Universidad de las Ciencias Informáticas
Facultad 7

Título: Propuesta de un procedimiento para garantizar la seguridad en el entorno de desarrollo y productos obtenidos en los proyectos de la Facultad 7.

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas

Autora: **Geidy Fernández Rodríguez.**

Tutora: **Ing. Arianne Méndez Mederos.**

Ciudad de la Habana, junio 2008

¨Año 50 del Triunfo de la Revolución¨

DECLARACIÓN DE AUTORÍA

Declaro ser autora de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los 19 días del mes de Junio del año 2008.

Geidy Fernández Rodríguez

Ing. Arianne Méndez Mederos

_____

Firma del Autora

_____

Firma del Tutora

Datos de Contacto

Ing. Arianne Méndez Mederos (arianne@uci.cu): Graduada de Ingeniero en Ciencias Informáticas en el año 2006. Posee la categoría docente de Instructor y actualmente esta cursando la maestría de Gestión de Proyectos Informáticos. Actualmente se encuentra laborando en la Facultad 7 de la Universidad de las Ciencias Informáticas como profesora de Seguridad Informática y es líder del proyecto Atención Remota a Servidores en el Área Temática Sistema de Apoyo a la Salud.

"Si piensas que la tecnología puede solucionar tus problemas de seguridad, está claro que ni entiendes los problemas, ni entiendes la tecnología"

Bruce Schneier.

Agradecimientos

A mi mamá, por su eterno amor, por ser mi mejor amiga, estar conmigo en todo momento y ser mi principal fuente de inspiración. A ella le debo lo que soy hoy.

A mi familia y en especial a mis hermanos, por su interés y preocupación.

A Vitico por cuidar de mi mami en todo este tiempo que estuve lejos.

A Jaylon por estar siempre que lo necesité, a pesar de la distancia.

A Pedro por su ayuda incondicional y por soportarme tanto tiempo.

A Daneysi por su infinita amistad.

A todas las compañeras de aula y apartamento, especialmente a Maria que me ayudó tanto en la etapa final.

A mi tutora por toda su ayuda y apoyo.

A Vladimir por siempre estar ahí.

A la Revolución Cubana por esta gran obra y a Fidel por su genial idea de crear la UCI.

Dedicatoria

Especialmente se la dedico a mi mamá porque sin ella nada de esto sería posible.

A mis hermanos Luis y Diógenes por su preocupación.

A Vitico ya que sin su ayudita en casa nada hubiera sido fácil.

A Jaylon que en la etapa final siempre me apoyo, me brindo su ayuda y su amor.

A Daneysi por ser la amiga que siempre quise tener.

A todas aquellas personas que me ayudaron y fueron parte de este periodo de mi vida.

Resumen

El presente trabajo se realizó en la Facultad 7 de la Universidad de las Ciencias Informáticas con el objetivo de realizar la propuesta de un procedimiento para garantizar la seguridad en el entorno de desarrollo y productos en los proyectos productivos de la facultad. Este propone una serie de pasos lógicos para el aseguramiento de la seguridad del software.

En la investigación se realizó un diagnóstico acerca de los problemas existentes en la producción del software, para ello se emplearon diferentes técnicas como la revisión de documentos, entrevistas, encuestas, entre otras .Se determinaron diferentes vulnerabilidades que pueden presentar los software, concluyendo que el aseguramiento de la seguridad es una actividad que se debe llevar a cabo durante todo el entorno de desarrollo del software.

Se proponen como principales etapas para aplicar el procedimiento: la planificación, construcción, supervisión y revisión de la seguridad. Cada una de ellas se rige por diferentes estándares, metodologías y paradigmas. Todas las etapas son importantes para la eficiencia del procedimiento, cada una depende de la eficacia de la anterior para su completamiento y en todas se debe hacer una supervisión de la seguridad para lograr una mayor eficiencia.

El trabajo tiene un valor teórico-práctico y metodológico pues se generaliza el concepto de seguridad como una necesidad inmediata para la institución, proponiéndose diferentes aspectos que deben cumplirse cuando se implemente la investigación en la producción.

Palabras clave: Procedimiento, Seguridad, Software, Entorno de desarrollo.

Tabla de Contenido

Tabla de Contenido

Introducción

Los requerimientos de seguridad que involucran las tecnologías de la información, en pocos años han cobrado un gran auge. El concepto de la seguridad en los sistemas de software es un área de investigación que ha pasado a ser vital dentro de la Ingeniería de Software. Con el crecimiento de Internet, y otras aplicaciones sobre redes, como el comercio electrónico, correo electrónico, etc., la posibilidad de ataques se ha incrementado notablemente, como también lo han hecho las consecuencias negativas de estos ataques.

“En la actualidad prácticamente todo sistema debe incorporar cuestiones de seguridad para defenderse de ataques maliciosos. El desarrollador ya no sólo debe concentrarse únicamente en los usuarios y sus requerimientos, sino también en los posibles atacantes. Esto ha motivado cambios importantes en el proceso de desarrollo del software para incorporar a la seguridad dentro de los requerimientos críticos del sistema.

Estos cambios son los primeros pasos en la concientización de los ingenieros de software acerca de la importancia de obtener un software seguro. Como todo gran cambio, no se logra de un día para otro, sino que es un proceso gradual que requiere tiempo y maduración. Todavía la Ingeniería de Software no ha dado una respuesta eficaz, coherente y aplicable que satisfaga plenamente a la comunidad informática, sino que las aproximaciones actuales están plagadas de fallas y debilidades fruto de que todavía es un proceso que se encuentra en su infancia.” [1]

“La seguridad es un requisito básico para los proveedores de software, obligados por las fuerzas del mercado, dada la necesidad de proteger infraestructuras de gran importancia y preservar la confidencialidad, integridad y disponibilidad de la información. Uno de los retos más importantes a los que se enfrentan todos los proveedores de software es crear un software más seguro que requiera menos actualizaciones y una administración de seguridad menos costosa.

Los proveedores de software deben adoptar un proceso de desarrollo más estricto, que se centre en mayor medida, en la seguridad. Este proceso debe diseñarse para minimizar el número de vulnerabilidades de seguridad presentes en el diseño, la implementación y la documentación, así como para detectarlas y eliminarlas en el ciclo de vida de desarrollo.” [2]

Cuba es un país donde el desarrollo del software es primario ya que no presenta muchas empresas que se dediquen a este sector. Por eso una de las principales obras del Gobierno cubano es el crecimiento en el desarrollo del software en las distintas ramas de la economía. No solamente con el beneficio de productos para la informatización de la sociedad sino también para la exportación. Para lograr este reto que se ha propuesto el país hay que obtener productos con calidad, lo que implica que el software posea la seguridad requerida. Esto conlleva a la utilización de procedimientos, metodologías, estándares para el desarrollo del software que permitan consolidar la filosofía de trabajo en aras de lograr una confidencialidad, integridad y disponibilidad de la información, a la vez que eleven la productividad, tanto para la labor de calidad, como para la obtención de un producto seguro. [3]

Actualmente en Cuba se están dando los primeros pasos para avanzar en cuanto al desarrollo del software, en ello participan diferentes empresas del Ministerio de la Informática y Comunicaciones como Desoft, Softel, PcMax, Sys, Segurmática, la Universidad de Ciencias Informáticas(UCI), entre otras.

La Universidad de las Ciencias Informáticas es un centro de nuevo tipo, que se encarga principalmente, de preparar ingenieros para el desarrollo de software tanto para la exportación como para la importación. La misma se divide en 10 facultades las cuales tiene un perfil vinculado a las distintas ramas de la economía. El reto que presenta hoy esta universidad es que los productos que en ella se realizan tenga una alta calidad, garantizando la seguridad de la información que manejan las aplicaciones.

La Facultad 7 tiene un perfil vinculado a la Informática en la Salud. En ella se han detectado problemas en el entorno de desarrollo del software, como por ejemplo que no se tiene un enfoque con respecto a la seguridad del software desde el inicio del producto, debido a que no se confeccionan los documentos que regulen la seguridad; no existe un Plan de Seguridad Informática, que es el documento que norma los procedimientos, medidas y mecanismos de seguridad durante todo el proceso; no se controla y ni verifica que se cumplan los requisitos no funcionales de seguridad que establece el cliente; no existen auditorias de seguridad. Cuando dan por concluido un producto, lo hacen sin realizar pruebas de seguridad, pues actualmente la facultad no tiene definidas estrategias para la realización de este tipo de pruebas.

Dada la situación anteriormente planteada, el **problema científico** radica en ¿Cómo garantizar la seguridad en el entorno de desarrollo y productos obtenidos en los proyectos de la Facultad 7?

El **Objeto de estudio** se enmarca en los mecanismos de seguridad que se llevan a cabo en el proceso de desarrollo de software.

El trabajo se concretará en los mecanismos de seguridad en el entorno de desarrollo y productos obtenidos en la Facultad 7, lo cual define el **Campo de acción.**

Para dar cumplimiento al problema científico se ha propuesto como **Objetivo de la Investigación:** Elaborar un procedimiento que garantice la seguridad en el entorno de desarrollo y productos en los proyectos de la Facultad 7.

Para dar cumplimiento al Objetivo planteado se han trazado las siguientes **tareas:**

1. Valorar los procesos llevados a cabo en la facultad 7 para garantizar la seguridad en el proceso de desarrollo del software.
2. Analizar los mecanismos de seguridad en el entorno de desarrollo del software.
3. Analizar las empresas que incorporan la seguridad desde la etapa inicial de concepción del producto.
4. Identificar las vulnerabilidades más comunes presentadas en el software.
5. Definir el procedimiento que deben seguir los proyectos productivos para garantizar la seguridad en la etapa de elaboración del producto.

Para su presentación, esta investigación se organizó en 3 capítulos, Conclusiones, Recomendaciones, Referencias bibliográficas, Glosario de términos y Anexos.

En el Capitulo I. Fundamentación Teórica: se definen los conceptos fundamentales y se analizan los fundamentos teóricos del tema , algunos aspectos importantes relacionados con la seguridad a nivel mundial , así como en Cuba , como también estándares , modelos y paradigmas que se encuentran relacionadas al tema.

En el Capitulo II. Se hace una caracterización de la universidad, para luego hacer un diagnóstico, a través de diferentes técnicas de la producción de software en la facultad 7.

En el Capitulo lll. Se realiza la propuesta de procedimiento para el fortalecimiento de la seguridad del software que se produce en la facultad 7 .dicho procedimiento tiene como principal objetivo promover la seguridad , logrando con esto la producción de software seguro en los proyectos productivos de la facultad 7.

Capítulo 1: Fundamentación Teórica

1.1 Introducción.

En este capítulo se realiza un análisis valorativo de la seguridad del software en el mundo y en Cuba .Todo esto visto también en la Universidad de la Ciencias Informáticas (UCI), siendo esta un punto de partida en el auge de la seguridad del software en Cuba. Así como también expondremos diferentes estándares para la realización de una seguridad adecuada en el software, como también uno de los paradigmas de la programación.

1.2 La seguridad en el software.

En pos de conseguir un software seguro, se debe dejar claro qué se entiende por seguridad en el software.

Como definición del concepto de seguridad en software, se adoptará en este trabajo la definición que propone Doshi Shreyas en [4]: la seguridad de un sistema de software es un concepto multi-dimensional. Las múltiples dimensiones de la seguridad son:

- **Autenticación:** el proceso de verificar la identidad de una entidad.
- **Control de acceso:** el proceso de regular las clases de acceso que una forma tiene sobre los recursos.
- **Auditoría:** un registro cronológico de los eventos relevantes a la seguridad de un sistema. Este registro puede luego examinarse para reconstruir un escenario en particular.
- **Confidencialidad:** la propiedad de que cierta información no esté disponible a ciertas entidades.
- **Integridad:** la propiedad de que la información no sea modificada en el trayecto fuente-destino.
- **Disponibilidad:** la propiedad de que el sistema sea accesible a las entidades autorizadas.
- **No repudio**: la propiedad que ubica la confianza respecto al desenvolvimiento de una entidad en una comunicación.

Bajo este punto de vista, se define un ataque a la seguridad como un intento de afectar en forma negativa una o más de las dimensiones del concepto de seguridad.

Una vez definido el concepto de seguridad, se pueden establecer objetivos básicos para un software seguro: [5]

- **Independencia de la seguridad**: la seguridad debe construirse y utilizarse de manera independiente de la aplicación.
- **Independencia de la aplicación**: la aplicación no debe depender del sistema de seguridad usado, debe ser desarrollada y mantenida en forma separada.
- **Uniformidad**: la seguridad debe aplicarse de manera correcta y consistente a través de toda la aplicación y del proceso que desarrolla la misma.
- **Modularidad**: mantener la seguridad separada. Entre otras ventajas, esto nos brinda mayor flexibilidad y menor costo de mantenimiento.
- **Ambiente seguro**: se debe partir de un entorno confiable. Es decir, las herramientas de desarrollo y lenguajes de programación no deben contener agujeros de seguridad.
- **Seguridad desde el comienzo**: la seguridad debe ser considerada como un requerimiento desde el inicio del diseño.

1.3 Problemática actual de la seguridad en el software.

Los puntos débiles más importantes de la Ingeniería de Software con respecto a la seguridad pueden ser clasificados en dos grandes categorías: [6]

- Fallas para implementar software seguro.
- Fallas para implementar seguridad en el software.

1.3.1 Fallas para implementar software seguro.

Lamentablemente, la mayoría de las herramientas que tiene disponible un desarrollador de software sufren de fallas propias de seguridad.

Una de las debilidades más trascendentes al momento de implementar software seguro surge del estado de los lenguajes de programación desde el punto de vista de la seguridad. Son escasos los lenguajes que proveen primitivas “seguras” que ayuden al programador a escribir un mejor código.

Dos de los lenguajes de programación más usados en la actualidad, C y C++,presentan graves problemas de seguridad Esto se debe a que al utilizar muchos de sus servicios provistos por sus

librerías estándar se introducen fallas de seguridad que pasarán inadvertidas al programador debido a que éste las considera libre de errores.[7]

Como ejemplo, podemos citar el problema de “buffer overflow”, fallas en la rutina malloc(a, b) y en la rutina rand () [8]. También, en un trabajo de John Viega y su equipo [9], se identifican numerosas fallas en el lenguaje C: en la rutina gets, donde también se explota el “overflow” de buffers, en las rutinas para manejar de strings strcat, strcpy, sprintf, en las rutinas system y popen, para correr programas desde la línea de comandos, que son generalmente usadas de manera incorrecta, y otras fallas más sutiles, que se pueden introducir al considerar cuestiones de sincronización. [10]

También lenguajes con arquitecturas de seguridad mucho más complejas, como Java, dejan mucho que desear. En [11], se establece que un alto porcentaje de aplicaciones tiene problemas de seguridad que están presenten desde la fase de diseño y que permanecen hasta la implementación, independientemente del lenguaje de programación usado. Las razones por las cuales estas fallas “internas” permanecen en la actualidad son varias: mal entendimiento de los protocolos de seguridad, una visión ingenua respecto a lo que un sistema debiera considerar como seguro, aproximaciones no serias a la seguridad como “corregir luego” o “no se van a dar cuenta”, o directamente desconocimiento, ya que lamentablemente estas fallas no son conocidas universalmente, y existen pocas fuentes de información para escribir código seguro. Otro tipo falla en esta categoría, que se establece en [12], nace del hecho de que la seguridad es un tema complejo y requiere un entendimiento completo sobre lo que puede ir mal y qué es lo que puede ser explotado por un posible atacante. Un programador promedio no cuenta con la experiencia suficiente como para poder determinar los requerimientos de seguridad que necesite su aplicación. Esto resulta en la subestimación de pequeños detalles que luego pueden llegar a introducir grandes fallas de seguridad. [13]

1.3.2 Fallas para implementar seguridad en el software.

En la actualidad, el desarrollador de software que quiera incorporar seguridad a su sistema se enfrentará con se la difícil realidad de las técnicas de programación tradicionales, y también, con una Ingeniería de Software que recién está aprendiendo sobre la seguridad. [14]

Típicamente la seguridad es considerada como un requerimiento no funcional. Luego, debido a los problemas de planificación y presupuesto, la seguridad sólo es tenida en cuenta una vez que los

requerimientos funcionales son obtenidos. Esto conduce a que la seguridad sea considerada como un concepto “afterthougth”, y que se incorpore tardíamente al sistema. Esto lleva a una implementación pobre, ineficiente, e inadecuada de la seguridad. También, la mayoría de las metodologías de diseño y herramientas dedicadas a la seguridad trabajan de esta forma, como herramientas “afterthougth”. Por lo tanto, es mandatorio que los conceptos de seguridad formen parte integral en todo el ciclo de vida de desarrollo de software, tal como se lo demanda en. [15]

“Una de las aproximaciones más ampliamente utilizada para la seguridad es la aproximación “ataque-parche” (en inglés “penetrate and patch”), en donde la seguridad es tratada de una manera ad-hoc, a medida que las fallas se van revelando.

Otra notable debilidad proviene del hecho de que las técnicas tradicionales de descomposición no logran encapsular correctamente el concepto de seguridad. Ya sea con la Programación Orientada a Objetos (POO), con la programación estructurada por bloques, o con la programación declarativa, no se consigue separar adecuadamente el concepto de seguridad del resto del sistema.” [16]

1.4 La seguridad del software a nivel mundial.

“La Industria del Software (ISW) es la industria que involucra la investigación, desarrollo, comercialización y distribución de software. En nuestros días vivimos una revolución en le ámbito de los sistemas informáticos y los paradigmas con los que veíamos la economía hace 10, 15, 20, 30 años no son los mismos .La ISW es una de las industrias con mayor crecimiento a nivel mundial valorada en 600 billones de dólares al año.

El desarrollo del software constituye un sector de gran importancia mundial se encuentra en le centro de todas las grandes transformaciones; teniendo en cuenta que los grandes temas son la economía digital, Internet, la evolución de las empresas y la administración del conocimiento se resuelven a través de software. La ISW interviene en todos los procesos que habilitan la economía, se le considera una industria blanca que no contamina y que genera fuentes de trabajo bien remuneradas .Actualmente el software representa una oportunidad para la economía de cualquier país ya que la misma se considera multimillonaria y no requiere grandes inversiones ya que su materia prima es el capital humano.” [17]

Existen países que han alcanzado un gran auge en la ISW como son India, Irlanda e Israel, como también en América como México, Colombia, Brasil, Argentina, Venezuela, Estados Unidos y Canadá, y otros como Rusia y España

El crecimiento de la ISW en la India es mundialmente conocido ya que es uno de los países que sea convertido en referencia mundial ya que exporta el 72% de lo que produce en materia de software .Allí se encuentran 42 de las 52 empresas de software certificadas con el nivel superior en calidad, estudios realizados estiman que para este año India exportara 57 billones en la ISW y servicios asociados y que dicha empresa dará empleo a 4 millones de personas. [18]

Brasil tiene una participación el mercado latinoamericano del 50 % representado por US $1.900 millones; seguido por México, con una participación del 17 %; y Argentina, con el 11% .En la lista Colombia que factura alrededor de US $190 millones al año y que el pasado año el sector de la Tecnologías de la Información en este país creció en un 11.5 % mas que en el 2006 y de esto un 10% gracias a la ISW, ocupa cuarto y quinto lugar Venezuela en Latinoamérica con respecto al software.[19]

En sentido general exciten grandes logros, sin embargo aun la industria presenta graves problemas como la carencia de personal capacitado. Ya que la falta de desarrolladores calificados en Argentina representa el mayor desafío para el crecimiento del 2008. La existencia de una escasez de gente calificada y la obtención de nuevos recursos para cumplir con la demanda fue el gran problema que se presentó durante el 2007, y que continuará el 2008.Esto no solo sucede en Argentina sino también en muchos otros países sobre todo en el área de América Latina. [20]

“Por otro lado, otro aspecto que afecta a la industria del software es la piratería, pues de acuerdo con información de la Alianza Internacional para la Propiedad Intelectual (IIPA, por sus siglas en inglés), en México dicha industria ha tenido pérdidas por 350 millones de dólares anuales. De hecho, la tasa de piratería en software se ubica en 65 por ciento. Algunos estudios han mostrado que si esa tasa se redujera 10 puntos porcentuales, a 55 por ciento, por ejemplo, ayudaría a la industria de software a crecer 50 por ciento.

Capítulo 1

Fundamentación Teórica

De acuerdo a los resultados obtenidos por un informe desarrollado recientemente por Business Software Alliance (BSA) e IDC, el 75% del software instalado en las computadoras argentinas sería ilegal, lo que representaría pérdidas por más de 300 millones de dólares para la industria del software.

Si bien la cifra refleja una caída de 2 puntos porcentuales con respecto a los valores registrados en 2005, el estudio revela que la tasa de piratería en dicho país volvió a superar a la de América latina, que cayó al 66% (2 puntos) y representa pérdidas de 3 mil millones de dólares para el sector.

El informe desarrollado por BSA e IDC recopila datos de 2006 en 102 países. En cuanto a las tendencias regionales, los países que incrementaron su nivel de uso de software ilegal en América Latina fueron Chile, Colombia, El Salvador, Panamá, República Dominicana y Venezuela, entre otros. Por su parte, Brasil redujo la tasa de piratería a un 60% (4 puntos porcentuales menos que en 2005), pese a lo cual la pérdida para la industria de ese país fue la mayor de la región, de 1.148 millones de dólares.

