplimiento del objetivo de la guía.

Por lo expuesto en el análisis de la gráfica 8 y como puede constatarse en la gráfica 9 (con una concordancia de un 85.7 %) los especialistas evalúan y responden que se alcanzó con la guía el objetivo principal de la investigación, quedando así evaluada satisfactoriamente por todos.

3.4.3 Resultado final

El porcentaje de respuestas de los especialistas a cada uno de los objetivos propuestos fue muy positivo, quedando por encima de la media y por tanto queda así de esta forma evaluada por especialistas la confección de la guía, los resultados generales se pueden observar en el siguiente gráfico. La solución propuesta es calificada por los especialistas como un procedimiento adecuado y la guía es avalada como necesaria para que el software que se desarrolla sea lo más óptimo posible.



Conclusiones

Al finalizar este trabajo de diploma se puede concluir que se pudieron alcanzar los resultados que se esperaban. Se obtuvo una guía de buenas prácticas que puede convertirse en medular para lograr una correcta administración de la calidad en el sistema para gestionar estaciones que se desarrolla en el proyecto antes mencionado. La investigación realizada también dejó como resultados positivos que:

- Se han investigado y estudiado una representativa variedad de sistemas existentes a nivel mundial para el control y la gestión de estaciones de forma centralizada, sus características fundamentales y las funcionalidades principales de dichos sistemas para tener un punto de partida a la hora de realizar la guía para gestionar la calidad en el sistema Nova Xerberos.
- El estudio acerca de los procesos, las técnicas, los modelos, estándares y normas que son utilizados tanto internacional como nacionalmente para la gestión de la calidad de software permitió sentar las bases en la investigación para luego diseñar los procesos adecuados según las características del sistema.
- La aplicación del método de expertos Delphi dio la posibilidad de evaluar la propuesta para la administración de la calidad, comprobando que la misma es correcta y completa. Además fundamentar que las actividades que la componen son las adecuadas y contribuyen a la obtención de un producto final exento de posibles errores gracias a las buenas prácticas plasmadas en la guía.

Recomendaciones

Para un perfeccionamiento de la solución brindada en la actual investigación se recomienda:

- ✓ Que se ponga en práctica la guía en el proyecto Nova – Xerberos para poder arribar al objetivo principal de la investigación, gestionar la calidad del sistema.
- ✓ Profundizar en el estudio de las temáticas referentes a la evaluación de los factores de calidad para así poder alcanzar un mayor nivel de optimización a la hora de desarrollar el sistema.
- ✓ Investigar sobre la realización de pruebas al sistema mediante el uso de herramientas automatizadas para que no sean netamente manuales, contribuyendo así a la eficiencia y fiabilidad total del sistema.

Referencias Bibliográficas

1. Lic. María Isabel de Arquer. *NTP 401: Fiabilidad humana: métodos de cuantificación, juicio de expertos*. [España]: Centro Nacional de condiciones de trabajo, enero 2010.
2. Anón. Normas ISO. [citado 10 diciembre 2011]. Available from world wide web: <<http://calisoft.uci.cu/index.php/documentos/12-normas-y-estandares-internacionales/24-iso>>.
3. Paulk, M. et al., *Capability Maturity Model for Software*, Software Engineering Institute, Carnie Mellon University, Pittsburgh, P.A., 2009.
4. Pressman, R.S., *Ingeniería del Software, un enfoque práctico*, 7ª ed., Mc Graw Hill, 2010.
5. VÁSQUEZ, G. R. *¿Cuánto conocemos de las normas ISO 9000?*, 2000.
6. Ing. Dayron Pérez Roldán, Ing. José Ernesto Torres Sánchez. xerberos - Revision 471: /Xerberos/stable. [citado 10 octubre 2011]. Available from world wide web: <<https://repositorio.geitel.prod.uci.cu/svn/xerberos/Xerberos/stable/Descripcion%20de%20Xerberos.odt>>.
7. Mikel Jiménez Hernández. *Irontec, Internet y sistemas sobre GNU/Linux*. [Bilbao]: IRONTEC, junio 2009.
8. Anón. Landscape | Canonical. [citado 4 noviembre 2011]. Available from world wide web: <<http://www.canonical.com/enterprise-services/ubuntu-advantage/landscape>>.
9. Miguel Á. García Lax, Fco. Javier García Ros, Angel Luís Mateo, Alfonso Marín Marín, Javier Tavira, Fco. Yepes Candel. *Gestión distribuida de servidores con herramientas Open Source: Cfengine y Subversion*. [España]: Universidad de Murcia, junio 2010 Available from