A nivel mundial**,** las pérdidas para la industria totalizaron alrededor de 40 mil millones de dólares y la tasa se ubicó en el 35%. Ese nivel se mantiene estable desde el inicio de la medición "en gran parte debido al incremento de la tasa en Asia, que compensó con creces los descensos en otras regiones", tal como describe el informe. No obstante, el estudio señala que "hubo avances en algunos mercados emergentes además de China, entre ellos, Rusia, donde la tasa cayó 7 puntos porcentuales en los últimos 3 años".

Otros de los países que presenta este problema es España que si se redujera el índice de piratería en 10 puntos –actualmente es del 46%-, se generarían casi 2.000 empleos adicionales, 1.400 millones de euros más en la facturación del sector, 196 millones de euros extra.

La piratería de software es un problema que afecta al sector en general y a la economía global. También afecta a las empresas legítimas que tienen que competir con las que venden software falsificado. La Business Software Alliance y la firma de investigación del mercado IDC pusieron de manifiesto en el estudio The Global Software Piracy Study (sólo en inglés) que la piratería de software produce unas pérdidas globales de casi 40.000 millones de dólares. Los usuarios gastan millones de dólares al año en copias falsificadas, financiando a estafadores que ponen en peligro la integridad de

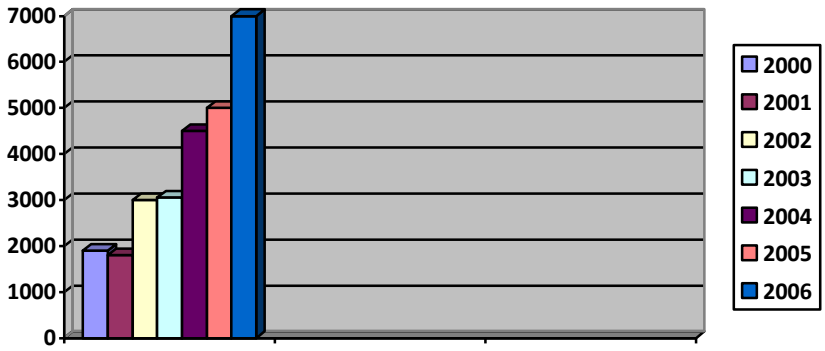
la industria de software y la seguridad de equipos domésticos y de empresas. La piratería de software supone un riesgo para todo el mundo. "[21]

Todo este auge en la piratería viene influenciado principalmente por el gran número de vulnerabilidades que se presentan hoy en día en le software y demás medios de informáticos. Las vulnerabilidades siempre han existido lo que en los últimos años han cobrado un mayor auge. Esto lo vemos reflejado en el exhaustivo trabajo de la investigación de X-Force de Sistemas de Seguridad de la Internet de la IBM y el equipo de desarrollo que estrechamente observaron y registraron las incidencias de vulnerabilidades desde el 2000 hasta el 2006 años donde comienza su crecimiento. La IBM ha estado catalogando, analizando e indagando las vulnerabilidades desde 1997. La base de datos de X-Force es lo más grande, con más de 30,000 vulnerabilidades de seguridad catalogadas. [22]

Con 7,247 vulnerabilidades reveladas en 2006, la cuenta total de vulnerabilidad aumentó casi 40 por ciento sobre el año anterior. Ha habido un 261 incremento de por ciento en vulnerabilidades, un promedio de 23 por ciento anualmente. Se espera que esta tendencia continúa a todo lo largo de 2007. [23]

El incremento desde 2000 al 2006 en las vulnerabilidades puede ser observado en la siguiente gráfica:

Año	Vulnerabilidades	% de incremento entre años
2000	2007	
2001	1918	-4.4%
2002	3210	67.4%
2003	3156	-1.7%
2004	4606	14.5%
2005	5195	12.8%
2006	7247	39.5%



Grafica 1.1 Auge de las vulnerabilidades desde 2000 a 2006.

Tipos de vulnerabilidades.

Dependiendo del origen de la vulnerabilidad, esta puede tener una o varias de las siguientes características: [24]

- **Error en validación de entrada:** Cuando la entrada que procesa un sistema no es comprobada adecuadamente de forma que una vulnerabilidad puede ser aprovechada por una cierta secuencia de entrada. Este tipo de vulnerabilidad y sus subcategorías solo se aplican a entradas formadas maliciosamente.
- **Desbordamiento de límites.** Ocurre cuando la entrada recibida por un sistema, sea de origen humano o máquina, hace que exceda los límites de funcionamiento normal y produzca una vulnerabilidad. Por ejemplo, el sistema se queda sin memoria, sin espacio en disco, o colapsar la red. También podrían ser variables del sistema mal controladas, que lleguen a su máximo valor y salte al mínimo, o forzar una división por cero no tratada.
- **Desbordamiento de buffer.** Sin duda la más clásica de las fuentes de vulnerabilidades. Se produce cuando la entrada de un sistema es mayor que el área de memoria asignada para contenerla (buffer) y el sistema no lo comprueba adecuadamente. Entonces el buffer de entrada "se desborda" y escribe en zonas de memoria contiguas. Construyendo inteligentemente el exceso de entrada, un atacante puede hacer caer el sistema e incluso ejecutar instrucciones de forma arbitraria.

- **Secuencias de comandos en sitios cruzados** (*cross-site scripting*). Se abrevia XSS o CSS, aunque esta última se puede confundir con las hojas de estilo en cascada (Cascading Style Sheet). Aplicable en principio a aplicaciones web o sitios web dinámicos, Consiste en que se pueda ejecutar código de un dominio desde otro dominio, de forma que se perjudica a otro usuario, no al sitio web directamente.

  Por ejemplo, supongamos un sitio web de subastas que en el mensaje de error de "página no encontrada" incluye la página pedida, sin filtrar. Un usuario malicioso puede poner un enlace a una página inexistente y con código (java script, VBScript, ActiveX o cualquiera que se pueda ejecutar en el ordenador de la víctima) de forma que en el mensaje de error se incruste ese código, que se ejecuta en el navegador de la víctima. Por ejemplo puede redirigirla a una falsificación del sitio de subastas y capturar su nombre de usuario y contraseña.

- **Error de validación de acceso.** Se produce cuando el mecanismo de control de acceso es defectuoso.
- **Error de manejo de condición excepcional.** Se produce cuando el sistema se vuelve vulnerable cuando se produce una condición de funcionamiento no habitual, por ejemplo errores en la red, que el sistema no maneja adecuadamente.
- **Error de entorno.** Una vulnerabilidad se caracteriza de esta manera si el entorno en que un sistema está instalado de alguna manera hace al sistema vulnerable. Esto puede ser debido, por ejemplo, a una interacción no prevista entre una aplicación y el sistema operativo, o entre dos aplicaciones corriendo en el mismo anfitrión. Este sistema probablemente es perfectamente seguro en las pruebas que ha hecho el desarrollador, pero en el entorno en que se ha instalado de alguna manera no cumple las condiciones de seguridad supuestas.
- **Error de configuración.** Si la configuración controlable por el usuario es tal que el sistema es vulnerable. La vulnerabilidad no es debida al diseño del sistema si no a como el usuario final configura el sistema. También se considera error de este tipo cuando la configuración por defecto del sistema es insegura, por ejemplo una aplicación recién instalada.
- **Condición de carrera.** Se produce cuando la no atomicidad de una comprobación de seguridad causa la existencia de una vulnerabilidad. Por ejemplo un sistema comprueba si una operación es válida es permitida por el modelo de seguridad y luego la ejecuta. Sin embargo, en el tiempo que pasa desde que se hace la comprobación hasta que se ejecuta la operación

las condiciones cambian de forma que la operación ya no es válida. Un atacante puede aprovechar esta pequeña ventana de oportunidad y hacer a un sistema efectuar operaciones no válidas, como escribir en un fichero de contraseñas.

- **Error de diseño.** Una vulnerabilidad se caracteriza como "error de diseño" si no hay errores en la implementación ni en la configuración de un sistema, si no que el diseño inicial es erróneo.
- **Otros.** Cuando se encuentra una vulnerabilidad que no cae en ninguna de las categorías anteriores.

De todas estas vulnerabilidades anteriormente mencionadas las más comunes son en el 2005 y 2006, el “cross-site-scripting” (XSS) era el número 1, y el “SQL Injection” era el número 2. El “PHP remote file inclusión” es el número 3 del 2006. Para la categoría de sistemas operativos Los buffer overflow siguen siendo el número 1 según lo divulgado por asesores de ventas de sistemas operativos (SO). XSS sigue siendo alto en esta categoría, el número 2 de 2005 y número 3 de 2006.

Además de estas, también existe la categoría de las vulnerabilidades más comunes para la Web que son:

Las 10 vulnerabilidades más comunes de una aplicación Web.

La comunidad pro-código abierto OWASP (Open Web Application Security Project) ha publicado un informe en el que se detallan los diez principales problemas de seguridad que se encuentran hoy día en las aplicaciones Web. [25]

“El OWASP, un grupo que promueve el software de código libre y ayuda a las organizaciones a comprender y mejorar la seguridad de sus aplicaciones y servicios Web, ha publicado un decálogo con las principales vulnerabilidades que se encuentran en Internet.

El OWASP intenta enfatizar que la seguridad en Internet no depende exclusivamente de la seguridad de las redes, sino de las propias aplicaciones. El informe relata que las vulnerabilidades son sorprendentemente comunes y pueden ser explotadas por atacantes no sofisticados con herramientas fácilmente disponibles.

La guía de vulnerabilidades publicada por OWASP comprende aspectos claves para facilitar el desarrollo de aplicaciones Web, estableciendo los principios básicos de seguridad que cualquier aplicación o servicio Web debe cumplir: "[26]

1. **Entrada no validada.** La información de entradas Web no es validada antes de ser usadas por la aplicación Web. Los agresores pueden usar estas fallas para atacar los componentes internos a través de la aplicación Web.
   Tipo de datos (strings, integer, real, etc…)
   - ✓ Conjunto de caracteres permitidos
   - ✓ Longitud mínima y máxima
   - ✓ Si nulo es permitido
   - ✓ Si el parámetro es requerido o no
   - ✓ Si los duplicados son permitidos
   - ✓ El rango numérico
   - ✓ Valores específicos permitidos (enumeración)
   - ✓ Patrones específicos (expresiones regulares)
2. **Control de Acceso Interrumpido.** Las restricciones de aquello que tienen permitido hacer los usuarios autentificados no se cumplen correctamente. Los agresores pueden explotar estas fallas para acceder a otras cuentas de usuarios, ver archivos sensitivos o usar funciones no autorizadas.
3. **Administración de Autenticación y Sesión Interrumpida.** Las credenciales de la cuenta y los tokens de sesiones no están propiamente protegidos. Los agresores que pueden comprometer las contraseñas, claves, cookies de sesiones u otro token, pueden vencer las restricciones de autenticación y asumir la identidad de otros usuarios.
4. **Fallas de Cross Site Scripting (XSS).** La aplicación Web puede ser usada como un mecanismo para transportar un ataque al navegador del usuario final. Un ataque exitoso puede comprometer el token de sesión del usuario final, atacar la maquina local o enmascarar contenido para engañar al usuario.
   - ➢ Validación de los scripts de salida.
   - ➢ Uso de correo electrónico

5. **Desbordamiento del Búfer.** Los componentes de aplicaciones Web en ciertos lenguajes que no validan adecuadamente las entradas de datos pueden ser derribados y, en algunos casos, usados para tomar control de un proceso. Estos componentes pueden incluir CGI, bibliotecas, rutinas y componentes del servidor de aplicación Web.
    - Staks de los componentes
    - No ambiente Java
6. **Fallas de Inyección.** La aplicación Web puede pasar parámetros cuando accede a sistemas externos o al sistema operativo local. Si un agresor puede incrustar comandos maliciosos en estos parámetros, el sistema externo puede ejecutar estos comandos por parte de la aplicación Web. Llamadas a:
    - System.
    - Exec.
    - Fork.
    - Runtime.exec.
    - Solicitudes SQL.
7. **Manejo Inadecuado de Errores.** Condiciones de error que ocurren durante la operación normal que no son manejadas adecuadamente. Si un agresor puede causar que ocurran errores que la aplicación Web no maneja, éste puede obtener información detallada del sistema, denegar servicios, causar que mecanismos de seguridad fallen o tumbar el servidor.
8. **Almacenamiento Inseguro.** Las aplicaciones Web frecuentemente utilizan funciones de criptografía para proteger información y credenciales. Estas funciones y el código que integran a ellas han sido difíciles de codificar adecuadamente, lo cual frecuentemente redunda en una protección débil.
    - Fallar al encriptar información crucial.
    - Almacenamiento inseguro de llaves, certificados y contraseñas.
    - Almacenamiento incorrecto de secretos en memoria.
    - Fuentes pobres de aleatorización.
    - Elección pobre de algoritmo.
    - Intentar inventar el nuevo algoritmo de encriptación.
    - Fallar al incluir soporte para cambios en las llaves de encriptación
9. **Negación de Servicio.** Los agresores pueden consumir los recursos de la aplicación Web al punto de que otros usuarios legítimos no puedan ya acceder o usar la aplicación. Los

agresores también pueden dejar a los usuarios fuera de sus cuentas y hasta causar que falle una aplicación entera.

10. **Administración de Configuración Insegura.**

Tener una configuración de servidor estándar es crítico para asegurar una aplicación Web

Estos servidores tienen muchas opciones de configuración que afectan la seguridad y no son seguros desde la instalación original del software.

- Fallas de seguridad no parchadas en el software del servidor
- Fallas de seguridad en el software del servidor o malas configuraciones que permiten ataques de listado de directorio o cross directory.Innecesarios archivos por defecto, de respaldo o de ejemplo, incluyendo Scripts, aplicaciones, archivos de configuración y páginas Web.
- Permisos no adecuados en archivos y directorios.
- Servicios innecesarios habilitados, incluyendo manejo de contenido y administración remota.
- Cuentas por defecto con contraseñas por defecto.
- Funciones administrativas o de depuración que son habilitadas o accesibles.
- Certificados SSL y opciones de encriptación mal configurados.
- Uso de certificados por defecto.
- Autenticación inadecuada con sistemas externos.

Debido a lo expuesto anteriormente cabe preguntarse **¿Cómo se encuentra la seguridad informática en el mundo y sobre todo la seguridad en el software?**

1.4.1 La seguridad informática en el mundo.

Tal ha sido el desarrollo de lo que hoy entendemos por seguridad informática que ya no sólo se habla de protección de equipos informáticos ante virus y gusanos, cifrado de la información o vulnerabilidades en redes de comunicaciones.[27]

“Hoy en día, una entidad que trabaje con cualquier tipo de entorno informático, desde pequeñas empresas con negocios no relacionados directamente con las nuevas tecnologías hasta grandes telcos de ámbito internacional, está - o debería estar - preocupada por su seguridad. Y no es para menos: el número de amenazas a los entornos informáticos y de comunicaciones crece casi exponencialmente

año tras año, alcanzando cotas inimaginables hace apenas una década. Y con que el futuro de la interconexión de sistemas sea tan solo la mitad de prometedor de lo que nos tratan de hacer creer, es previsible que la preocupación por la seguridad vaya en aumento conforme nuestras vidas estén más y más 'conectadas' a Internet.

Hoy en día la seguridad va más allá de lo que pueda ser un cortafuego, un sistema de autenticación biométrico o una red de censores de detección de intrusos: ya se contemplan aspectos que hasta hace poco se reservaban a entornos altamente cerrados, como bancos u organizaciones militares. Y es que nos hemos empezado a dar cuenta de que tan importante o más como un buen *firewall* es un plan de continuidad del negocio en caso de catástrofe. "[28]

La seguridad informática es clave hoy para el buen funcionamiento de las empresas. El 100% de las empresas hace copias de seguridad y un 97% tiene medidas de protección de accesos y de comunicaciones. [29]

"Los expertos, en primer término, han revelado que el 75,1% de las empresas dispone de una función encargada de asumir las responsabilidades de seguridad, ubicada en su mayoría en el departamento de informática. Y que un 10,4% de las compañías ha contratado apoyo de empresas externas.

En seguridad informática, como en otros ámbitos, Finanzas, Seguros y Bienes Raíces, donde el 28,4% de las empresas reconocen disponer de un departamento para gestionarla, es el sector más preparado. Y Agricultura, Ganadería, Minería y Pesca el que registra una menor especialización.

Un 76,3% de las empresas tiene entre uno y tres empleados dedicados a esta tarea de seguridad; pero el número de personas dedicadas a la misma no guarda relación con el tamaño de la plantilla. Casi el 70% de las empresas dispone de una política que recoge los principales preceptos de seguridad. Así, el 100% de las empresas dispone de una política para realizar copias de seguridad, el 96,8% documenta aspectos relacionados con el control de accesos, el uso de contraseñas (96%) y la protección de comunicaciones (96,8%). El aspecto internamente menos regulado es el relativo a la identificación de riesgos que afecten a los activos de la empresa (67,1%).

El 85,3% de las empresas indican que su situación de seguridad ha mejorado en el último año, y un 69,3% declara no haber tenido incidencias de seguridad destacables. En cuanto a los sectores más

castigados por incidencias destacan Construcción y Contratas con un 37,5%, y Finanzas, Seguros y Bienes y Raíces con un 38,1%. "[30]

Hasta tal punto se ha popularizado el mundo de la seguridad que surgen empresas `especializadas' en todos los ámbitos de la seguridad tales como:

MacroSeguridad Latino América: comercializa productos de seguridad basados en hardware para VPN, firmas digitales, y transporte de certificados digitales; además, provee soluciones completas para protección de software contra copias ilegales, protección de datos, administración de licencias y seguridad para Internet. [31]

Deloitte: Proveen soluciones integrales de seguridad, desde el diagnóstico hasta el diseño e implementación de modelos que permitan la confidencialidad de la información como un proceso de negocio. [32]

Digiware: Apoyar a las organizaciones a nivel nacional e internacional en la protección de la información y los sistemas que la soportan por medio de la prestación de servicios especializados de seguridad informática sustentados en una sólida base de conocimiento, calidad y experiencia.[33]

ETEK International Holding: "Fundada en 1974, ETEK International tiene oficinas en Argentina, Brasil, Chile, Colombia y su oficina principal en Estados Unidos. La oficina de Colombia fue establecida en 1995 y desde entonces se ha posicionado como líder en soluciones de Seguridad de la Información.

Para garantizar que sus diseños y procedimientos siguen un estándar riguroso, ETEK es una compañía certificada ISO 9001 y para garantizar la seguridad de la información propia y de clientes, ETEK es una compañía certificada ISO/IEC 27001: 2005." [34]

Symantec: Es un proveedor líder global de software, dispositivos y servicios para ayudar a personas, así como a pequeñas, medianas y grandes empresas a garantizar la seguridad, la disponibilidad y la integridad de su bien más preciado: la información. [35]

“Cada vez es más habitual que las empresas contraten los servicios de seguridad de una compañía externa, especializada en la materia, y que permita olvidarse relativamente, al personal de esa empresa de los aspectos técnicos y organizativos de la seguridad, para poder centrarse así en su línea de negocio correspondiente; esta política es lo que se conoce como ***outsourcing*** y se intenta traducir por `externalización'.

Todas estas empresas de seguridad son en su mayoría muy eficientes en su labor pero muchos expertos consideran que el *outsourcing* presenta *a priori* graves inconvenientes, y quizás el más importante sea el de dejar toda nuestra seguridad en manos de desconocidos, por muy buenas referencias que podamos tener de ellos.

Otros tipos de problemas a tener también en cuenta; uno de ellos es justamente el límite de uno de los beneficios de esta política: ya que la externalización permite a una empresa `despreocuparse' de su seguridad, podemos encontrar el caso - nada extraño - de un excesivo `despreocupamiento'. Actualmente, el abanico de servicios que ofrece cualquier consultora de seguridad suele abarcar desde auditorias puntuales hasta una delegación total del servicio pasando por todo tipo de soluciones intermedias, y lo que justifica la elección de un modelo u otro es un simple análisis de riesgos: el riesgo de la solución externalizada ha de ser menor que el nivel de riesgo existente si se gestiona la seguridad de forma interna. En cualquier caso, al externalizar se suele introducir una cierta pérdida de control directo sobre algunos recursos de la compañía, y cuando esa pérdida supera un umbral nos encontramos ante un grave problema; en ningún caso es recomendable un desentendimiento total de los servicios externalizados, y el contacto e intercambio de información entre las dos organizaciones (la contratante y la contratada) han de ser continuos y fluidos. “[36]

Otras empresas desarrolladoras de software sin embargo toman la decisión de realizar la seguridad de su institución y sus productos internamente.

1.4.2 La seguridad en el software en el mundo.

Históricamente se han abordado los problemas de Seguridad en la red y en la información con herramientas como Cortafuegos, Antivirus, Detectores de Intrusos, Políticas de seguridad etc... Esto

es hacer seguridad reactiva, ya que se reaccionan a los problemas que surgen en este medio (Intranet, Internet, etc.) y no resuelve el problema original, que es el Software o aplicación mal programada.

Seguridad en el Software es la mejor forma de hacer seguridad preventiva, hace que la probabilidad de encontrar “agujeros” o fallas de programación en el software sea casi nula y a su vez no pueda haber robo de información y otros delitos telemáticos.

Seguridad en el Software busca reducir de forma preventiva el riesgo de sufrir un ataque o el riesgo de perder información de vital importancia para las organizaciones a través de los agujeros en el software.

Los agujeros en el software son utilizados de forma directa o indirecta para ganar acceso no autorizado o como punto para atacar a otros ordenadores. Pero no sólo el software o aplicaciones informáticas pueden tener “agujeros” de seguridad, también los Sistemas Operativos (Windows, Linux, Unix, Solaris, etc.), Hardware Informáticos, etc. Esto se debe básicamente a descuidos o fallos en su construcción. La mejor forma de evitar estos descuidos es adoptando estándares y buenas conductas a lo largo del proceso de construcción.

La mayoría de las empresas de software invierten en la corrección de los errores en la etapa final de los productos .Hasta ahora la única conocida que se encuentra implementando otra forma de realizar la seguridad en el software es Microsoft con el ciclo de vida de desarrollo de seguridad (SDL, Security Development Lifecycle) de Trustworthy Computing (computación confiable). Este proceso incorpora varias actividades y materiales relacionados con la seguridad a cada una de las fases del proceso de desarrollo de software de Microsoft.

Estas actividades y materiales incluyen el desarrollo de modelos de amenazas durante el diseño de software, el uso de herramientas de exploración del código de análisis estático durante la implementación y la realización de revisiones del código y pruebas de seguridad durante una "campaña de seguridad". Antes del lanzamiento de software sometido al SDL, un equipo independiente del grupo de desarrollo debe realizar una revisión final de seguridad. En comparación con el software que no se ha sometido al SDL, el software que ha seguido este proceso ha presentado una reducción considerable en el número de detección externa de vulnerabilidades de seguridad.

1.5 La Seguridad del Software en Cuba.

"Cuba es un país donde el desarrollo del software es aun incipiente. Es por ello que una de las principales tareas del Gobierno Cubano es desarrollar la Industria del Software .Lo que ha traído consigo la creación de variadas estrategias con el fin de elevar la producción y la calidad del software cubano, logrando así una seguridad en el mismo.

En el año 2001, existían en el Ministerio de la Informática y las Comunicaciones (MIC) 24 empresas o entidades que tenían en su objeto social la producción del SW y servicios informáticos, esta estructura sufrió diversas reorganizaciones. Actualmente hay 8 entidades en el MIC que desarrollan esta actividad.

La Industria Cubana del Software (ICSW) está llamada a convertirse en una significativa fuente de ingresos para el país, como resultado del correcto aprovechamiento de las ventajas del alto capital humano disponible.

La promoción de la industria cubana del software en el ámbito internacional ha tenido como línea estratégica aprovechar la enorme credibilidad que tiene Cuba en sectores tales como la salud, la educación y el deporte. El continuar la producción sostenida de software de alta calidad en prestaciones, imagen y soporte, para satisfacer las necesidades nacionales en estos sectores, tendrá una positiva repercusión en el incremento de la exportación.

La ICSW ha alcanzado notables logros. Se han desarrollado software para todos los aspectos de la economía cubana, incluyendo las telecomunicaciones, educación y el mayor logro se ha obtenido en el área de equipamiento medico, donde el producto ha demostrado eficacia incrementando la calidad de la atención de la salud en el país y han tenido aceptación en el país." [37]

Para fomentar el desarrollo de la industria se prioriza en el país, la enseñanza masiva de la computación en todo el sistema educacional, se cuenta con 26 politécnicos de informática, parte de los cuales son nuevos o fueron completamente remozados, donde se preparan alrededor de 40 mil técnicos medios en informática, a los que se unen mas de 11 800 estudiantes a nivel superior, de ellos 10 mil de la Universidad de la Ciencias Informáticas (UCI).

Para lograr el desarrollo en la ISW cubana se tiene que tomar en cuenta la seguridad del mismo, aspecto fundamental para un buen posicionamiento en el mercado mundial y en el cual no estamos muy desarrollados aún .En Cuba solo hay una empresa que se dedica a la seguridad, pero realiza seguridad reactiva principalmente ya que ofrece servicios como:

- Diagnóstico Externo.
- Certificados Digitales.
- Planes de Seguridad y Contingencia Informática.
- Soporte Técnico.
- Dictámenes.
- Custodia de material informático.
- Adiestramientos en Seguridad Informática.
- Dispositivos para la seguridad física de la PC.

Los cuales basan su funcionamiento en la seguridad de la información del software física, es decir desde fuera no desde la hora de elaboración del mismo. Otra de las acciones que Cuba toma con respecto al tema es la creación de empresas de calidad como; La Empresa de producción y Desarrollo de Software de Calidad (SOFTCAL), Grupo Nacional de Expertos en Calidad del Software (GNECS), Laboratorio Nacional de Certificación de la Calidad del Software (CALISOFT), las cuales su función principal es la obtención de software de calidad en la región, patentizando con ello la seguridad del mismo.

1.6 La seguridad del software en la UCI

La UCI pretende ser la vanguardia del desarrollo de las empresas de software en Cuba y de llevar la informatización a todos los sectores de la sociedad: Salud, Educación, Cultura, Deporte, Turismo, Prensa, etc. Regir y propiciar un avance tecnológico y de la industria del software en Cuba y convertir la industria del software en un renglón fundamental de la economía e insertarnos en el mercado internacional, por lo que el reto de la universidad es producir software de alta calidad, logrando así un software con una alta seguridad. [38]

El aspecto de la seguridad en el software es todavía un aspecto muy débil en la UCI ya que realiza como la mayoría de las empresas en el mundo de software seguridad reactiva y no se realiza la seguridad preventiva, desde el comienzo mismo de la elaboración del software. La UCI cuenta con un grupo de calidad a nivel central, como también uno en cada facultad, los procesos de calidad que se llevan a cabo en cada uno de estos grupos no son suficientes para lograr que los productos obtengan la seguridad requerida.

1.7 La seguridad del software en la Facultad 7.

La Facultad 7 es una de las 10 facultades de la UCI, donde se realizan proyectos de gran importancia para el país como en todas las demás facultades, pero principalmente en el sector de la salud. La misma esta constituida por dos polos de producción, los cuáles son el Polo de Salud y el Polo de Imágenes, ellos a su vez están compuestos por Áreas Temáticas. El Polo de Salud esta organizado por las Áreas Temáticas de Sistema de apoyo a la Salud (SAS), Hospitales, Atención Primaria de la Salud (APS), Especialidades. El Polo de Imágenes presenta el Área Temática de Software Medico e Imageneologia.

Cada una de estas Áreas Temáticas esta compuesta por los diferentes proyectos y estos a su vez los diferentes módulos en producción, entre ellas se encuentran el Área Temática de SAS el proyecto de Estadística, Docencia, Balance Material y Colaboración, en Especialización encontramos el proyecto de Nefrología, SIUM, CSI etc.

Todos estos proyectos presentan una seguridad reactiva, pero no una seguridad preventiva ya que no presentan un procedimiento que desde el inicio mismo de la elaboración de los módulos les permita obtener una seguridad en cada etapa de desarrollo, todos pasan por una etapa de calidad lo mismo dentro del proyecto que cuando concluye el módulo por el grupo de calidad de la facultad , pero esto no es suficiente ya que con estos procesos que se llevan a cabo no hay una seguridad óptima del producto.

1.8 Estándares Internacionales para lograr la seguridad en el software.

Hablar de estándares para lograr la seguridad del software es muy complejo ya que este tema no se ha tratado con profundidad, no obstante se han logrado grandes avances en el tema como son: [39]

Estándar ISO 17799 y 27001 basado en las Tecnología de la Información – Técnicas de seguridad – Código para la práctica de la gestión de la seguridad de la Información.

Este estándar ISO 17799 contiene 11 cláusulas de control de seguridad conteniendo colectivamente un total de 39 categorías de seguridad principales y una cláusula introductoria que presenta la evaluación y tratamiento del riesgo.

La serie 27000

A semejanza de otras normas ISO, la 27000 es realmente una serie de estándares. Los rangos de numeración reservados por ISO van de 27000 a 27019 y de 27030 a 27044.

- **ISO 27000:** En fase de desarrollo; su fecha prevista de publicación es Noviembre de 2008. Contendrá términos y definiciones que se emplean en toda la serie 27000. La aplicación de cualquier estándar necesita de un vocabulario claramente definido, que evite distintas interpretaciones de conceptos técnicos y de gestión. Esta norma está previsto que sea gratuita, a diferencia de las demás de la serie, que tendrán un coste.

- **ISO 27001:** Publicada el 15 de Octubre de 2005. Es la norma principal de la serie y contiene los requisitos del sistema de gestión de seguridad de la información. Tiene su origen en la BS 7799-2:2002 y es la norma con arreglo a la cual se certifican por auditores externos los SGSI de las organizaciones. Sustituye a la BS 7799-2, habiéndose establecido unas condiciones de transición para aquellas empresas certificadas en esta última.

- **ISO 27002:** Desde el 1 de Julio de 2007, es el nuevo nombre de ISO 17799:2005, manteniendo 2005 como año de edición. Es una guía de buenas prácticas que describe los objetivos de control y controles recomendables en cuanto a seguridad de la información. No es certificable. Contiene 39 objetivos de control y 133 controles, agrupados en 11 dominios. Como se ha

mencionado en su apartado correspondiente, la norma ISO27001 contiene un anexo que resume los controles de ISO 27002:2005.

- **ISO 27003:** En fase de desarrollo; su fecha prevista de publicación es Mayo de 2009. Consistirá en una guía de implementación de SGSI e información acerca del uso del modelo PDCA y de los requerimientos de sus diferentes fases.

- **ISO 27004:** En fase de desarrollo; su fecha prevista de publicación es Noviembre de 2008. Especificará las métricas y las técnicas de medida aplicables para determinar la eficacia de un SGSI y de los controles relacionados. Estas métricas se usan fundamentalmente para la medición de los componentes de la fase "Do" (Implementar y Utilizar) del ciclo PDCA.

- **ISO 27005:** En fase de desarrollo; su fecha prevista de publicación es Mayo de 2008. Consistirá en una guía de técnicas para la gestión del riesgo de la seguridad de la información y servirá, por tanto, de apoyo a la ISO27001 y a la implantación de un SGSI.

- **ISO 27006:** Publicada el 1 de Marzo de 2007. Especifica los requisitos para la acreditación de entidades de auditoría y certificación de sistemas de gestión de seguridad de la información. Es una versión revisada de EA-7/03 (Requisitos para la acreditación de entidades que operan certificación/registro de SGSIs) que añade a ISO/IEC 17021 (Requisitos para las entidades de auditoría y certificación de sistemas de gestión) los requisitos específicos relacionados con ISO 27001 y los SGSIs. Es decir, ayuda a interpretar los criterios de acreditación de ISO/IEC 17021 cuando se aplican a entidades de certificación de ISO 27001, pero no es una norma de acreditación por sí misma.

- **ISO 27011:** En fase de desarrollo; su fecha prevista de publicación es Enero de 2008. Consistirá en una guía de gestión de seguridad de la información específica para telecomunicaciones, elaborada conjuntamente con la ITU (Unión Internacional de Telecomunicaciones).

- **ISO 27031:** En fase de desarrollo; su fecha prevista de publicación es Mayo de 2010. Consistirá en una guía de continuidad de negocio en cuanto a tecnologías de la información y comunicaciones.

- **ISO 27032:** En fase de desarrollo; su fecha prevista de publicación es Febrero de 2009. Consistirá en una guía relativa a la ciberseguridad.

- **ISO 27033:** En fase de desarrollo; su fecha prevista de publicación es entre 2010 y 2011. Es una norma consistente en 7 partes: gestión de seguridad de redes, arquitectura de seguridad de redes, escenarios de redes de referencia, aseguramiento de las comunicaciones entre redes mediante gateways, acceso remoto, aseguramiento de comunicaciones en redes mediante VPNs y diseño e implementación de seguridad en redes.

- **ISO 27034:** En fase de desarrollo; su fecha prevista de publicación es Febrero de 2009. Consistirá en una guía de seguridad en aplicaciones.

- **ISO 27799:** En fase de desarrollo; su fecha prevista de publicación es 2008. Es un estándar de gestión de seguridad de la información en el sector sanitario aplicando ISO 17799 (actual ISO 27002).

Beneficios de la serie de ISO 27000

- Establecimiento de una metodología de gestión de la seguridad clara y estructurada.
- Reducción del riesgo de pérdida, robo o corrupción de información.
- Los clientes tienen acceso a la información a través medidas de seguridad.
- Los riesgos y sus controles son continuamente revisados.
- Confianza de clientes y socios estratégicos por la garantía de calidad y confidencialidad comercial.
- Las auditorías externas ayudan cíclicamente a identificar las debilidades del sistema y las áreas a mejorar.
- Posibilidad de integrarse con otros sistemas de gestión (ISO 9001, ISO 14001, OHSAS 18001…).
- Continuidad de las operaciones necesarias de negocio tras incidentes de gravedad.
- Conformidad con la legislación vigente sobre información personal, propiedad intelectual y otras.
- Imagen de empresa a nivel internacional y elemento diferenciador de la competencia.

- Confianza y reglas claras para las personas de la organización.
- Reducción de costes y mejora de los procesos y servicio.
- Aumento de la motivación y satisfacción del personal.
- Aumento de la seguridad en base a la gestión de procesos en vez de en la compra sistemática de productos y tecnologías.

ISO (Organización Internacional para la Estandarización)

"La **Organización Internacional para la Estandarización** o *International Organization for Standardization*, que nace después de la segunda guerra mundial (fue creada en 1946), es el organismo encargado de promover el desarrollo de normas internacionales de fabricación, comercio y comunicación para todas las ramas industriales a excepción de la eléctrica y la electrónica. Su función principal es la de buscar la estandarización de normas de productos y seguridad para las empresas u organizaciones a nivel internacional. La ISO es una red de los institutos de normas nacionales de 157 países, sobre la base de un miembro por el país, con una Secretaría Central en Ginebra, Suiza, que coordina el sistema. La Organización Internacional de Normalización (ISO), con base en Ginebra, Suiza, está compuesta por delegaciones gubernamentales y no gubernamentales subdivididos en una serie de subcomités encargados de desarrollar las guías que contribuirán al mejoramiento ambiental. Las normas desarrolladas por ISO son voluntarias, comprendiendo que ISO es un organismo no gubernamental y no depende de ningún otro organismo internacional, por lo tanto, no tiene autoridad para imponer sus normas a ningún país.

Es una organización internacional no gubernamental, compuesta por representantes de los organismos de normalización nacionales, que produce normas internacionales industriales y comerciales. Dichas normas se conocen como **normas ISO** y su finalidad es la coordinación de las normas nacionales, en consonancia con el Acta Final de la Organización Mundial del Comercio, con el propósito de facilitar el comercio, facilitar el intercambio de información y contribuir con unos estándares comunes para el desarrollo y transferencia de tecnologías. "[40]

1.9 Paradigma de programación para la seguridad del software.

"Un **paradigma de programación** representa un enfoque particular o filosofía para la construcción del software. No es mejor uno que otro sino que cada uno tiene ventajas y desventajas. También hay situaciones donde un paradigma resulta más apropiado que otro.

Algunos ejemplos de paradigmas de programación:

El paradigma imperativo o por procedimientos es considerado el más común y está representado, por ejemplo, por el C o por BASIC.

El paradigma funcional está representado por la familia de lenguajes LISP (en particular Scheme), ML o Haskell.

El paradigma lógico, un ejemplo es PROLOG.

El paradigma orientado a objetos. Un lenguaje completamente orientado a objetos es Smalltalk.

En la actualidad se esta llevando a cabo un nuevo paradigma de programación:

La programación orientada a aspectos (POA), el cual nos permite encapsular requerimientos o conceptos típicamente no funcionales, como la seguridad, se convierte en una herramienta atractiva para el desarrollo de software seguro. La POA nos permitiría encapsular las políticas de seguridad de forma separada e independiente del resto del sistema, con las consecuentes ventajas que esto implica:

- Mayor reusabilidad.
- Mejoras sustanciales respecto al mantenimiento, modificaciones, etc.
- Mayor grado de especialización.
- Menor complejidad del sistema.

Vemos que si bien la POA no agrega por si misma seguridad a sus aplicaciones, nos brinda un entorno ideal para incorporar el concepto de seguridad de forma coherente y natural en el desarrollo de software. Una de las reglas generales que gobiernan la correcta implementación de la seguridad es que la misma debe aplicarse correcta y consistentemente a través de todo el sistema. Esto se logra

naturalmente con este nuevo paradigma. Este nuevo paradigma es considerarlo como la alternativa más viable y prometedora en pos de lograr software seguro a nivel mundial.” [41]

1.10 Conclusiones.

El presente capitulo arrojo como conclusiones que:

- La Industria del Software es un mercado que crece día a día, perfeccionándose cada vez más, incrementándose nuevas empresas productores de software con grandes potenciales.
- Entre los componentes para el éxito del desarrollo del software esta producir software con una alta seguridad, logrando así mitigar las vulnerabilidades, piratería, etc y conseguir así certificar todos los productos con las normas implementadas para la seguridad.
- La seguridad del software es una actividad de protección, que se debe llevar a cabo en todo el entorno de desarrollo de cualquier producto; y debe existir en cualquier organización que este encaminada a obtener una producción con la calidad requerida.

Capítulo 2: Caracterización y diagnóstico a los proyectos productivos de la facultad 7.

2.1 Introducción.

En este capitulo se realiza una caracterización de la UCI, donde se analiza como se lleva a cabo la principal tarea en la universidad que es la producción del software , específicamente de la facultad 7 la cual es el objetivo de estudio de la investigación .Para dar cumplimiento a estos objetivos se realiza un diagnostico con vista a detectar las necesidades e insuficiencia que presenta la producción del software desde se fase inicial , donde se emplearon diferentes técnicas como es la entrevista a diferentes personas relacionadas con la producción del software en la faculta y la universidad , revisión de documentos internos , encuesta a los principales integrantes de los proyectos de la facultad 7 , entre otras ; lo que facilita detectar las deficiencias que presenta la producción del software en la facultad, relacionadas con la seguridad del productos que se desarrolla , durante su elaboración y entrega .

2.2 Caracterización de la UCI.

“La Universidad de la Ciencias Informáticas es un programa de la Revolución en el marco de la Batalla de Ideas. Su primer curso se inicio el 23 de septiembre de 2002 .La UCI nació para socializar y multiplicar como nunca antes los estudios superiores de la informática en Cuba. Es una universidad atípica precisamente porque fue creada sobre la base del nuevo concepto de la universidad productiva, logrando una fuerte vinculación Universidad –Empresa. Está formada por estudiantes y profesores de las diferentes provincias y de todos los municipios del país y su misión fundamental es: formar profesionales, comprometidos con su Patria, altamente calificados en la rama de la informática; así como producir software y servicios informáticos, a partir de la vinculación estudio –trabajo como modelo de formación.

Está constituida por 10 facultades y en cada facultad hay alrededor de un promedio de 2000 estudiantes completando en septiembre de 2006 una matricula de 10 000 estudiantes alcanzando así el pasado año el quinto año y la primera graduación de la universidad. La UCI desde abril del 2007 cuenta con tres facultades regionales en el país, con una matricula de 982 estudiantes en el curso 2007/2008.

Capítulo 2

Caracterización y diagnóstico a los proyectos productivos de la facultad 7

En la UCI la producción es un problema social, político, y económico, el cien por ciento de los estudiantes y profesores están vinculados a la producción participando en proyectos de alto valor tanto para el mercado nacional, como internacional, se plasma la concepción de que la docencia se realice desde la producción. Para lograr esta tarea se asigna a cada facultad un perfil, dicho perfil se desarrollan varias áreas de proyectos con el objetivo de obtener un software que contribuye a la economía del país y al proceso de informatización de la sociedad cubana. (Ver figura 2.2)

La Universidad de las Ciencias Informáticas soporta su formación con un entorno virtual de aprendizaje donde el 100% de las asignaturas están montadas en esta plataforma .Esto ofrece una gran cantidad de recursos para el estudio continuo.

La UCI representa la visión del futuro de nuestro país y así lo manifestó nuestro comandante en jefe en su visita en el año 2002 al referirse a la universidad y a los profesionales que se formaran en la misma de la siguiente manera:"La UCI, seria mejor decir la informática, se convertirá en una poderosísima fuerza científica, económica e incluso político, para el país….ustedes son los profesionales mas importantes que vamos a formar en cuanto a perspectiva económica".

"Las producciones intelectuales serán el sustento fundamental de Cuba. La idea es convertir la informática en una de las ramas mas productivas y portadoras de recursos para la nación…."

"Nuestra sociedad será una sociedad de trabajadores intelectuales .Dando vueltas y meditando- no es mucho lo que todavía hemos pensado y profundizado - , parece ser que aquí esta la cantera principal."

"La UCI es una Universidad con alto nivel de flexibilidad, centro docente experimental, es centro docente productor".