- world wide web: <<http://www.rediris.es/rediris/boletin/85-86/ponencia4.pdf>>.
10. Imobach González Sosa. *Gestión de sistemas con Puppet*. [Feria de Informática y las Telecomunicaciones]: TIC 2011, febrero 2011 Available from world wide web: <<http://projects.puppetlabs.com/projects/puppet/>>.
 11. Anón. Spacewalk: Free & Open Source Linux Systems Management. [citado 10 noviembre 2011]. Available from world wide web: <<http://spacewalk.redhat.com/>>.
 12. Anón. Remote System Management | Product Overview | DameWare NT Utilities. [citado 15 noviembre 2011]. Available from world wide web: <<http://www.dameware.com/products/dameware-nt-utilities/product-overview.aspx>>.
 13. Anón. Secospace TSM. [citado 15 noviembre 2011]. Available from world wide web: <http://www.huaweismantec.com/spanish/Product___Solution/Products/Security/Software/Secospace/Secospace_TSM/>.
 14. Daniel Marino Miralles Taset, César Raúl García Jacas. *T-arenal v2.0: Desarrollo del back - end*. junio 2009.
 15. Ing. Elizabeth Santana Beoto. *COSMO. Sistema de Gestión de Laboratorios o Agrupaciones de Computadoras*. enero 2010.
 16. INFORMÁTICAS, U. D. L. C. *Calidad del Software*. Conferencia Congreso FEU UCI, 2006. p.
 17. Lic. Cecilia Palazzolo. *Calidad de Software, Herramientas de Software*. [España]: Facultad de Informática, Ciencias de la Comunicación y Técnicas Especiales. junio 2009.
 18. TECHNOLOGIES, C. I. Q. *Introducción a la calidad y mejora del proceso software*. Catálogo de Cursos, 49 p.
 19. Anón. Normas IEEE. [citado 10 diciembre 2011]. Available from world wide web: <<http://calisoft.uci.cu/index.php/documentos/12-normas-y-estandares-internacionales/23-ieee>>.
 20. Gema de la Incera Torres. *Medidas de calidad en proceso, producto y mantenimiento, aplicadas al Control Estadístico de Procesos*. Escuela Superior de Informática de la Universidad de Castilla – La Mancha, abril 2008.
 21. GARCÍA, F. *El Proceso de Medición Software*. Proceso Software y Gestión del Conocimiento, Departamento de Tecnologías y Sistemas de Información. Escuela Superior de Informática. Universidad de Castilla-La Mancha, 2009. p.
 22. Dr. Angélica de Antonio. *LA GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE*.

- [España]: UPM, marzo 2009.
23. McCall, J.A., Richards, P.K. and Walters, G.F. “*Factors in Software Quality*”, RADC TR-77-369, US Rome Air Development Center Reports NTIS AD/A-049 014, 015, 055, 1977.
 24. Boehm, B.W., Kaspar, J.R. y otros “*Characteristics of Software Quality*”, TRW Series of Software Technology, 1978.
 25. GARCÍA, A. M. and N. ARAGÓN. *Análisis de modelos de calidad internacionales con respecto a su aplicación a la industria cubana del software*. *Revista Cubana de Ciencias*, 2006. 1.
 26. Anón. cmmi. [citado 10 diciembre 2011]. Available from world wide web: <<http://calisoft.uci.cu/index.php/documentos/12-normas-y-estandares-internacionales/26-cmmi>>.
 27. BAÑERES, J. P. *Ingeniería de procesos de software*. Compendio de Ingeniería del Software II, 2006a. p.
 28. Ing. Gladys Marsi Peñalver Romero. *MA-GMPR-UR2 Metodología ágil para proyectos de software libre*. octubre 2008.
 29. Eva- UCI. *El Proceso Unificado de Modelado*. Available from world wide web: <[http://eva.uci.cu/file.php/161/Documentos/Materiales complementarios/UD 1 Procesos/Metodologias/RUP/RUP.pdf](http://eva.uci.cu/file.php/161/Documentos/Materiales_complementarios/UD_1_Procesos/Metodologias/RUP/RUP.pdf)>.
 30. KRUCHTEN, K. Y. *The Rational Unified Process Made Easy*. Addison Wesley, 2003. p.
 31. Anón. Documentos Programa de Mejora. [citado 10 enero 2012]. Available from world wide web: <<http://calisoft.uci.cu/index.php/proceso-de-mejora/46>>.