"La idea es convertir la informática en una de las ramas mas productivas y apartadoras de recursos para la nación. Es el empleo a fondo de la inteligencia y el capital humano que tenemos y principalmente del que podemos crear casi como espina dorsal de la economía"." [42]

2.2.1 La producción del software en la UCI y en la Facultad 7.

La producción: proceso central, alrededor del cual se articulan los demás los procesos fundamentales de la universidad: formación de pre y posgrado, investigación, u obtención de grados y categorías científicas. En la UCI la formación de los estudiantes se hace desde la producción, para esto se diseña la vinculación del estudiante a los proyectos desde el primer año, en roles más sencillos, hasta quinto

curso ya en roles de alta complejidad .El objetivo fundamental de las producciones es obtener proyectos para la informatización del país, la informatización de la vida de la propia universidad, y la exportación; estos se desarrollan fundamentalmente en la UCI, no en las entidades clientes. Los proyectos se organizan, gestionan, ejecutan y dirigen en las Facultades, y estas asumen los proyectos según las áreas en que se hacen expertas; en cada facultad hay un Vicedecano de Producción, y especialistas al frente de los temas de Arquitectura y Calidad, que asesoran al Vicedecano, además funciona un Consejo de Producción. [43]

Figura 2.2 Relación de las áreas de proyecto de las diferentes facultades.

Caracterización y diagnóstico a los proyectos productivos de la facultad 7

La UCI es una universidad de todos .Un programa de la Revolución diseñada para lograr resultados concretos, actualmente la universidad produce para diferentes entidades y órganos del país los cuales están representados por las diferentes empresas. (Ver figura 2.3)

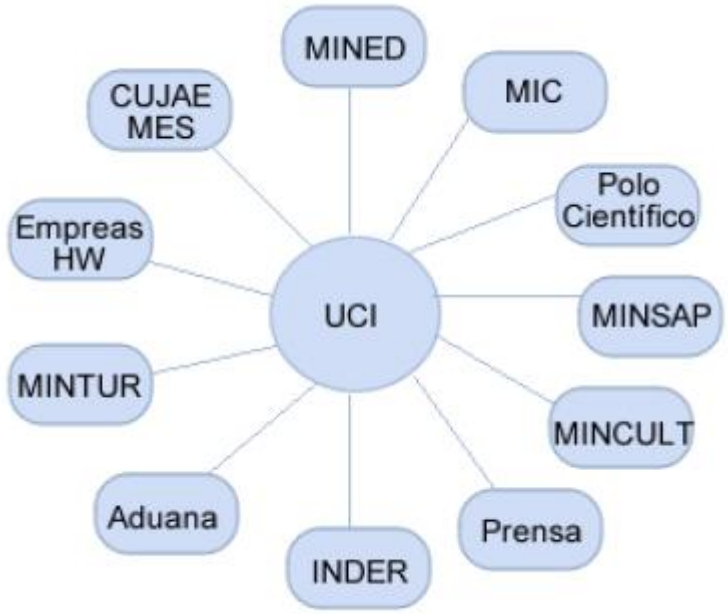


Figura 2.3 Principales entidades y empresas para las que se produce software en la universidad.

Para gestionar el tema de la producción se crea una Infraestructura Productiva (IP) como órgano metodológico, regulador, controlador y balancista, que presta servicios generales a todas las Facultades, la misma cuentan con 30 Polos Productivos que desarrollan proyectos temáticos.

La IP es la encargada de gestionar todos los procesos involucrados con la producción de software en la universidad esta estructurada por diferentes colectivo de dirección. Presenta 12 direcciones especializadas para coordinar y dar servicio al ciclo completo de las producciones de software que cuenta con el personal capacitado que trabaja en la documentación del proceso y lineamiento que rigen la producción de la universidad .Ellas son: Dirección Técnica, de Calidad, de Servicios Legales, de Comunicación Visual, de Auditoría de DVD y de Informatización que se encarga de convertir la universidad en una ciudad digital y 5 direcciones de Producción que coordinan el balance de la producción y los programas nacionales e internacionales de colaboración y cooperación para los sectores de: Salud , Educación , Deporte , cultura , Industrial , Tecnologías y Gestión Empresarial. [44]

Capítulo 2

Caracterización y diagnóstico a los proyectos productivos de la facultad 7

En sentido general las diferentes direcciones de producción en la universidad tienen los siguientes objetivos: [45]

- ✓ Proporcionar a las facultades metodologías, procesos y procedimientos definidos para llevar a cabo la ejecución de los proyectos productivos de la universidad.
- ✓ Definir y establecer indicadores y mecanismos de control a los proyectos.
- ✓ Efectuar chequeos y evaluaciones periódicas a los proyectos productivos de las facultades según los indicadores establecidos.
- ✓ Velar por la eficiente administración de los recursos humanos y materiales en cada proyecto.
- ✓ Participar en las negociaciones con los clientes que soliciten servicios de la universidad.
- ✓ Efectuar chequeos periódicos con los clientes para medir el avance del proyecto, el grado de satisfacción del cliente y la muestra de los resultados alcanzados.
- ✓ Brindar el servicio de consultoría a los proyectos en temas de Gestión de Proyectos Informáticos, Metodologías de Desarrollo y Arquitectura del Software.
- ✓ Impartir cursos de capacitación en los temas de Gestión de Proyectos Informáticos, Metodologías de Desarrollo y Arquitectura del Software.

La Dirección de Informatización de la Universidad de la Ciencias Informáticas, perteneciente a la Infraestructura Productiva, tiene como misión: dirigir, organizar, coordinar, chequear, diseñar y definir la Informatización de todos los procesos internos en cada una de las áreas que rigen la vida de la universidad , desde la perspectiva de una Ciudad Digital , logrando una total integración de todas las entidades , flujos y/o procesos ,basado en el funcionamiento armónico de la tecnología y los servicios informáticos. Proporcionando además el uso ordenado y masivo de las tecnologías con que cuenta esta importante institución .Convirtiéndose en el prototipo para la Informatización de la Sociedad Cubana.

La UCI participa o coordina más de 20 programas de informatización con ministerios o entidades nacionales. Se destacan los resultadas en el desarrollo de sistemas para la salud , educación , aduana , sistemas de identificación , automatización , prensa , bioinformática , procesamiento de imágenes y

señales , realidad virtual , geomántica , teleformación, estadística , entre otros. Además ha desarrollado una distribución (sistema operativo) de software libre que ya se encuentra instalada en varias instituciones del país, brindando además toda la asesoría y capacitación que requiera el personal.

En la IP se brindar diferentes servicios; los servicios tecnológicos son los que facilitan las tecnologías y la arquitectura de la misma, así como la estandarización y la arquitectura de información; por otro lado los servicios de calidad realizan las actividades de aseguramiento de la calidad a través de un grupo de personas que realizan diferentes auditorias y revisiones en el laboratorio de certificación; los servicios de informatización que tienen la misión de crear plataformas y soporte de software :así como los servicios de comunicación visual encargados de gestionar la información visualmente esta tarea la realizan fundamentalmente tres grupos :creativo , producción , realización y por ultimo también brinda servicios legales de gestión de la propiedad Intelectual así como asesoría especializada. La dirección de calidad de la UCI es la encargada de certificar el software desarrollado.

“Cada negocio es un nuevo modelo de proyecto; cualquier proyecto productivo esta compuesto por:

- Personal de la UCI y una empresa productora de SW (negocia al cliente).
- Expertos funcionales.
- Equipo de desarrollo (altamente comprometido).
- Condiciones logísticas y tecnologías adecuadas.

La encargada de la comercialización con otros países es la empresa de comercio exterior ALBET (alternativa bolivariana para la exportación de tecnologías); creada en el 2005 es una Sociedad Mercantil Cubana, de la UCI, representa la “cara comercial “para todos los proyectos de exportación coordinados a través de la Infraestructura Productiva y en el marco del Programa de la Revolución para la Producción de Software. Además integra esfuerzos de múltiples organizaciones y empresas para garantizar las soluciones tecnológicas integrales. “[46]

2.2.2 Evolución de la producción de software en la UCI.

Curso 2002-2003: No existen proyectos ni claridad de cómo iba a enfrentar la UCI el tema de la producción. Se estudian los Parques Tecnológicos. Se ganan una licitación en México, pero el proyecto no resulta dada la inmadurez de la organización. [47]

Curso 2003-2004: **"**Se buscan proyectos en todos los organismos del país. Gran dispersión de pequeños proyectos. La fuerza de estudiantes esta en primer y segundo año. No están claras las prioridades. Primeros intentos de organización metodológica y de una estructura organizativa para la producción .Experiencia de impacto en Venezuela.

Curso 2004-2005: Se crea la Vicerrectoría Primera para unir la atención a los procesos de formación y producción. Se crean en la IP direcciones de SW para la Salud, SW Educativo, Calidad, Exportación de SW, Comunicación Visual (Diseño). Los proyectos más importantes se organizan y dirigen desde la IP.Entidades productivas externas se ubican en la IP como por ejemplo por Softel. Se definen conceptos para la producción de la UCI. .Se comienzan grandes proyectos en el marco del Convenio de Colaboración Cuba-Venezuela.

Curso 2005-2006: Se consolida la IP, se definen las Facultades como ejecutoras de los proyectos y la IP como la estructura que se responsabiliza con el proceso productivo. La UCI enfatiza su papel como coordinadora y desarrolladora de proyectos en el marco de la Alternativa Bolivariana para las Américas (ALBA).Propuestas de alianzas y negocios. Se crea ALBET (Alternativa Bolivariana para la Exportación de Tecnologías).

Curso 2006 – 2007: Se encontraban en ejecución más de 150 proyectos y se propone el modelo de **Polo Productivo** como espacio natural para ejecutar proyectos temáticos. Se fortalece la colaboración con las entidades nacionales y surge el modelo de **Informatización Participativa** para ejecutar los programas nacionales de cooperación , donde la entidad dirige la ejecución del proyecto , designa los especialistas funcionales conocedores del tema a informatizar y la UCI pone a disposición sus recursos humanos especializados y su infraestructura técnica , destacando la responsabilidad de la UCI por la informatización de la entidad y con la formación del nuevo profesional.

Curso 2007-2008: Se prioriza el trabajo dirigido al fortalecimiento de la Soberanía Tecnológica a partir de la formación del personal, el desarrollo de tareas de investigación para crear bases tecnológicas libres para la realización de proyectos propios y la formulación de un modelo de colaboración y cooperación para los del ALBA. " [48]

2.2.3 Necesidad de Producción.

"Proceso productivo real: entendiendo esto no como la mera actividad de hacer el software (de programarlo) sino de asistir al conjunto de actividades que lo garantizan, que van desde un proceso

organizacional de división del trabajo por roles, desarrollo de base tecnológicas que implican enfrentarse a problemas desconocidos (al menos parcialmente), hasta el echo de lograr un trabajo integrado en equipo. Luego el estudiante no se enfrenta a ejercicios académicos, sino estos quedan superados por una práctica real, donde los problemas se corresponden a situaciones practicas determinados por la necesidad.

La formación desde la producción tiene como objetivo formar en los estudiantes las competencias necesarias para desempeñarse de manera eficaz en el desempeño de un rol dentro de un equipo de desarrollo de software.

Al mismo tiempo ayuda a que se consoliden las bases de la industria del software, propiciando: mecanismos de gestión del conocimiento, sistemas de certificación, cursos de capacitación ajustados a los intereses de los proyectos, etc.

El aprendizaje autónomo aparece en este contexto con mayor presencia, la necesidad de alcanzar una gran experiencia en poco tiempo hace necesario que se utilicen múltiples métodos de enseñanza: presénciales, semipresenciales y a distancia.

El profesor visto en un sentido amplio debe convertirse en un tutor, ya que las funciones del profesor en un grupo de proyecto superan tanto al profesor que esta detrás de una plataforma como el clásico profesor de un aula, esencialmente por las funciones educativas. En consecuencia el colectivo de profesores pasa a ser junto a las direcciones de las organizaciones el colectivo educativo por excelencia.

Es precisamente en el plano educativo y ético donde este modelo manifiesta todas sus potencialidades.

En los proyectos es donde los estudiantes tienen mayor permanencia, en ellos se desarrollan las habilidades y aptitudes que luego se reproducirá en su vida profesional, ya que la mayoría una vez graduado se mantendrán trabajando en un proyecto y capacitándose de manera continua. Igualmente es donde se puede lograr mayor interacción entre estudiantes y profesores, convirtiéndose este en el patrón más cercano en que los estudiantes se reflejan.

Es importante en cada profesor y en especial los Guías y Jefes de Proyectos, se apropien de técnicas para desarrollar los valores que deseamos y necesitamos que estén formados en nuestros estudiantes en relación con nuestro proyecto social socialista, nuestra condiciones sociales y nuestra UCI.

Cada estudiante debe sentirse comprometido con el rol que desempeña, dentro de su proyecto debe ser capaz de aplicarlo eficientemente y en el tiempo que se le ha propuesto, por eso es muy importante que nuestros estudiantes adquieran una conciencia y un sentido de pertenencia , y que valoren la importancia de la producción de software para nuestro país." [49]

Estos valores sólo pueden ser detectados, valorados y formados a partir del ejemplo, en la interacción de la convivencia de profesores y estudiantes, pues sólo en un ambiente natural es donde se manifiestan todas las aristas de la personalidad como la sinceridad y la correspondencia de los parlamentos con las acciones y compartir e integrar estos esquemas de valores.[50]

2.2.4 Desarrollo de software en los proyectos productivos.

El proceso fundamental que se lleva a cabo en la universidad es el proceso de desarrollo del software:

Un proceso es la definición del conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto, explica los pasos necesarios para terminar el proyecto.

Un proceso define: "Quién", "Qué", "Cuándo" y "Cómo" hay que realizar las cosas para alcanzar un determinado producto de software.

Para llevar a cabo los diferentes procesos dentro del desarrollo de software se utilizan metodologías y herramientas de desarrollo de software. De forma general en los proyectos de la universidad se utiliza una sola metodología, excepto en algunos casos que utilizan más de una metodología.

Una metodología se encarga de elaborar estrategias de desarrollo de software que promueven prácticas adoptivas centradas en las personas o los equipos, orientados hacia la funcionalidad y la entrega, de comunicación intensiva que requiere implicación directa del cliente.

Las metodologías para desarrollo de software más utilizadas en la universidad son: RUP y XP (Programación Extrema).

RUP tiene tres características fundamentales es Iterativo e Incremental, dirigido por casos de usos y centrado en la arquitectura. Además define nueve procesos fundamentales para obtener un software

con alta calidad y a su vez con la seguridad requerida; los seis primeros se consideran flujos ingenieriles de modelación del negocio, levantamiento de requisitos, análisis y diseño, implementación, prueba y despliegue. Los tres restantes se consideran flujos de apoyo: planificación de proyectos, gestión de la configuración y ambiente para realizar las diferentes actividades. En cada uno de los flujos de trabajo, RUP define cuatro fases: fase de inicio, elaboración, construcción y transición, cada fase culmina con un hito; el de la primera fase es tener definidos claramente los objetivos de la organización, en la segunda es definir la arquitectura del sistema, el de la tercera es tener la capacidad operacional inicial del sistema y el de la ultima tener el release del sistema.

La característica fundamental de XP es que su principal proceso esta enfocado a la programación además un aspecto significante es que el cliente forma parte del equipo de desarrollo, también define actividades de planificación, diseño y pruebas.

RUP es una metodología robusta y XP es una metodología ágil, las principales semejanzas entre ambas metodologías es que son un instrumento fundamental para el desarrollo de software, que aseguran la calidad del software implementado, logrando así la seguridad del mismo, además definen roles para realizar las diferentes tareas en cada fase de desarrollo; sin embargo entre ambas metodologías existen varias diferencias. (Ver figura 2.4)

En la universidad cada proyecto desarrolla software con distintas características, de forma general los más comunes que se desarrollan son software de aplicaciones, además de multimedia, entre otros.

Una actividad fundamental en el desarrollo del software es la implementación, para ello en nuestra universidad se utilizan varios lenguajes de programación por ejemplo C #, C ++, Java, PHP, entre otros.

2.3 Producción de software en la Facultad 7.

La producción de software en la facultad siete esta vinculada al sector de la salud, su objetivo es elaborar y mantener software que permiten contribuir al proceso de informatización en dicho sector. Está responsabilidad es de todos los trabajadores de la facultad, a través de la dirección del vicedecano de producción, el cual organiza la producción por áreas temáticas, facilitando que se creen por varios grupos diferentes proyectos. Cuando surge un nuevo proyecto se discute en la IP, entonces se decide si se acepta o no, en caso positivo se lleva el proyecto a la facultad y se le asigna un área

temática. Por lo general cada proyecto tiene como promedio 70 estudiantes y 4 profesores, los que tienen asignado un laboratorio con el equipamiento necesario, constituyendo este su puesto de trabajo.

METODOLOGÍA ÁGIL		METODOLOGÍA ROBUSTA
Se basan en heurística para la producción de software	⇨	Se basan en Normas
Aceptan y fomentan el cambio	⇨	Mayor o menor resistencia a los cambios
El cliente forma parte del equipo	⇨	El cliente no forma parte del equipo, solo se tiene encuentros planificados con el cliente, entrevistas, etc
Equipo de trabajo pequeño o mediano	⇨	Equipos grandes mayores de 15-20 miembros.
Procesos con pocas reglas	⇨	Procesos con muchas normas.
Poca documentación Poco análisis y diseño	⇨	Mucha documentación Mucho análisis y diseño
Conceden poca importancia a la arquitectura de los sistemas	⇨	Conceden mucha importancia a la arquitectura de los sistemas

Figura 2.4 Principales diferencias entre metodología ágil y metodología robusta.

En este momento en la facultad existen 12 proyectos vinculados a la salud y otros proyectos que tienen otra misión como por ejemplo, el portal de GPI (Grupo de Procesamiento de Imágenes y Señales) y el proyecto GICAC (Grupo de Investigación Control y Aseguramiento de la calidad).

En la figura 2.5 aparece la relación de los proyectos productivos de la Facultad 7 por áreas temáticas.

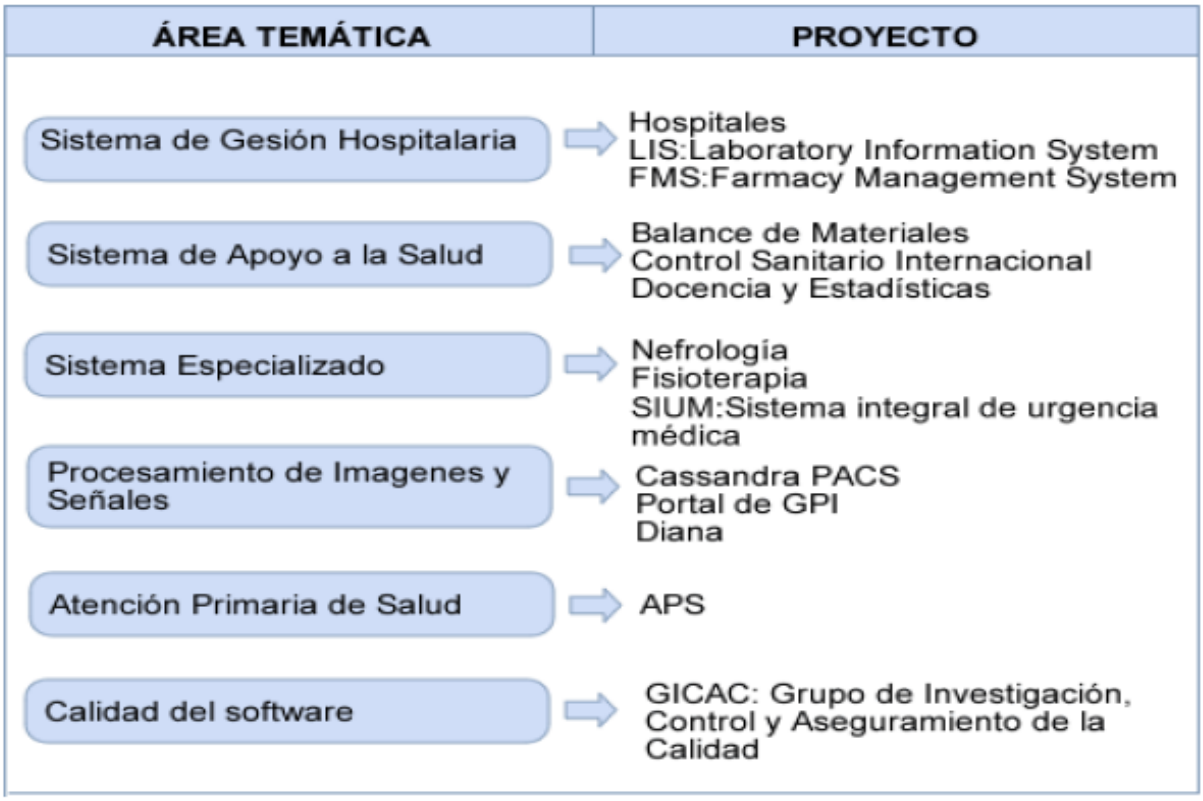


Figura 2.5 Relación de los proyectos productivos de la Facultad 7 por área temática.

2.4 Aplicación de Técnicas para el diagnóstico.

Las principales técnicas empleados que permitieron fundamentar la investigación de este trabajo fueron las siguientes:

La entrevista

Es un método de recopilación de datos empíricos, un proceso de comunicación entre dos o más personas, generalmente de forma oral donde las preguntas al entrevistado se hacen por vía directa, se deben desarrollar preguntas que permiten respuestas precisas. [52]

Existen varios tipos de entrevistas pero las utilizadas en esta investigación son las siguientes**.**

Entrevista no estructurada

"Es más flexible y abierta, aunque los objetivos de la investigación rigen a las preguntas, su contenido, orden, profundidad y formulación se encuentran por entero en manos del entrevistador. Si bien el investigador, sobre la base del problema, los objetivos y las variables, elabora las preguntas antes de realizar la entrevista, modifica el orden, la forma de encauzar las preguntas o su formulación para adaptarlas a las diversas situaciones y características particulares de los sujetos de estudio.

Entre las **ventajas** de este tipo de Entrevista se tienen:

- Es adaptable y susceptible de aplicarse a toda clase de sujetos en situaciones diversas.
- Permite profundizar en temas de interés.
- Orienta posibles hipótesis y variables cuando se exploran áreas nuevas.

Entre sus **desventajas** se mencionan:

- Se requiere de mayor tiempo.
- Es más costoso por la inversión de tiempo de los entrevistadores.
- Se dificulta la tabulación de los datos.
- Se requiere mucha habilidad técnica para obtener la información y mayor conocimiento del tema.

Dentro de la Entrevista no estructurada se comentarán tres tipos de: Entrevista a profundidad, Entrevista enfocada y Entrevista focalizada. A continuación se explica la entrevista enfocada que fue la utilizada en esta investigación.

Entrevista Enfocada:

Se puede decir que la Entrevista enfocada, es una Entrevista en profundidad pero específicamente dirigida a situaciones concretas. Va dirigida a un individuo concreto, caracterizado y señalado previamente por haber tomado parte de la situación o experiencia definida.

A diferencia de la Entrevista a profundidad, la Entrevista enfocada no revive toda la vida, sino la reconstrucción de una experiencia personal concreta. De alguna manera el entrevistador conoce de antemano directa o indirectamente, esta situación con los elementos, procesos y estructura total de la misma y la ha analizado sistemáticamente. En base de este análisis es que se elabora la guía de preguntas.

Funciones de la Entrevista:

Existen cuatro funciones básicas y principales que cumple la Entrevista en la investigación científica:

- Obtener información de individuos y grupos
- Facilitar la recolección de información
- Influir sobre ciertos aspectos de la conducta de una persona o grupo (opiniones, sentimientos, comportamientos, etc.)
- Es una herramienta y una técnica extremadamente flexible, capaz de adaptarse a cualquier condición, situación, personas, permitiendo la posibilidad de aclarar preguntas, orientar la investigación y resolver las dificultades que pueden encontrar la persona entrevistada.

Ventajas:

- La Entrevista es una técnica eficaz para obtener datos relevantes y significativos desde el punto desde el punto de vista de las ciencias sociales, para averiguar
- La información que el entrevistador obtiene a través de la Entrevista es muy superior que cuando se limita a la lectura de respuesta escrita
- Su condición es oral y verbal.
- A través de la Entrevista se pueden captar los gestos, los tonos de voz, los énfasis, etc., que aportan una importante información sobre el tema y las personas entrevistadas.

La ventaja esencial de la Entrevista reside en que son los mismos actores sociales quienes nos proporcionan los datos relativos a sus conductas, opiniones, deseos, actitudes, expectativas, etc. Cosas que por su misma naturaleza es casi imposible observar desde fuera.

La revisión de documentos

Se utiliza para recoger la información que se encuentra registrada en un documento establecido. Se utilizaron los listados de los perfiles de las facultades, registros de proyectos por área temática de la facultad siete, además de documentos de gran importancia en el tema, etc.

La encuesta

Técnica cuantitativa que consiste en una investigación realizada sobre una muestra de sujetos, representativa de un colectivo más amplio que se lleva a cabo en el contexto de la vida cotidiana, utilizando procedimientos estandarizados de interrogación con el fin de conseguir mediciones cuantitativas sobre una gran cantidad de características objetivas y subjetivas de la población.

Ventajas:

-Técnica más utilizada y que permite obtener información de casi cualquier tipo de población.
-Permite obtener información sobre hechos pasados de los encuestados.

-Gran capacidad para estandarizar datos, lo que permite su tratamiento informático y el análisis estadístico.
-Relativamente barata para la información que se obtiene con ello.

Inconvenientes:

-No permite analizar con profundidad temas complejos (recurrir a grupos de discusión).
El Cuestionario es el instrumento de la encuesta y es un instrumento de recogida de datos rigurosamente estandarizado que operacionaliza las variables objeto de observación e investigación, por ello las preguntas de un cuestionario son los indicadores.

Tipos de encuestas:

Las encuestas tienen por objetivo obtener información estadística indefinida, mientras que los censos y registros vitales de población son de mayor alcance y extensión. Este tipo de estadísticas pocas veces otorga, en forma clara y precisa, la verdadera información que se requiere, de ahí que sea necesario realizar encuestas a esa población en estudio, para obtener los datos que se necesitan para un buen análisis. "[53]

De acuerdo con la información que se quiere obtener los tipos de preguntas a obtener se pueden clasificar de la forma siguiente:

Cerradas: Se limita su respuesta a varias posibilidades previstas, donde la respuesta esta estructurada por comparaciones.

Abiertas: Son preguntas para ser respondidas libremente, no permiten obtener con exactitud la información deseada, sólo se logra conocer la opinión del encuestado.

Semicerradas: Limita la respuesta pero deja espacio libre para emitir opiniones sobre el tema.

Directas: Cuando el objetivo de la pregunta coincide con el objeto de interés del investigador.

Indirectas: Cuando de la respuesta se infiere la verdadera información que se quiere obtener. La formulación de este tipo de pregunta es un de las tareas mas difíciles que se enfrenta en la elaboración de un cuestionario.

De contenido: Por el contenido pueden ser objetivas cuando se refieren a hechos concretos o subjetivos cuando se buscan opiniones, actitudes del encuestado, etc.

De filtro: Permiten acceder a preguntas para las cuales se necesita cierta información.

De colchón: Para relajar tensiones que se producen por preguntas complejas o controvertidas.

De control: Se usan para valorar la consistencia de las respuestas dadas a determinadas preguntas.

En la investigación realizada los tipos de encuestas que se utilizaron fueron las cerrada, directa y de filtro.

En la investigación realizada los tipos de encuestas utilizados fueron cerradas, de filtro y del control.

"Otro tipo de Encuestas es ***Encuestas por Muestreo*** en donde se elige una parte de la población que se estima representativa de la población total. Debe tener un diseño muestral, necesariamente debe tener un marco de donde extraerla y ese marco lo constituye el censo de población. La encuesta (muestra o total), es una investigación estadística en que la información se obtiene de una parte representativa de las unidades de información o de todas las unidades seleccionadas que componen el universo a investigar. La información se obtiene tal como se necesita para fines estadístico-demográficos.

Una forma reducida de una encuesta por muestreo es un "*sondeo de opinión*", esta forma de encuesta es similar a un muestreo, pero se caracteriza porque la muestra de la población elegida no es suficiente para que los resultados puedan aportar un informe confiable. Se utiliza solo para recolectar algunos datos sobre lo que piensa un número de individuos de un determinado grupo sobre un determinado tema.

Actualmente, existen sistemas de gestión de encuestas en internet, que están acercando su utilización a investigadores que hasta el momento no tenían acceso a los medios necesarios para ejecutarlas. Es un novedoso método de hacer encuestas, que permite que cualquier persona o empresa realice un estudio de investigación de una forma rápida y a un precio asequible." [54]

Muestreo Aleatorio Estratificado

Existen pocas áreas donde el impacto del desarrollo de la estadística se haya hecho sentir más que en la ingeniería. La estadística ha venido a ser una herramienta vital para ingenieros, les permite comprender fenómenos sujetos a variaciones y predecirlos o controlarlos eficazmente. En este trabajo se utilizó para determinar la cantidad de población que será encuestada, y el tamaño de la muestra es decir la cantidad de encuestas que se realizarán.

El Muestreo Aleatorio Estratificado consiste en subdividir la población en subpoblaciones de tal forma que la unión de ellos será la población, y la intersección de cualquiera de los dos dará como resultado el conjunto vació, es decir, no tendrán elementos comunes.

A las subpoblaciones se les llamará estratos, se tratará de conformar estos de modo que los elementos dentro de ellas sean homogéneos. El tamaño de la muestra se distribuirá entre los estratos,

en función de distintos criterios pero lo que caracteriza al MAE (Muestro aleatorio estratificado) es que la selección de la muestra de cada estrato se hará bajo el procedimiento de muestreo irrestricto aleatorio y se realizará independientemente de los diferentes estratos.
Es recomendable estratificar en función del tamaño de las unidades y distribuir la muestra proporcionalmente al número de unidades de los estratos.

Notación

Se supone que la población consta de n unidades y están distribuidas en L estratos, constituyen una partición de la población; se representará por N_h en el n_i de u en el estado h-ésimo, de aquí:

$$N = N_1 + N_2 + \ldots + N_h + \ldots + N_l$$

$$x_h = \sum_{1}^{N_h} x_{hi} \quad \text{media} \quad \bar{x}_h = \frac{1}{N_h} \sum_{i}^{N_h} x_{hi}$$

Total de población:

$$f_h = \frac{n_h}{N_h}$$

Fracción de muestreo del estrato:
Si el tamaño de la muestra de los estratos se distribuye proporcionalmente al numero de unidades en el estrato, es decir se cumple que:

$$n_h = n\frac{N_h}{N}$$

Para todo h .En este caso se dice que la distribución de la muestra se ha hecho con asignación proporcional.

Para la determinación del tamaño de muestra de una población con S^2 desconocida se puede determinar mediante la expresión:

$$n = \frac{\left(\frac{Z_{1-\frac{\alpha}{2}}}{d}\right)^2 P(1-P)}{1+\frac{1}{N}\left(\frac{Z_{1-\frac{\alpha}{2}}}{d}\right)^2 p(1-P)-\frac{1}{N}}$$

Donde 1-α es el nivel de confianza, d es el error absoluto, Z_1 percentil de la distribución normal, P Proporción de la población y N tamaño de la muestra.

Realizando el cálculo del tamaño de muestra para un nivel de confianza

$1-\alpha=90\%$, con un error absoluto $d=0.10$, se obtiene un percentil de la distribución normal

$Z_1=1.64$ y se asume como proporción de la población P=0.5, $N=323$ que es la cantidad total de los integrantes de los proyectos de la facultad; se obtiene que:

$$n = \frac{\left(\frac{1.64}{0.10}\right)^2 0.5(1-0.5)}{1+\frac{1}{323}\left(\frac{1.64}{0.10}\right)^2 0.5(1-0.5)-\frac{1}{323}} = \frac{67.24}{1.2052} = 55{,}79 \approx 56$$

Se aplicar un total de 56 encuestas:

Conociendo que n=56, N_h es la cantidad de integrantes de cada proyecto, N=323 que es el total de integrantes de los diferentes proyectos.

Calculando el tamaño de la muestra por estrato (se considera estrato a cada proyecto)

$$n_h = n\frac{N_h}{N}$$

aplicando ; resultan los tamaños de muestra por estrato que se muestran por estrato que se muestran en la tabla 2.6

Estratos	Cantidad	Tamaño de la Muestra
GPI	81	14
APS	80	14
Hospitales	70	12
Docencia	35	6
Nefrología	17	3
Balance de Materiales	14	2
SIUM	13	2
Fisioterapia	13	2

Tabla 2.1 Tamaños de muestra por estratos de la población.

2.5 Diagnóstico de la producción de software en la Facultad 7.

Resultados de la encuesta aplicada a los integrantes de los proyectos productivos de la Facultad 7.

Para detectar algunas deficiencias existentes en los proyectos productivos relacionadas con la seguridad, se aplicaron diferentes técnicas para el diagnóstico, entre ella esta la encuesta realizada a los integrantes de los diferentes proyectos productivos de la facultad. (Ver Anexo 1)

La muestra hallada por regiones representa el 17,28; 17,5; 17,14; 17,14; 17,64; 14,28; 15,38; 15,38 % de los integrantes (estudiantes y profesores) de los diferentes proyectos de la facultad. A estos grupos se les aplico la encuesta relacionada con el aseguramiento de la calidad que fomenta a la vez como en los proyectos productivos de la Facultad 7 no se tiene una idea de la seguridad del software desde el inicio del producto. En la que se recoge diferentes opiniones acerca de cómo se manifiesta la seguridad del software que se desarrolla; así como los principales procesos que se llevan a cabo en el desarrollo del software, con vista a detectar los principales problemas que están afectando la seguridad de los proyectos productivos de la Facultad 7.

Se decidió segmentarlos por regiones para evitar que los diferentes procesos de desarrollo del software que presentan cada proyecto puedan influir significativamente en los resultados, estratificándose entonces por las distintas regiones.

Los resultados obtenidos relevan un cierto desconocimiento en sentido general de los principales estándares y modelos para la seguridad de los productos, sin embargo todos los integrantes de los proyectos tiene bien definidos los conceptos de seguridad informática y aseguramiento de la seguridad. Los proyectos que sólo se dedican a la producción de software representan un 85,8%; mientras los que aparte de producir se dedican a otras actividades solo el 14,2%.

El 3,57% de los encuestados creen en la seguridad del producto que desarrollan es excelente, el 1,78% buena, el 82,14% regular y el 12,5 % no sabe que es la seguridad del software que desarrollan. El 78,63% de los encuestados platean que el cliente se siente satisfecho con el servicio que brindan mientras que el 16,07% creen que no y solo el 5,30% no sabe si el cliente se siente satisfecho. El 94,64% de los proyectos utilizan para el desarrollo de software la metodología RUP y el 3,75% XP y existe en la facultad un solo proyecto que aparte de estas dos también utilizan otras lo que representa el 1,78%.

El 19,64% de los encuestados plantean que el proyecto que pertenecen se realizan técnicas de control de la calidad y que conocen métodos de prueba y el 80,35% platean que no. A continuación aparecen tabulados los resultados de la encuesta con mayor precisión a través de la siguientes tablas.

Procedimiento de Seguridad	**Total**	**Porciento%**
Si	48	85,8
No	8	14,2

Tabla 2.2 Resultados de los proyectos que solo se dedican a la producción.

Procedimiento de Seguridad	**Total**	**Porciento%**
Si	3 no definidos	5,35
No	53	94,64

Tabla 2.3 Resultados de los proyectos que utilizan procedimientos de seguridad.

Seguridad del producto	**Total**	**Porciento%**
Excelente	**2**	**3.57**
Bien	**1**	**1.78**
Regular	**46**	**82,14**
No sabe	**7**	**12,5**

Tabla 2.4 Resultados sobre la seguridad del producto que se realiza.

Satisfacción al cliente	Total	Porciento%
Si	44	78,63
No	9	16,07
No sabe	3	5,30

Tabla 2.5 Resultados sobre la satisfacción del cliente con el producto que se desarrolla.

Metodología de Desarrollo	Total	Porciento%
RUP	53	94,64
XP	2	3,57
Otras	1	1,78

Tabla 2.6 Resultados de la metodología de desarrollo que se utilizan en los diferentes proyectos.

Métodos de Prueba de Seguridad	Total	Porciento%
Si	1	1,78
No	55	98,22

Tabla 2.7 Resultados sobre los métodos de pruebas de seguridad en el software.

Téc. Control de la Calidad	Total	Porciento%
Si	11	19,64
No	45	80,35

Tabla 2.8 Resultados sobre las técnicas de control de la calidad que se realizan en los proyectos.

2.6 Conclusiones.

La investigación anteriormente expuesta arrojo como conclusión que:

1. El proceso productivo en la universidad, no está dedicado solamente a programar software también debe asistir al conjunto de actividades que lo garantizan. Estas van, desde el proceso organizacional de división del trabajo por roles, el desarrollo de bases tecnológicas que implican enfrentarse a problemas desconocidos, hasta el hecho de lograr un trabajo integrado en equipo.

2. El objetivo fundamental de la producción en la UCI está dedicado a desarrollar proyectos para la informatización del país. Así como la informatización de la vida de la propia universidad, y la exportación. Convirtiéndose la universidad en una de las instituciones más productivas y apartadoras de recursos para el país.

3. Los proyectos productivos de la Facultad 7 no tienen definido un procedimiento que garantice la seguridad en el entorno de desarrollo y productos obtenidos.

Capítulo 3: Propuesta de procedimiento.

3.1 Introducción.

Llevar la seguridad interna en un proyecto productivo, es una tarea que requiere de varios factores y solo es conformada bajo el cumplimiento de diversos pasos fundamentales. Para el desarrollo de este capitulo se propone un procedimiento para garantizar la seguridad en el entorno de desarrollo y productos obtenidos en los proyectos productivos de la Facultad 7. Este pretende proporcionar la secuencia de pasos mínimos necesarios, para lograr el cumplimiento del propósito fundamental: la seguridad en la producción. El mismo se rige por el siguiente hilo conductor.

Figura 3.1 Hilo conductor del procedimiento de seguridad.

El procedimiento se basa en su curso inicial en lograr una seguridad de la información con la que se cuenta en los proyectos, para así tener una seguridad mínima a la hora de pasar a la fase de construcción que es donde se va a fomentar la seguridad en el software.

El mismo se divide por etapas y estas a su vez están constituidas por diferentes modelos, estándares y paradigmas .En la primera etapa del procedimiento que es la planificación de la seguridad se establece el Sistema de Gestión de la Seguridad de la Información (SGSI), el cual se encuentra propuesto en el estándar internacional ISO/IEC 27001.El SGSI se distingue por garantizar la seguridad de la información a través de que define una política de seguridad para la organización ,

identifica, analiza y evalúa los riesgos, actualiza los planes de seguridad , etc. La segunda etapa que se relaciona con la construcción de la seguridad, se guía por la metodología RUP por ser esta la más utilizada en los proyectos de software de la facultad, según la encuesta aplicada.

La metodología RUP se guiará por el modelo SDL para la obtención de la seguridad en el software de Microsoft, el mismo se empleara de manera diferente en este procedimiento ya que Microsoft no utiliza RUP .Se propone realizar una serie de actividades del modelo SDL que se deben cumplir en cada uno de los flujos de trabajo que propone RUP (Modelamiento del Negocio, Requerimientos, Análisis y Diseño, Implementación y Prueba) y durante todas la fases de RUP(Inicio, Elaboración , Construcción, y Transición).

En el flujo de trabajo de implementación además de lo que propone el modelo SDL, también se dará como variante la aplicación del paradigma POA (Programación Orientada a Aspectos), el cual nos permite encapsular y abstraer el concepto de seguridad, logrando con esto: bajo costo de mantenimiento, mayor flexibilidad, y mayor especialización .Durante el flujo de trabajo de pruebas se expondrán las pruebas de seguridad que se le deben realizar al software para procurarlo concluido. Ya en la tercera etapa se hará una supervisión de la seguridad de manera general, ya que en cada una de las etapas se debe hacer una supervisión de la seguridad; esta supervisión general se hará a través de una auditoria de seguridad. En la etapa final del procedimiento se hará una Revisión Final de Seguridad (RFS)

A continuación se describen los epígrafes del procedimiento propuesto.

3.2 Objetivo.

Asegurar la seguridad en el proceso de desarrollo del software en los productos obtenidos de los proyectos productivos de la facultad 7.

3.3 Alcance.

Para los proyectos productivos de la Facultad 7 de la UCI.

3.4 Propósito.

Definir los pasos a seguir para lograr que el software que se construye obtenga la seguridad requerida.

3.5 Acrónico.

ISO/IEC: ISO (la Organización Internacional de Estandarización) e IEC (la Comisión Electrotécnica Internacional) forman el sistema especializado para la estandarización mundial.

SDL: Ciclo de vida de desarrollo de seguridad (SDL, Security Development Lifecycle) de Trustworthy Computing (computación confiable), un proceso que Microsoft utiliza ahora para desarrollar software que pueda resistir ataques malintencionados. Este proceso incorpora varias actividades y materiales relacionados con la seguridad a cada una de las fases del proceso de desarrollo de software de Microsoft.

SGSI: Sistema de Gestión de Seguridad de la Información es el concepto central sobre el que se construye ISO 27001.

RUP: Proceso Unificado de Rational (Rational Unifield Process- RUP). RUP es un proceso de desarrollo de software que constituye una forma disciplinada de asignar tareas y responsabilidades en una organización de desarrollo. Tiene como objetivo la producción de software de calidad dentro de plazos y costos predecibles.

RFS: Revisión Final de Seguridad, se implementa en el SDL en la etapa final.

POA: Programación Orientada a Aspectos. Nuevo paradigma de programación.

3.6 Descripción.

3.6.1 1ra Etapa: Planificación de la Seguridad.