Bibliografía

- 1- **Roger S Pressman, 2006.** “*Ingeniería del Software: Un enfoque práctico*”, Sexta edición, México DF, Editorial McGraw Hill.
- 2- **Roger S Pressman, 2010.** “*Software Engineering a Practitioners Approach*”. Seventh edition Editorial McGraw Hill.
- 3- **Ian Sommerville, 2005.** “*Ingeniería del Software*”, Séptima edición, México DF, Editorial Pearson.
- 4- **Gladys Marsi Peñalver Romero, Sergio Jesús García de la Puente, Abel Meneses Abad, 2010.** “*SXP, metodología de desarrollo de software*”.
- 5- **IEEE Std. 610.12 (1990).** *IEEE Standard Glossary of Software Engineering Terminology*. Available from world wide web: <<http://www.standards.ieee.org>>.

6- **Cusmai, Marcelo.** *Revista digital.* **Abril 2011.** *Calidad del Software desde Latinoamérica. Infotesting. Un espacio para testers profesionales.* Disponible en: <http://www.infotesting.com>.

7- **Santana, Ildian Guzmán.** **Ciudad de la Habana: s.n., 2008.** *Plan de Aseguramiento de la Calidad para proyectos de software sobre la Arquitectura MDA (Model Driven Architecture).*

8- **GARCÍA, A. M. and N. ARAGÓN. S.I.: Revista Cubana de Ciencias, 2006, Vol. Vol1.** *Análisis de modelos de calidad internacionales con respecto a su aplicación a la industria cubana del software.*

9- **ISO 900-3** "Guía para la aplicación de ISO 9001 para el desarrollo, implementación y mantenimiento de software".

Anexos

1- Cuestionario

Las siguientes preguntas tienen como finalidad la evaluación de la Guía para asegurar la calidad de software en el desarrollo del sistema para gestionar estaciones Nova - Xerberos.

PREGUNTA 1

¿Cree usted que es importante la creación y utilización de la guía para orientar a un mejor aseguramiento de la calidad de software en el proceso de desarrollo del sistema para gestionar estaciones Nova - Xerberos?

Si No No sé. Argumente su respuesta.

PREGUNTA 2

Demuestre qué tan abarcadora cree usted que es la guía propuesta. Tenga en cuenta que 1 representa que no es abarcadora y que el máximo valor es 5. En caso de que su respuesta sea 1, argumente.

1	2	3	4	5

PREGUNTA 3

¿Considera usted que actualmente existen resultados o situaciones que reflejan la necesidad de recurrir a la guía de gestión de la calidad de software durante el desarrollo de sistemas para gestionar estaciones?

Si No No sé. Argumente su respuesta.

PREGUNTA 4

¿Según sus conocimientos cree usted que existirá alguna dificultad en el proceso de aplicación de la guía?

___ Si ___ No. Argumente su respuesta.

PREGUNTA 5

¿Considera usted que la guía de buenas prácticas pudiera ser efectiva?

___ Si ___ No ___ No sé ¿Por qué?

PREGUNTA 6

¿Piensa que la guía es adaptable a proyectos desarrolladores de sistemas para gestionar estaciones?

___ Si ___ No. Argumente su respuesta.

PREGUNTA 7

Refiriéndose a cuán efectivo considera usted que es el proceso de aseguramiento de la calidad si se tiene en cuenta en el proceso de desarrollo de sistemas para gestionar estaciones otorgue a cada uno de ellos una calificación entre 1 y 5 puntos, teniendo en cuenta que 1 es el nivel más bajo y 5 el máximo nivel. Si cree preciso eliminar o poner alguno, méncionelo y explíquelo brevemente.

PROCESOS DE ASEGURAMIENTO DE LA CALIDAD	VALOR
Formación al personal	
Auditoria de la gestión de la configuración en el proyecto	
Verificación al proceso	
Control de la calidad al proceso	
Validación al producto	
Control de la calidad al producto	

PREGUNTA 8

Con la propuesta de la guía de buenas prácticas, como resultado del trabajo ¿En qué medida cree usted que se incrementará la calidad del sistema para gestionar estaciones Nova – Xerberos que se desarrolla?

___100% ___75% ___50% ___25% ___0%

PREGUNTA 9

Indique en qué por ciento cree que la guía cumple con el objetivo para la cual fue creada.

___100% ___75% ___50% ___25% ___0%

Otras consideraciones

PREGUNTA 10

¿Qué usted le añadiría a la guía?