La mayoría de los productores de software y empresas consumidoras de estos justifican la seguridad del software bajo la seguridad de la información, confundiéndose dos aspectos diferentes. Un típico ejemplo de lo anterior es el uso de “firewall” y técnicas criptográficas, que si bien es cierto, pueden ser útiles a la hora de reforzar la seguridad de un producto software, no son suficientes por si mismos si la aplicación no se ha desarrollado teniendo en cuenta una serie de aspectos relacionados con la seguridad desde el inicio. La seguridad de la información y la del software están muy relacionadas y si no logras una no puedes lograr la otra. Por eso en la primera etapa del procedimiento se planificara la seguridad pero desde la información que se tiene previamente en el proyecto, logrando con esto resguardar lo que ya se tiene y además de comenzar a trabajar el termino de la seguridad mucho más

fuerte en los equipos de desarrollo. Con los siguientes pasos que llevaremos a cabo en la etapa de planificación lograremos lo expuesto.

1) Confección de los equipos de desarrollo. Inclusión del equipo seguridad.
2) Proceso de capacitación del equipo de seguridad.
3) Proceso de capacitación del equipo de desarrollo entorno a la seguridad..
4) Aplicación del SGSI.

- **Confección de los equipos de desarrollo. Inclusión del equipo seguridad.**

Cuando se va a realizar un producto de software por una organización que se dedica a esto, lo primero que se realiza, después de tener todo el equipamiento, es la confección de los equipos. En la UCI esto se realiza por proyecto y en cada proyecto se confecciona por módulo, que el módulo es el producto como tal.

Este módulo esta conformado por un equipo de desarrollo, que en estos se encuentra integrado por sus diseñadores, implementadores, arquitectos y personal necesario para la elaboración del mismo, con lo que no se cuenta en estos momentos es con personal capacitado en seguridad y como primer paso para la realización del procedimiento que se esta proponiendo es la confección del equipo de segurida, equipo que pueda guiar a todo se equipo de desarrollo en la confección de un software seguro.

Esto conlleva primeramente que a nivel central se cuente con un grupo central de seguridad que controle el desarrollo y la evolución de las prácticas recomendadas de seguridad y las mejoras de los procesos, actúa como fuente de conocimientos para toda la organización. En cada modulo se debe contar con un equipo en seguridad, pero además se debe contar con uno a nivel de cada proyecto así como de área temática.

El equipo de seguridad debe estar disponible para recurrir a él con frecuencia durante el diseño y el desarrollo del software, y es preciso confiarle información técnica y empresarial confidencial. Entre las responsabilidades del equipo de seguridad se incluyen:

- Desarrollo, mantenimiento y mejora del SDL, incluida la definición de los aspectos obligatorios del proceso.
- Desarrollo, mejora y realización de los cursos a los ingenieros.
- Asignación de los "asesores de seguridad" que guían a los equipos de producto a través del proceso, realizan revisiones de los equipos de producto y se aseguran de que se responde a las preguntas planteadas por los equipos de producto de manera puntual, precisa y autorizada.
- Servicio de expertos en la materia en una amplia gama de temas de seguridad, asegurándose de que las preguntas realizadas a los asesores de seguridad o planteadas mediante ellos reciben respuestas puntuales y precisas.
- Realización de las revisiones finales de seguridad antes del lanzamiento del software.
- Investigación técnica de las vulnerabilidades detectadas en el software entregado a los clientes, para asegurarse de que se entienden las causas que las han originado y se tomen las medidas adecuadas.

- **Proceso de capacitación del equipo de seguridad.**

En este paso del procedimiento lo que se desea es lo que actualmente se esta haciendo en cada uno de los proyectos de la facultad , la realización de cursos para la capacitación de los integrantes de los equipos de desarrollo, según la función que van a realizar .Pero como un nuevo aspecto a tener en cuenta es la capacitación del equipo seguridad, en cada uno de los niveles donde este integrado , y que este capacitación sea brindada fundamentalmente por el equipo de seguridad central (aunque tal vez sea preciso contratar a consultores que participen en la creación y entrenamiento de los miembros del equipo) , para una mayor comunicación y reciprocidad entre las personas encargadas del tema de la seguridad en el software en la universidad.

- **Proceso de capacitación del equipo de desarrollo entorno a la seguridad.**

Ya en este aspecto lo que se quiere lograr es la capacitación de todos los miembros del equipo con respecto a la seguridad, que todos de alguna manera conozcan y dominen la información de las normas que existen para la realización de la seguridad, además de las principales vulnerabilidades a las que puede estar sometido el software, como también dominar los aspectos del nuevo paradigma de la programación relacionado al tema. Con todo esto se logra una integración del equipo de seguridad, con el equipo de desarrollo del que es miembro, al trasmitirle parte de los conocimientos que el

domina, logrando que su equipo este preparado en al menos algunos temas de seguridad para la realización de un software seguro.

- **Aplicación del SGSI.**

Después de la confección de los equipos de desarrollo y la capacitación de los mismos lo que se desea es la gestión de la seguridad de la información, ya que para lograr la seguridad en el software primeramente debemos gestionar la seguridad de la información que ya dominamos y de los medios que la poseen. Para esto se aplicara el SGSI el cual se encuentra propuesto en el estándar internacional ISO/IEC 27001.

El SGSI (Sistema de Gestión de Seguridad de la Información) es el concepto central sobre el que se construye ISO 27001.

La gestión de la seguridad de la información debe realizarse mediante un proceso sistemático, documentado y conocido por toda la organización.

Garantizar un nivel de protección total es virtualmente imposible, incluso en el caso de disponer de un presupuesto ilimitado. El propósito de un sistema de gestión de la seguridad de la información es, por tanto, garantizar que los riesgos de la seguridad de la información sean conocidos, asumidos, gestionados y minimizados por la organización de una forma documentada, sistemática, estructurada, repetible, eficiente y adaptada a los cambios que se produzcan en los riesgos, el entorno y las tecnologías.

¿Qué es un SGSI?

SGSI es la abreviatura utilizada para referirse a un Sistema de Gestión de la Seguridad de la Información. ISMS es el concepto equivalente en idioma inglés, siglas de *Information Security Management System*.

En el contexto aquí tratado, se entiende por información todo aquel conjunto de datos organizados en poder de una entidad que posean valor para la misma, independientemente de la forma en que se guarde o transmita (escrita, en imágenes, oral, impresa en papel, almacenada electrónicamente,

proyectada, enviada por correo, fax o e-mail, transmitida en conversaciones, etc.), de su origen (de la propia organización o de fuentes externas) o de la fecha de elaboración.[26]

La seguridad de la información, según ISO 27001, consiste en la preservación de su confidencialidad, integridad y disponibilidad, así como de los sistemas implicados en su tratamiento, dentro de una organización. Así pues, estos tres términos constituyen la base sobre la que se cimienta todo el edificio de la seguridad de la información:

• Confidencialidad: la información no se pone a disposición ni se revela a individuos, entidades o procesos no autorizados.

• Integridad: mantenimiento de la exactitud y completitud de la información y sus métodos de proceso.

• Disponibilidad: acceso y utilización de la información y los sistemas de tratamiento de la misma por parte de los individuos, entidades o procesos autorizados cuando lo requieran.

Para garantizar que la seguridad de la información es gestionada correctamente, se debe hacer uso de un proceso sistemático, documentado y conocido por toda la organización, desde un enfoque de riesgo empresarial. Este proceso es el que constituye un SGSI.

¿Para qué sirve un SGSI?

La información, junto a los procesos y sistemas que hacen uso de ella, son activos muy importantes de una organización. La confidencialidad, integridad y disponibilidad de información sensible pueden llegar a ser esenciales para mantener los niveles de competitividad, rentabilidad, conformidad legal e imagen empresarial necesarios para lograr los objetivos de la organización y asegurar beneficios económicos. [55]

"Las organizaciones y sus sistemas de información están expuestos a un número cada vez más elevado de amenazas que, aprovechando cualquiera de las vulnerabilidades existentes, pueden someter a activos críticos de información a diversas formas de fraude, espionaje, sabotaje o vandalismo. Los virus informáticos, el "hacking" o los ataques de denegación de servicio son algunos ejemplos comunes y conocidos, pero también se deben considerar los riesgos de sufrir incidentes de seguridad causados voluntaria o involuntariamente desde dentro de la propia organización o aquellos provocados accidentalmente por catástrofes naturales y fallos técnicos.

El cumplimiento de la legalidad, la adaptación dinámica y puntual a las condiciones variables del entorno, la protección adecuada de los objetivos de negocio para asegurar el máximo beneficio o el aprovechamiento de nuevas oportunidades de negocio, son algunos de los aspectos fundamentales en los que un SGSI es una herramienta de gran utilidad y de importante ayuda para la gestión de las organizaciones.

El nivel de seguridad alcanzado por medios técnicos es limitado e insuficiente por sí mismo. En la gestión efectiva de la seguridad debe tomar parte activa toda la organización, con la gerencia al frente, tomando en consideración también a clientes y proveedores de bienes y servicios. El modelo de gestión de la seguridad debe contemplar unos procedimientos adecuados y la planificación e implantación de controles de seguridad basados en una evaluación de riesgos y en una medición de la eficacia de los mismos.

El Sistema de Gestión de la Seguridad de la Información (SGSI) ayuda a establecer estas políticas y procedimientos en relación a los objetivos de negocio de la organización, con objeto de mantener un nivel de exposición siempre menor al nivel de riesgo que la propia organización ha decidido asumir.

Con un SGSI, la organización conoce los riesgos a los que está sometida su información y los asume, minimiza, transfiere o controla mediante una sistemática definida, documentada y conocida por todos, que se revisa y mejora constantemente.

¿Qué incluye un SGSI?

En el ámbito de la gestión de la calidad según ISO 9001, siempre se ha mostrado gráficamente la documentación del sistema como una pirámide de cuatro niveles. Es posible trasladar ese modelo a un Sistema de Gestión de la Seguridad de la Información basado en ISO 27001 de la siguiente forma :

Documentos de Nivel 1

Manual de seguridad: por analogía con el manual de calidad, aunque el término se usa también en otros ámbitos. Sería el documento que inspira y dirige todo el sistema, el que expone y determina las intenciones, alcance, objetivos, responsabilidades, políticas y directrices principales, etc., del SGSI.

Documentos de Nivel 2

Procedimientos: documentos en el nivel operativo, que aseguran que se realicen de forma eficaz la planificación, operación y control de los procesos de seguridad de la información.

Documentos de Nivel 3

Instrucciones, lista de chequeo (*checklists)* y formularios: documentos que describen cómo se realizan las tareas y las actividades específicas relacionadas con la seguridad de la información.

Documentos de Nivel 4

Registros: documentos que proporcionan una evidencia objetiva del cumplimiento de los requisitos del SGSI; están asociados a documentos de los otros tres niveles como *output* que demuestra que se ha cumplido lo indicado en los mismos.

De manera específica, ISO 27001 indica que un SGSI debe estar formado por los siguientes documentos (en cualquier formato o tipo de medio):

- Alcance del SGSI: ámbito de la organización que queda sometido al SGSI, incluyendo una identificación clara de las dependencias, relaciones y límites que existen entre el alcance y aquellas partes que no hayan sido consideradas (en aquellos casos en los que el ámbito de influencia del SGSI considere un subconjunto de la organización como delegaciones, divisiones, áreas, procesos, sistemas o tareas concretas).

- Política y objetivos de seguridad: documento de contenido genérico que establece el compromiso de la dirección y el enfoque de la organización en la gestión de la seguridad de la información.

- Procedimientos y mecanismos de control que soportan al SGSI: aquellos procedimientos que regulan el propio funcionamiento del SGSI.

- Enfoque de evaluación de riesgos: descripción de la metodología a emplear (cómo se realizará la evaluación de las amenazas, vulnerabilidades, probabilidades de ocurrencia e impactos en relación a los activos de información contenidos dentro del alcance seleccionado), desarrollo de criterios de aceptación de riesgo y fijación de niveles de riesgo aceptables .

- Informe de evaluación de riesgos: estudio resultante de aplicar la metodología de evaluación anteriormente mencionada a los activos de información de la organización.

- Plan de tratamiento de riesgos: documento que identifica las acciones de la dirección, los recursos, las responsabilidades y las prioridades para gestionar los riesgos de seguridad de la información, en función de las conclusiones obtenidas de la evaluación de riesgos, de los objetivos de control identificados, de los recursos disponibles, etc.

- Procedimientos documentados: todos los necesarios para asegurar la planificación, operación y control de los procesos de seguridad de la información, así como para la medida de la eficacia de los controles implantados.

- Registros: documentos que proporcionan evidencias de la conformidad con los requisitos y del funcionamiento eficaz del SGSI.

- Declaración de aplicabilidad: (SOA -*Statement of Applicability*-, en sus siglas inglesas); documento que contiene los objetivos de control y los controles contemplados por el SGSI, basado en los

resultados de los procesos de evaluación y tratamiento de riesgos, justificando inclusiones y exclusiones.

Control de la documentación

Para los documentos generados se debe establecer, documentar, implantar y mantener un procedimiento que defina las acciones de gestión necesarias para:

- Aprobar documentos apropiados antes de su emisión.
- Revisar y actualizar documentos cuando sea necesario y renovar su validez.
- Garantizar que los cambios y el estado actual de revisión de los documentos están identificados.
- Garantizar que las versiones relevantes de documentos vigentes están disponibles en los lugares de empleo.
- Garantizar que los documentos se mantienen legibles y fácilmente identificables.
- Garantizar que los documentos permanecen disponibles para aquellas personas que los necesiten y que son transmitidos, almacenados y finalmente destruidos acorde con los procedimientos aplicables según su clasificación.
- Garantizar que los documentos procedentes del exterior están identificados.
- Garantizar que la distribución de documentos está controlada.
- Prevenir la utilización de documentos obsoletos.
- Aplicar la identificación apropiada a documentos que son retenidos con algún propósito.

¿Cómo se implementa un SGSI?

Para establecer y gestionar un Sistema de Gestión de la Seguridad de la Información en base a ISO 27001, se utiliza el ciclo continuo PDCA, tradicional en los sistemas de gestión de la calidad. “[56]

Figura 3.1 El ciclo continuo PDCA

- ***Plan (planificar):*** establecer el SGSI.
- ***Do (hacer):*** implementar y utilizar el SGSI.
- ***Check (verificar):*** monitorizar y revisar el SGSI.
- ***Act (actuar):*** mantener y mejorar el SGSI.

***Plan:* Establecer el SGSI**

- Definir el alcance del SGSI en términos del negocio, la organización, su localización, activos y tecnologías, incluyendo detalles y justificación de cualquier exclusión.
- Definir una política de seguridad que:
  - Incluya el marco general y los objetivos de seguridad de la información de la organización;
  - Considere requerimientos legales o contractuales relativos a la seguridad de la información;
  - Esté alineada con el contexto estratégico de gestión de riesgos de la organización en el que se establecerá y mantendrá el SGSI;
  - Establezca los criterios con los que se va a evaluar el riesgo;

– Esté aprobada por la dirección.

- Definir una metodología de evaluación del riesgo apropiada para el SGSI y los requerimientos del negocio, además de establecer los criterios de aceptación del riesgo y especificar los niveles de riesgo aceptable. Lo primordial de esta metodología es que los resultados obtenidos sean comparables y repetibles (existen numerosas metodologías estandarizadas para la evaluación de riesgos, aunque es perfectamente aceptable definir una propia).

- Identificar los riesgos:
  – Identificar los activos que están dentro del alcance del SGSI y a sus responsables directos, denominados propietarios;
  – Identificar las amenazas en relación a los activos;
  – Identificar las vulnerabilidades que puedan ser aprovechadas por dichas amenazas;
  – Identificar los impactos en la confidencialidad, integridad y disponibilidad de los activos.

- Analizar y evaluar los riesgos:
  – Evaluar el impacto en el negocio de un fallo de seguridad que suponga la pérdida de confidencialidad, integridad o disponibilidad de un activo de información.
  – Evaluar de forma realista la probabilidad de ocurrencia de un fallo de seguridad en relación a las amenazas, vulnerabilidades, impactos en los activos y los controles que ya estén implementados.
  – Estimar los niveles de riesgo.
  – Determinar, según los criterios de aceptación de riesgo previamente establecidos, si el riesgo es aceptable o necesita ser tratado.

- Identificar y evaluar las distintas opciones de tratamiento de los riesgos para:
  – Aplicar controles adecuados;

  – Aceptar el riesgo, siempre y cuando se siga cumpliendo con las políticas y criterios establecidos para la aceptación de los riesgos.

  – Evitar el riesgo, p. ej., mediante el cese de las actividades que lo originan.

  – Transferir el riesgo a terceros, p. ej., compañías aseguradoras o proveedores de *outsourcing*.

- Seleccionar los objetivos de control y los controles del Anexo A de ISO 27001 para el tratamiento del riesgo que cumplan con los requerimientos identificados en el proceso de evaluación del riesgo.

- Aprobar por parte de la dirección tanto los riesgos residuales como la implantación y uso del SGSI.

- Definir una declaración de aplicabilidad que incluya:
  – Los objetivos de control y controles seleccionados y los motivos para su elección.
  – Los objetivos de control y controles que actualmente ya están implantados.

En relación a los controles de seguridad, el estándar ISO 27002 (antigua ISO 17799) proporciona una completa guía de implantación que contiene 133 controles, según 39 objetivos de control agrupados en 11 dominios. Esta norma es referenciada en ISO 27001, en su segunda cláusula, en términos de "documento indispensable para la aplicación de este documento" y deja abierta la posibilidad de incluir controles adicionales en el caso de que la guía no contemplase todas las necesidades particulares.

***Do:* Implementar y utilizar el SGSI**

- Definir un plan de tratamiento de riesgos que identifique las acciones, recursos, responsabilidades y prioridades en la gestión de los riesgos de seguridad de la información.

- Implantar el plan de tratamiento de riesgos, con el fin de alcanzar los objetivos de control identificados, incluyendo la asignación de recursos, responsabilidades y prioridades.

- Implementar los controles anteriormente seleccionados que lleven a los objetivos de control.

- Definir un sistema de métricas que permita obtener resultados reproducibles y comparables para medir la eficacia de los controles o grupos de controles.

- Procurar programas de formación y concienciación en relación a la seguridad de la información a todo el personal.
- Gestionar las operaciones del SGSI.
- Gestionar los recursos necesarios asignados al SGSI para el mantenimiento de la seguridad de la información.
- Implantar procedimientos y controles que permitan una rápida detección y respuesta a los incidentes de seguridad.

Figura 3.2 Gestion de riegos.

Check: **Monitorizar y revisar el SGSI**

La organización deberá:

• Ejecutar procedimientos de monitorización y revisión para:

  – Detectar a tiempo los errores en los resultados generados por el procesamiento de la información.

  – Identificar brechas e incidentes de seguridad;

  – Ayudar a la dirección a determinar si las actividades desarrolladas por las personas y dispositivos tecnológicos para garantizar la seguridad de la información se desarrollan en relación a lo previsto.

  – Detectar y prevenir eventos e incidentes de seguridad mediante el uso de indicadores;

  – Determinar si las acciones realizadas para resolver brechas de seguridad fueron efectivas.

• Revisar regularmente la efectividad del SGSI, atendiendo al cumplimiento de la política y objetivos del SGSI, los resultados de auditorías de seguridad, incidentes, resultados de las mediciones de eficacia, sugerencias y observaciones de todas las partes implicadas.

• Medir la efectividad de los controles para verificar que se cumple con los requisitos de seguridad.

• Revisar regularmente en intervalos planificados las evaluaciones de riesgo, los riesgos residuales y sus niveles aceptables, teniendo en cuenta los posibles cambios que hayan podido producirse en la organización, la tecnología, los objetivos y procesos de negocio, las amenazas identificadas, la efectividad de los controles implementados y el entorno exterior -requerimientos legales, obligaciones contractuales, etc.-.

• Realizar periódicamente auditorías internas del SGSI en intervalos planificados.

• Revisar el SGSI por parte de la dirección periódicamente para garantizar que el alcance definido sigue siendo el adecuado y que las mejoras en el proceso del SGSI son evidentes.

- Actualizar los planes de seguridad en función de las conclusiones y nuevos hallazgos encontrados durante las actividades de monitorización y revisión.

- Registrar acciones y eventos que puedan haber impactado sobre la efectividad o el rendimiento del SGSI.

***Act:* Mantener y mejorar el SGSI**

La organización deberá regularmente:

- Implantar en el SGSI las mejoras identificadas.

- Realizar las acciones preventivas y correctivas adecuadas en relación a la cláusula 8 de ISO 27001 y a las lecciones aprendidas de las experiencias propias y de otras organizaciones.

- Comunicar las acciones y mejoras a todas las partes interesadas con el nivel de detalle adecuado y acordar, si es pertinente, la forma de proceder.

- Asegurarse que las mejoras introducidas alcanzan los objetivos previstos.

PDCA es un ciclo de vida continuo, lo cual quiere decir que la fase de *Act* lleva de nuevo a la fase de *Plan* para iniciar un nuevo ciclo de las cuatro fases. Téngase en cuenta que no tiene que haber una secuencia estricta de las fases, sino que, p. ej., puede haber actividades de implantación que ya se lleven a cabo cuando otras de planificación aún no han finalizado; o que se monitoricen controles que aún no están implantados en su totalidad.

3.6.2 2da Etapa: Construcción de la Seguridad.

Ya logrado el proceso de planificación de la seguridad eficazmente, se pasa a la etapa de la construcción de la seguridad, con esto lo que se quiere es que durante el proceso de desarrollo del software se apliquen medidas de seguridad en cada una de las etapas de desarrollo, para la obtención de un software seguro.

Para lograr un software seguro, hay que tener en cuenta tres aspectos: proceso reproducible, conocimientos del ingeniero e indicadores y responsabilidad. Este procedimiento se centra en la reproducibilidad del proceso que propone el SDL. El SDL implica cambiar los procesos de una

organización de desarrollo de software mediante la integración de medidas que mejoren la seguridad del software. Mediante este procedimiento se resumen estas medidas y se describe la manera de integrarlas en un ciclo de vida de desarrollo de software habitual. Estas modificaciones no pretenden examinar el proceso de manera exhaustiva, sino agregar puntos de control y materiales de seguridad bien definidos. [57]

El grupo de seguridad en la facultad que controla el desarrollo y la evolución de las prácticas recomendadas de seguridad y las mejoras de los procesos, actúa como fuente de conocimientos para todas las áreas temáticas y la existencia de tal grupo es vital para implementar adecuadamente el SDL, así como para mejorar la seguridad del software.

Se describe a continuación la integración de un conjunto de pasos destinados a aumentar la seguridad del software durante el proceso de desarrollo de software. El objetivo de estas mejoras de procesos es reducir el número y la gravedad de las vulnerabilidades de seguridad del software utilizado por los clientes.

3.6.2.1 El proceso de línea de base.

El proceso de desarrollo de software aceptado de manera general en la Facultad 7 sigue aproximadamente los flujos de trabajo y las fases de RUP que se muestra en la Figura 1

RUP es un proceso de desarrollo de software que constituye una forma disciplinada de asignar tareas y responsabilidades en una organización de desarrollo. Tiene como objetivo la producción de software de calidad dentro de plazos y costos predecibles.

También se considera un producto desarrollado y mantenido por Rational y que es actualizado constantemente para tener en cuenta las mejores prácticas de acuerdo a la experiencia obtenida.

RUP es un proceso y en su modelación define como sus principales elementos:

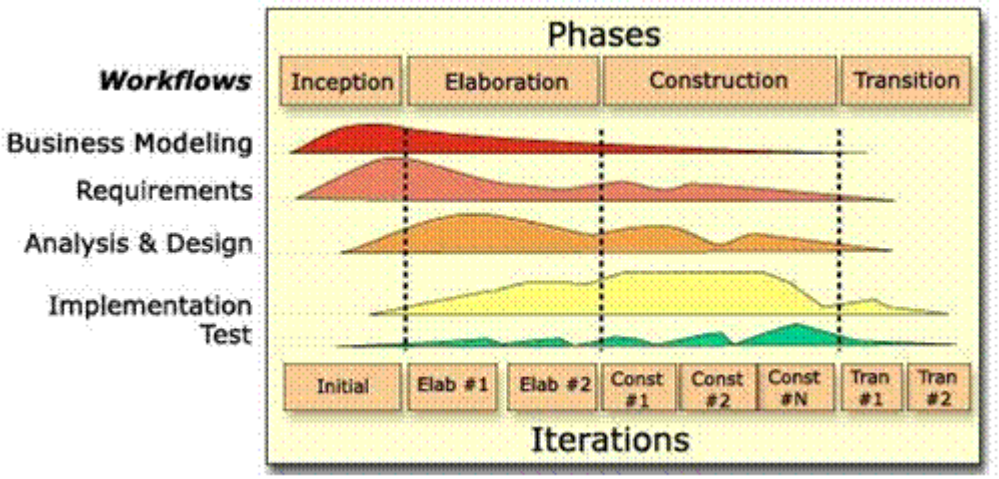


Figura 3.3 (RUP en dos dimensiones) representa el proceso en el que se grafican los flujos de trabajo y las fases y muestra la dinámica expresada en iteraciones y puntos de control.

Trabajadores ("quién"): Define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de los elementos.

Actividades ("cómo"): Es una tarea que tiene un propósito claro, es realizada por un trabajador y manipula los elementos.

Artefactos ("qué"): Productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.

Flujo de actividades ("Cuándo"): Secuencia de actividades realizadas por los trabajadores y que producen un resultado de valor observable.

RUP agrupa las actividades en grupos lógicos definiéndose 9 flujos de trabajo principales. Los 6 primeros son flujos de ingeniería del desarrollo del software y los tres últimos de apoyo.

Flujos de trabajo

1. **Modelación del negocio:** Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.

2. **Requerimientos:** Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.

3. **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.

4. **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.

5. **Prueba (Testeo):** Es una actividad de control de la calidad que busca los defectos a los largo del ciclo de vida de un producto.

6. **Despliegue**: Produce release del producto y realiza actividades (empaque, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.

7. **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.

8. **Administración de proyecto:** Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.

9. **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.

Fases

1. **Concepción o Inicio:** Se describe el negocio y se delimita el proyecto, describiendo sus alcances con la identificación de los casos de uso del sistema. En esta fase se debe tener en cuenta cuál es el objetivo del proyecto, si es factible, si se construye o se compra, cuánto cuesta, además de explorar si se continúa con el proyecto o se deja. Se debe mostrar una arquitectura candidata, estimar el coste en recursos de todo el proyecto, estimar los riesgos y las fuentes de incertidumbre.

2. **Elaboración:** El propósito de esta fase es analizar el dominio del problema, definir, validar y cimentar la arquitectura, completar la visión, desarrollar el Plan de Desarrollo del Software y eliminar

los mayores riesgos. En esta fase se obtiene un prototipo de la arquitectura, el que debe evolucionar hasta convertirse en el producto final, se debe demostrar que la arquitectura propuesta soportará la visión con un coste y tiempo razonable. Cuando termina esta fase se llega al punto de no retorno del proyecto.

3. **Construcción**: La finalidad de esta fase es alcanzar capacidad operacional del producto de forma incremental, a través de las sucesivas iteraciones. Además se pretende minimizar los costes de desarrollo mediante la optimización de recursos y evitando el tener que rehacer el trabajo o rechazarlo, conseguir la calidad adecuada y versiones funcionales que han pasado las pruebas realizadas. Se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario

4. **Transición:** El producto ya está listo para su instalación en las condiciones reales. Puede implicar reparación de errores.

Características del ciclo de vida de RUP

1. **Dirigido por casos de uso:** Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo).

2. **Centrado en la arquitectura:** La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente.

3. **Iterativo e Incremental:** RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Es práctico dividir el trabajo en partes más pequeñas o miniproyectos. Cada miniproyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en los flujos de trabajo y los incrementos, al crecimiento del producto. Cada iteración se realiza de forma planificada es por eso que se dice que son miniproyectos.

La etapa de construcción de la seguridad se llevara a cabo a lo largo de los primeros cinco flujos de trabajo de RUP (Modelación del negocio, Requerimientos, Análisis y Diseño, Implementación y Prueba), todo esto llevado a cabo en todas la fases de RUP. Todo esto implementando el SDL (Ciclo de vida de desarrollo de seguridad) de Microsoft, pero de manera modificada ya que Microsoft no utiliza RUP.

3.6.2.2 Introducción al ciclo de vida de desarrollo de seguridad.

La experiencia en seguridad del software real ha permitido establecer una serie de principios de alto nivel para lograr un software más seguro. Se hace referencia a estos principios como SD3+C: Seguro por diseño, Seguro por definición, Seguro en distribución y Comunicaciones. A continuación, se incluye una breve definición de estos principios: [58]

- Seguro por diseño: la arquitectura, el diseño y la implementación del software se deben realizar de manera que proteja tanto el software como la información que procesa, además de poder resistir ataques.
- Seguro por definición: en el mundo real, el software no es nunca totalmente seguro, por lo que los diseñadores deben asumir que habrá errores de seguridad. Para minimizar los daños que se producirán cuando los atacantes descubran estos errores, el estado predeterminado del software debe elegir las opciones más seguras. Por ejemplo, el software debe ejecutarse con los mínimos privilegios necesarios y los servicios y las características que no sean necesarios de manera habitual deben deshabilitarse de manera predeterminada o establecer que sólo unos pocos usuarios puedan tener acceso a ellos.
- Seguro en distribución: se debe incluir con el software información y herramientas que ayuden a los administradores y a los usuarios a utilizar este software con seguridad. Además, la implementación de las actualizaciones debe ser sencilla
- Comunicaciones: los programadores de software deben estar preparados para detectar las vulnerabilidades de seguridad del producto y deben comunicarse de manera abierta y responsable con los usuarios y los administradores para ayudarles a tomar las medidas de protección adecuadas (como la actualización o la implementación de soluciones alternativas).

Aunque todos los elementos de SD3+C imponen ciertos requisitos durante el proceso de desarrollo, los dos primeros elementos, seguro por diseño y seguro por definición, son los que más favorecen la

seguridad. Seguro por diseño obliga a utilizar procesos que tratan de evitar la inclusión de vulnerabilidades de seguridad desde el principio, mientras que Seguro por definición exige que la exposición predeterminada del software, la "superficie de ataque", sea la mínima posible.

La incorporación de las medidas de seguridad que pretende integrar el paradigma SD3+C en el proceso de desarrollo existente conduce a la organización global del proceso.

3.6.2.3 El proceso de ciclo de vida de desarrollo de seguridad.

Toda organización que quiera desarrollar software seguro deberá asumir la responsabilidad de asegurarse de que sus empleados adquieren los conocimientos necesarios en temas de seguridad y esto es de vital importancia a la hora de poner en práctica el SDL ya que se necesita para su aplicación de personal capacitado en el tema. Por lo que la primera etapa del procedimiento es de vital importancia ya que le da continuidad a la segunda y sin una adecuada realización y culminación de la etapa de planificación no se obtendrá en la etapa de construcción un software seguro.

3.6.2.4 Flujo de Trabajo Modelación del Negocio.

Un sistema, por pequeño que sea, generalmente es complicado. Por eso se necesita dividirlo en piezas para comprender y gestionar su complejidad. Esas piezas se pueden representar a través de modelos que permitan abstraer sus características esenciales.

Una técnica para la especificación de los requerimientos más importantes del sistema, que da soporte al negocio, es el modelo del negocio, con lo cual se refuerza la idea de que sea el propio negocio lo que determine los requerimientos. Uno de los modelos útiles previo al desarrollo de un software es el Modelo del Negocio.

Por lo que es muy importante la realización de la seguridad en esta etapa inicial ya que uno de los principios de los sistemas seguros es considerar la seguridad “de abajo para arriba”, ya que si al inicio logramos obtener una buena seguridad serán menos los fallos en las demás etapas y teniendo en cuenta que muchos proyectos de desarrollo generan la siguiente versión a partir de la anterior.

Propósito

La Modelación del Negocio tiene como propósito, comprender la estructura y la dinámica de la organización en la cual se va a implantar un sistema, comprender los problemas actuales de la organización e identificar las mejoras potenciales, asegurar que los consumidores, usuarios finales y desarrolladores tengan un entendimiento común de la organización y derivar los requerimientos del sistema que va a soportar la organización.

Para lograr esos propósitos, el proceso de Modelación del Negocio permite obtener una visión de la organización que permita definir los procesos, roles y responsabilidades de la organización en los modelos de casos de uso del negocio y de objetos.

Ya implementada una seguridad de la información con que trabajamos en la etapa inicial, la principal medida de seguridad que se debe llevar a cabo en este flujo de trabajo es la máxima comunicación entre los miembros del equipo y los clientes.

3.6.2.5 Flujo de Trabajo de Requerimientos.

La identificación de requerimientos es el punto de partida en el proceso de desarrollo de software. Es necesario lograr una comunicación efectiva entre los usuarios y el equipo de proyecto, con el objetivo de llegar a un entendimiento de lo que hay que hacer, esta es la clave del éxito en la producción de un software seguro.

Propósito.

La disciplina de Requerimientos tiene como propósito establecer y mantener acuerdos con los clientes y stakeholders en lo que debe hacer el sistema, así como proveer a los desarrolladores de un mejor entendimiento de los requisitos del mismo. Definir los límites del sistema, proporcionar una base para planear el contenido técnico de iteraciones y estimar coste y tiempo de desarrollo del sistema. Definir una interfaz de usuario para el sistema, centrándose en las necesidades y las metas de los usuarios.

Durante la fase de requisitos, el equipo de producto se pone en contacto con el equipo de seguridad central para solicitar la asignación de un asesor de seguridad (conocido como el encargado de la seguridad en la implementación del SDL) que actúa como punto de contacto, recurso y guía a través de los procedimientos de planeamiento.

“El equipo de seguridad ayuda al equipo de producto revisando los planes, aportando recomendaciones y asegurándose de que se planteen los recursos necesarios para la obtención de la

seguridad. Este advierte al equipo de producto de los puntos básicos de seguridad y los criterios de salida que serán necesarios en función del tamaño, la complejidad y los riesgos del proyecto.

La fase de requisitos es la oportunidad ideal para que el equipo de producto y el equipo de seguridad se planteen cómo se integrará la seguridad en el proceso de desarrollo, identifique los objetivos de seguridad clave y, por lo demás, maximice la seguridad del software procurando minimizar el impacto sobre los planes y los programas.

Como parte de este proceso, el equipo debe considerar cómo se integrarán las características de seguridad y las medidas de control con otros programas que probablemente se utilizarán con el software que están desarrollando. (El funcionamiento con otros programas es vital para responder a la necesidad de los usuarios de integrar los productos en sistemas seguros.) La consideración general por parte del equipo de producto de los objetivos, los retos y los planes de seguridad debe reflejarse en los documentos de planeamiento generados durante la fase de requisitos. Aunque es posible que estos planes cambien a medida que el proyecto avanza, articularlos desde el principio garantiza que no se pasa por alto ningún requisito ni surgen sorpresas de última hora.” [59]

Cada equipo de producto debe considerar los requisitos de características de seguridad como parte de esta fase. Aunque algunos requisitos de características de seguridad aparecerán a partir del modelo de amenazas, es probable que sean los requisitos de los usuarios los que dictaminen la inclusión de características de seguridad como respuesta a las demandas de los clientes. Los requisitos de características de seguridad también surgirán a partir de la necesidad de cumplir los estándares del sector y los procesos de certificación, como los criterios comunes. El equipo de producto debe detectar y reflejar estos requisitos como parte de su proceso de planeamiento normal.

3.6.2.6 Flujo de trabajo de Análisis y Diseño.

La fase de análisis y diseño identifica la estructura y los requisitos globales del software. Desde el punto de vista de la seguridad, los elementos clave de la fase de análisis y diseño son: [60]

- Definir la arquitectura de seguridad y las directrices de diseño: definir la estructura global del software desde el punto de vista de la seguridad e identificar los componentes cuyo correcto funcionamiento es esencial para la seguridad (la "base de computación confiable"). La

identificación de técnicas de diseño, como el uso de capas o lenguaje con tipos inflexibles, la aplicación de privilegios mínimos y la minimización de la superficie de ataque, que se aplican al software de manera global. (El uso de capas se refiere a la organización del software en componentes bien definidos que se estructuran para evitar dependencias circulares entre componentes. Los componentes se organizan en capas y una capa superior puede depender de los servicios de capas inferiores, pero se prohíbe que las capas inferiores dependan de las capas superiores.) Los detalles específicos de cada uno de los elementos de la arquitectura se indican en las especificaciones de diseño individuales, pero la arquitectura de seguridad corresponde a una perspectiva global sobre el diseño de seguridad.

- Documentar los elementos de la superficie de ataque del software. Teniendo en cuenta que el software no logrará una seguridad perfecta, es importante que únicamente se expongan de manera predeterminada las características que utilicen la mayoría de los usuarios y que dichas características se instalen con el mínimo nivel de privilegios posible. La medición de los elementos de la superficie de ataque ofrece al equipo de producto un indicador continuo de la seguridad predeterminada y les permite detectar las instancias en las que el software es más susceptible de recibir ataques. Aunque algunas instancias con mayor superficie de ataque pueden estar justificadas por una mayor facilidad de uso o unas mejores funciones del producto, es importante detectar y considerar cada una de estas instancias durante el diseño y la implementación para lanzar el software con la configuración predeterminada más segura posible.
- Realizar un modelado de las amenazas. El equipo debe realizar un modelado de amenazas por componentes. Mediante una metodología estructurada, el equipo de componentes identifica los activos que debe administrar el software y las interfaces que permitirán el acceso a dichos activos. El proceso de modelado de amenazas identifica las amenazas que pueden dañar a estos activos y la probabilidad de que se inflija dicho daño (estimación del riesgo). A continuación, el equipo de componente identifica las contramedidas que pueden mitigar el riesgo, ya sea mediante características de seguridad (por ejemplo, el cifrado) o mediante un funcionamiento correcto del software que proteja a los activos del daño. Por tanto, el modelado de amenazas ayuda al equipo de producto a identificar las necesidades de características de seguridad y las áreas en las que es necesario revisar con especial minuciosidad el código y probar la seguridad. El proceso de modelado de amenazas debe realizarse con una

herramienta capaz de capturar modelos de amenazas en un formato que pueda leer un equipo para almacenarlo y actualizarlo.

- Definir los criterios de publicación adicionales. Aunque los criterios de publicación de seguridad básicos deben definirse para toda la organización, puede que existan criterios concretos para determinados equipos de producto o lanzamientos de software que sea preciso cumplir para poder lanzar el software. Por ejemplo, un equipo de producto dedicado al desarrollo de una versión actualizada del software que se enviará a los clientes y que está expuesta a un gran número de ataques puede optar por exigir que la nueva versión no presente ninguna de las vulnerabilidades de seguridad detectadas durante cierto tiempo antes de considerar que está lista para su lanzamiento. (Es decir, el proceso de desarrollo debe descubrir las vulnerabilidades de seguridad y solucionarlas antes de que se detecten, en vez de tener que solucionarlas después de su detección.

3.6.2.7 Flujo de trabajo de implementación.

"Durante el flujo de trabajo de implementación, el equipo del producto programa, prueba e integra el software. Los pasos destinados a eliminar los errores de seguridad o a impedir que se incluyan desde el principio son de gran utilidad, ya que reducen considerablemente la probabilidad de que las vulnerabilidades de seguridad lleguen a la versión final del software que se lanzará a los clientes.

Los resultados del modelado de amenazas ofrecen una orientación especialmente importante durante la fase de implementación. Los programadores deben asegurarse de que escriben correctamente el código para mitigar las amenazas de alta prioridad, mientras que los encargados de las pruebas deberán asegurarse de que estas amenazas se han bloqueado o mitigado de manera efectiva.

Los elementos del SDL que se aplican en la fase de implementación son:

- Aplicar estándares de codificación y de pruebas. Los estándares de codificación evitan que los programadores incluyan errores que puedan producir vulnerabilidades de seguridad. Por ejemplo, el uso de construcciones de manipulación de búferes y de manejo de cadenas más seguras y coherentes ayuda a evitar la aparición de vulnerabilidades de seguridad de saturación del búfer. Los estándares de pruebas y las prácticas recomendadas permiten garantizar que las pruebas se centran en detectar posibles vulnerabilidades de seguridad, en

vez de centrarse únicamente en el funcionamiento correcto de las características y las funciones del software.

- Aplicar herramientas de comprobación de seguridad, incluidas herramientas de confusión. Estas herramientas ofrecen entradas estructuradas pero no válidas a las interfaces de programación de aplicaciones (API) de software y a las interfaces de red para maximizar la probabilidad de detectar errores que puedan ocasionar vulnerabilidades de seguridad del software.
- Aplicar herramientas de exploración del código de análisis estático. Las herramientas pueden detectar algunos tipos de errores de codificación que producen vulnerabilidades de seguridad, incluidas saturaciones de búfer, desbordamientos con enteros y variables no inicializadas.
- Realizar revisiones del código. Las revisiones del código complementan las herramientas automatizadas y las pruebas, ya que aplican el esfuerzo de programadores expertos para examinar el código fuente y detectar y eliminar posibles vulnerabilidades de seguridad. Estas revisiones constituyen un paso fundamental para eliminar las vulnerabilidades de seguridad del software durante el proceso de desarrollo."[61]

Además de todo lo que plantea el SDL también se quiere mediante este procedimiento poner en practica en este flujo de trabajo de implementación el nuevo paradigma de la programación POA (Programación Orientada a Aspectos) como otra paso para la confección de un software seguro. A continuación se expone lo que se debe llevar a cabo para la implantación del paradigma.

Programación Orientada a Aspectos

"Existen conceptos que no pueden ser encapsulados con las metodologías de programación actuales dentro de una unidad funcional, debido a que atraviesan todo el sistema, o varias partes de él. Algunos de estos conceptos son: sincronización, manejo de memoria, distribución, chequeo de errores, profiling, seguridad o redes, entre otros.

Aquí se muestran algunos ejemplos:

1) Consideremos una aplicación que incluya conceptos de seguridad y sincronización, como por ejemplo, asegurarnos que dos usuarios no intenten acceder al mismo dato al mismo tiempo. Ambos conceptos requieren que los programadores escriban la misma funcionalidad en varias partes de la aplicación. Los programadores se verán forzados a recordar todas estas partes, para que a la hora de efectuar un cambio

y / o una actualización puedan hacerlo de manera uniforme a través de todo el sistema. Tan solo olvidarse de actualizar algunas de estas repeticiones lleva al código a acumular errores.

2) Manejo de errores y de fallas: agregar a un sistema simple un buen manejo de errores y de fallas requiere muchos y pequeños cambios y adiciones por todo el sistema debido a los diferentes contextos dinámicos que pueden llevar a una falla, y las diferentes políticas relacionadas con el manejo de una falla .

3) En general, los aspectos en un sistema que tengan que ver con el atributo performance, resultan diseminados por todo el sistema. Se puede afirmar entonces que las técnicas tradicionales no soportan de una manera adecuada la separación de las propiedades de aspectos distintos a la funcionalidad básica, y que esta situación tiene un impacto negativo en la calidad del software.

Como respuesta a este problema nace la Programación Orientada a Aspectos. La POA permite a los programadores escribir, ver y editar un aspecto diseminado por todo el sistema como una entidad por separado, de una manera inteligente, eficiente e intuitiva. La POA es una nueva metodología de programación que aspira a soportar la separación de las propiedades para los aspectos antes mencionados. Esto implica separar la funcionalidad básica y los aspectos, y los aspectos entre sí, a través de mecanismos que permitan abstraerlos y componerlos para formar todo el sistema.

“La idea central que persigue la POA es permitir que un programa sea construido describiendo cada concepto separadamente. El soporte para este nuevo paradigma se logra a través de una clase especial de lenguajes, llamados lenguajes orientados a aspectos (LOA), los cuales brindan mecanismos y constructores para capturar aquellos elementos que se diseminan por todo el sistema. A estos elementos se les da el nombre de aspectos. Se define entonces a un aspecto como un concepto que no es posible encapsularlo claramente, y que resulta diseminado por todo el código. Los aspectos son la unidad básica de la programación orientada a aspectos.

Los tres principales requerimientos de la POA son:

- Un lenguaje para definir la funcionalidad básica, conocido como lenguaje base o componente. Podría ser un lenguaje imperativo, o un lenguaje no imperativo (C++, Java, Lisp, ML).
- Uno o varios lenguajes de aspectos, para especificar el comportamiento de los aspectos. (COOL, para sincronización, RIDL, para distribución, AspectJ de propósito general.)

- Un tejedor de aspectos (del inglés weaver), que se encargará de combinar los leguajes .Tal proceso se puede retrasar para hacerse en tiempo de ejecución o en tiempo de compilación.

Los lenguajes orientados a aspectos definen una nueva unidad de programación de software para encapsular aquellos conceptos que cruzan todo el código. A la hora de "tejer" los componentes y los aspectos para formar el sistema final, es claro que se necesita una interacción entre el código base y el código de los aspectos. También es claro que esta interacción no es la misma que ocurre entre los módulos del lenguaje base, donde la comunicación está basada en declaraciones de tipo y llamadas a procedimientos y funciones. La POA define entonces una nueva forma de interacción, provista a través de los puntos de enlace. Los puntos de enlace brindan la interfaz entre aspectos y componentes; son lugares dentro del código donde es posible agregar el comportamiento adicional que destaca a la POA. Dicho comportamiento adicional es especificado en los aspectos. Dado un punto de enlace *pde*, este comportamiento adicional puede agregarse, en general, en tres momentos particulares:

- "antes de *pde*".

- "después de *pde*"

- "en lugar de *pde*"

Por ejemplo, dentro de la POO, algunos puntos de enlace serían: llamadas a métodos, creación de objetos, acceso a atributos, etc. Aún nos falta introducir el encargado principal en el proceso de la POA. Este encargado principal conocido como tejedor debe realizar la parte final y más importante: tejer" los diferentes mecanismos de abstracción y composición que aparecen tanto en "los lenguajes de aspectos como en los lenguajes de componentes, guiado por los puntos de enlace. La estructura general de una implementación basada en aspectos es análoga a la estructura de una implementación tradicional. Una implementación tradicional consiste de:

- Un lenguaje.

- Un compilador o intérprete para ese lenguaje.

- Un programa escrito en ese lenguaje que implemente la aplicación.

Una implementación basada en la Programación Orientada a Aspectos consiste en:

- El lenguaje base o componente para programar la funcionalidad básica.

- Uno o más lenguajes de aspectos para especificar los aspectos.

- Un tejedor de aspectos para la combinación de los lenguajes.

- El programa escrito en el lenguaje componente que implementa los componentes.

- Uno o más programas de aspectos que implementan los aspectos.

Gráficamente, se puede comparar ambas estructuras, como queda reflejado en la Figura 3.4.

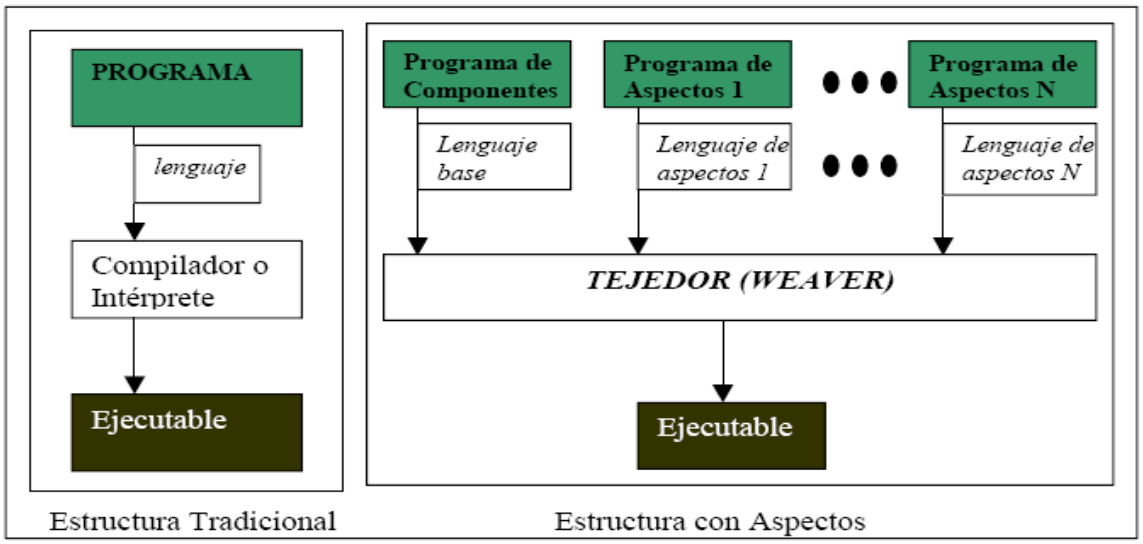


Figura 3.4: Comparación entre las estructuras tradicionales y las estructuras con aspectos.

Impacto de la POA en la seguridad

La POA nos permite encapsular y abstraer el concepto de seguridad. Con esto se

Logra :

- Bajo costo de mantenimiento: al tener todo el código relativo a la seguridad en una única unidad se simplifica la introducción de cambios.
- Mayor flexibilidad: al estar la seguridad implementada de forma independiente, es mucho más sencillo realizar cambios en las políticas de seguridad.
- Mayor especialización: los desarrolladores pueden implementar la funcionalidad básica, mientras que un experto en seguridad puede especificar las propiedades de seguridad.
- Independencia de tres niveles:
    1) La seguridad es independiente de la funcionalidad básica.
    2) La seguridad es independiente de los implementadores.
    3) La seguridad es independiente del momento en que se la genera.

También, se debe reconocer que existen ciertos puntos débiles. Por un lado, la POA es un paradigma que recién está naciendo, y por lo tanto continuamente surgen nuevos problemas. Los lenguajes orientados a aspectos no son todavía herramientas maduras ni libres de errores. Los investigadores afirman que el estado actual de la POA es similar al estado en que se encontraba la POO hace 20 años. Además, tampoco tiene la POA un respaldo teórico importante que lo sustente, sino que padece un estado de informalidad acentuado." [62]

3.6.2.8 Flujo de trabajo de Prueba.

"El flujo de trabajo de prueba es el punto en el que software ya incorpora toda la funcionalidad y los usuarios pueden comenzar a probar la versión beta. Durante este flujo de trabajo, mientras se prueba la versión beta del software, el equipo de producto realiza una campaña de seguridad que incluye revisiones del código de seguridad aparte de las realizadas en el flujo de trabajo de implementación, así como la realización de pruebas centradas en la seguridad. Además de que se realiza una campaña de seguridad.

Existían dos motivos para introducir la campaña de seguridad en el proceso:

- El ciclo de vida del software de las versiones en cuestión había alcanzado la fase de comprobación, que era un punto adecuado para realizar las revisiones del código y las pruebas necesarias.
- La realización de la campaña de seguridad durante la fase de comprobación asegura que la revisión del código y las pruebas se realizan con la versión terminada del software y ofrece una oportunidad de revisar tanto el código desarrollado o actualizado durante la fase de implementación como el código heredado que no se ha modificado.

El primero de estos motivos refleja un accidente histórico: la decisión de iniciar una campaña de seguridad se tomó en principio durante el flujo de trabajo de prueba. Es una buena idea realizar una campaña de seguridad durante esta fase, tanto para asegurar que el software final cumple los requisitos como para permitir una revisión en detalle de todo el código heredado de versiones anteriores del software. "[63]

Hay que resaltar que las revisiones del código y las pruebas del código de alta prioridad (código que forma parte de la "superficie de ataque" del software) son esenciales para varias partes del SDL. Por ejemplo, estas revisiones y pruebas deben ser obligatorias en el flujo de trabajo de implementación para corregir cuanto antes los problemas, así como para identificar y corregir los orígenes de dichos problemas. También son fundamentales en el flujo de trabajo de prueba cuando esté a punto de finalizarse.

A continuación se proponen pruebas de seguridad para la etapa final del software y que se pueden llevar a cabo en el área de calidad de la facultad para dar por concluido un producto.

Prueba de Seguridad en cuanto al control de acceso:

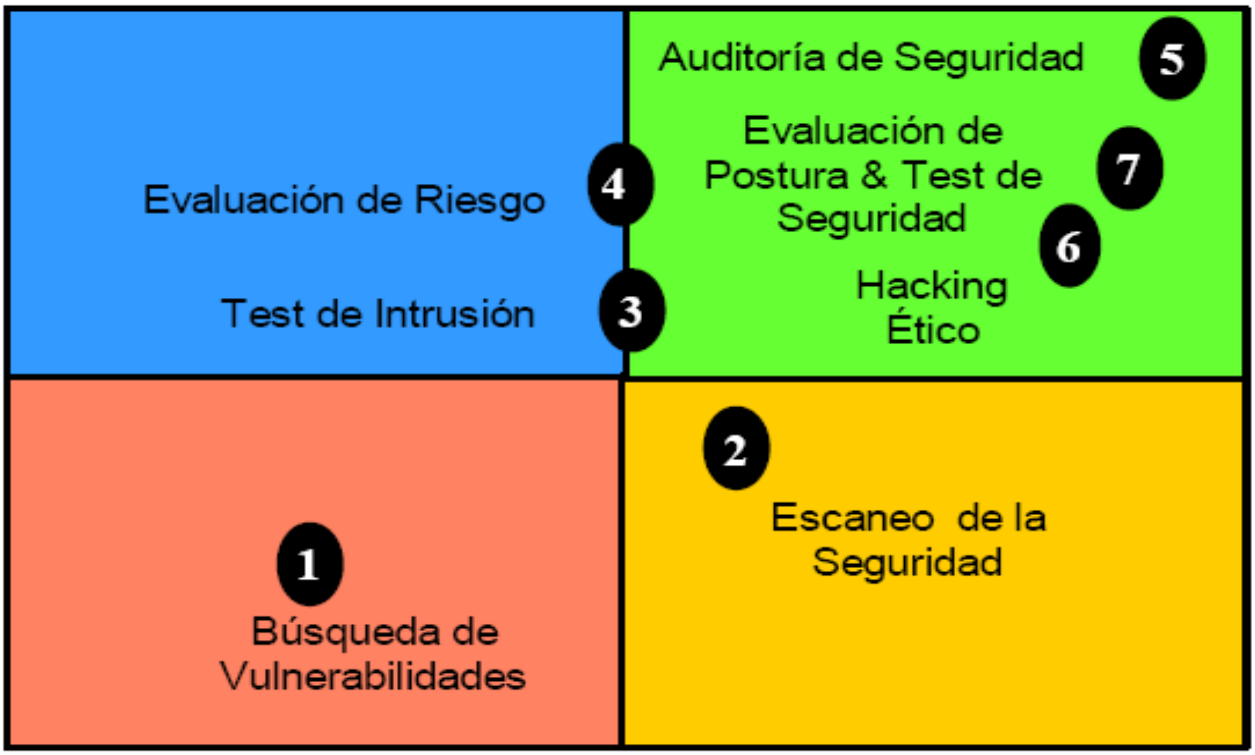
Esta prueba se enfoca en dos áreas de seguridad:

- **Seguridad en el ámbito de aplicación, incluyendo el acceso a los datos y a las funciones de negocios:** verifica que un actor pueda acceder solo a las funciones o datos para los cuales su tipo de usuario tiene permiso. Para ello se debe identificar y hacer una lista de cada tipo de usuario y las funciones y datos sobre las que cada tipo tiene permiso; crear pruebas para cada tipo de usuario y verificar cada permiso creando operaciones específicas para cada tipo de usuario y modificar el tipo de usuario y volver a ejecutar las pruebas para los mismos usuarios.

En cada caso, verificar que las funciones o datos adicionales están correctamente disponibles o son denegados.

- **Seguridad en el ámbito de sistema, incluyendo conexión, o acceso remoto al sistema:** verificar que solo los actores con acceso al sistema y a las aplicaciones, puedan acceder a ellos. Para esto el acceso al sistema debe ser discutido con el administrador del sistema o la red. Esta prueba no puede requerirse como tal, es una función del administrador del sistema o de la red.

Además de las pruebas de seguridad de control de acceso se llevara a cabo un proceso para todo el software en cuanto a pruebas de seguridad, implementado por el Instituto para la Seguridad y las Metodologías Abiertas (ISECOM) con el cual el instituto ha obtenido grandes resultados en cuanto a pruebas de seguridad ya que en el se recoge todo lo que se puede hacer en una prueba de seguridad llamado Manual de la Metodología Abierta de Testeo de Seguridad (MMATS).



1. **Búsqueda de Vulnerabilidades:** Se refiere generalmente a la búsqueda de todas las vulnerabilidades conocidas dentro del producto desarrollo, mediante en plan elaborado por el responsable de seguridad del modulo.

2. **Escaneo de la Seguridad:** Se refiere en general a las búsquedas de vulnerabilidades que incluyen verificaciones manuales de falsos positivos, identificación de los puntos débiles de la red y análisis profesional individualizado.

3. **Test de Intrusión:** Se centra en evaluar la seguridad de los sistemas de protección perimetral de una empresa así como los diferentes sistemas que están accesibles desde Internet (routers exteriores, firewall exterior, servidores web, de correo, de noticias, etc.). Intentando penetrar en ellos y de esta forma alcanzar zonas de la red de una empresa como puede ser la red interna o la DMZ (Zona Desmilitarizada).

4. Evaluación de Riesgo: Se refiere a los análisis de seguridad a través de entrevistas e investigación de nivel medio que incluye la justificación negocios, las justificaciones legales y las justificaciones específicas de la industria.

5. **Auditoría de Seguridad:** Hace referencia a la inspección manual con privilegios administrativos del sistema operativo y de los programas de aplicación del sistema o sistemas dentro de una red o redes.

6. **Hacking Ético**: Se refiere generalmente a los test de intrusión en los cuales el objetivo es obtener trofeos en la red dentro del tiempo predeterminado de duración del proyecto.

7. **Test de Seguridad y su equivalente militar, Evaluación de Postura**, es una evaluación de riesgo con orientación de proyecto de los sistemas y redes, a través de la aplicación de análisis profesional mediante escaneos de seguridad donde la intrusión se usa generalmente para confirmar los falsos positivos y los falsos negativos dentro del tiempo permitido de duración del proyecto.

3.6.3 3ra Etapa: Supervisión de la Seguridad.

La etapa de supervisión de la seguridad se llevara a cabo por el encargado de la seguridad en todos los niveles dentro de cada área temática y supervisado por el grupo de seguridad a nivel de facultad. Todo este proceso de supervisión de la seguridad se realizara a través de una auditoria de seguridad.

Una auditoría de la seguridad informática es la revisión y la evaluación de los controles, sistemas, procedimientos de informática; de los equipos de cómputo, su utilización, eficiencia, de la organización de los que participan en el procesamiento de la información.

La auditoría de la seguridad informática deberá comprender no sólo la evaluación de los equipos de cómputo, de un sistema o procedimiento específico, sino que además habrá de evaluar los sistemas de información en general desde sus entradas, procedimientos, controles, archivos, seguridad y obtención de información.

La auditoría de la seguridad informática es de vital importancia para el buen desempeño de los sistemas de información, ya que proporciona los controles necesarios para que los sistemas sean confiables y con un buen nivel de seguridad. Además debe evaluar todo (informática, organización de centros de información, hardware y software).

Fases de una auditoría

Los servicios de auditoría constan de las siguientes fases:

- Enumeración de redes, topologías y protocolos
- Identificación de sistemas y dispositivos
- Identificación de los sistemas operativos instalados
- Análisis de servicios y aplicaciones
- Detección, comprobación y evaluación de vulnerabilidades
- Medidas específicas de corrección
- Recomendaciones sobre implantación de medidas preventivas.

Tipos de Auditoría:

Los servicios de auditoría pueden ser de distinta índole:

Auditoría de seguridad interna. En este tipo de auditoría se contrasta el nivel de seguridad y privacidad de las redes locales y corporativas de carácter interno

Auditoría de seguridad perimetral. En este tipo de análisis, el perímetro de la red local o corporativa es estudiado y se analiza el grado de seguridad que ofrece en las entradas exteriores

Test de intrusión. El test de intrusión es un método de auditoría mediante el cual se intenta acceder a los sistemas, para comprobar el nivel de resistencia a la intrusión no deseada. Es un complemento fundamental para la auditoría perimetral.

Análisis forense. El análisis forense es una metodología de estudio ideal para el análisis posterior de incidentes, mediante el cual se trata de reconstruir cómo se ha penetrado en el sistema, a la par que se valoran los daños ocasionados. Si los daños han provocado la inoperatividad del sistema, el análisis se denomina análisis postmortem.

Auditoría de páginas web. Entendida como el análisis externo de la web, comprobando vulnerabilidades como la inyección de código SQL, Verificación de existencia y anulación de posibilidades de Cross Site Scripting (XSS), etc.

Auditoría de código de aplicaciones. Análisis del código tanto de aplicaciones páginas Web como de cualquier tipo de aplicación, independientemente del lenguaje empleado.

Se deben de poner en práctica todos estos tipos de auditorias ya que cada una de ellas tiene una característica diferente que evaluar, en caso que alguna no se ponga en practica debe ser documentado el por que por el grupo de seguridad.

Realizar auditorías con cierta frecuencia asegura la integridad de los controles de seguridad aplicados a los sistemas de información. Acciones como el constante cambio en las configuraciones, la instalación de parches, actualización del software y la adquisición de nuevo hardware hacen necesario que los sistemas estén continuamente verificados mediante auditoría.

Estándares orientados hacia las auditorias de la seguridad informática

Una auditoría se realiza con base a un patrón o conjunto de directrices o buenas practicas sugeridas. Existen estándares orientados a servir como base para auditorias de informática. Uno de ellos es CoBIT (Objetivos de Control de la Tecnologías de la Información), dentro de los objetivos definidos como parámetro, se encuentra el "Garantizar la Seguridad de los Sistemas". Adicional a este estándar podemos encontrar el estándar ISO 27002, el cual se conforma como un código internacional de buenas prácticas de seguridad de la información, este puede constituirse como una directriz de auditoria apoyándose de otros estándares de seguridad de la información que definen los requisitos de auditoria y sistemas de gestión de seguridad, como lo es el estándar ISO 27001.

3.6.4 4ta Etapa: Revisión Final de Seguridad.

“Cuando se da por concluido el producto, el software debe someterse a una revisión final de seguridad (RFS). Esta RFS debe responder a la siguiente pregunta: "Desde el punto de vista de la seguridad, ¿está este software preparado para los clientes?" La RFS se realiza en un plazo de dos a seis meses antes de la finalización del software, según el alcance del software. El software debe ser estable antes de la RFS y es de esperar que antes del lanzamiento sólo se realicen cambios mínimos y no relacionados con la seguridad.

La RFS es una revisión independiente del software que realiza el equipo de seguridad central de la organización. El asesor de seguridad del equipo de seguridad aconseja al equipo de producto sobre el

ámbito de la RFS que requiere el software y ofrece al equipo de producto una lista de los requisitos de recursos antes de la RFS. El equipo de producto proporciona al equipo de seguridad los recursos y la información necesarios para llevar a cabo la RFS Al comienzo de la RFS, el equipo de producto debe rellenar un cuestionario y entrevistarse con un miembro del equipo de seguridad asignado a la RFS. En toda RFS se deben revisar los errores que se identificaron en un principio como errores de seguridad, pero que tras analizarlos se consideró que no afectaban a la seguridad, para asegurarse de que este análisis es correcto. Una RFS también incluye una revisión de la capacidad del software para soportar vulnerabilidades de seguridad detectadas recientemente en un software similar. Una RFS para una versión de software importante requerirá realizar pruebas de penetración y, posiblemente, recurrir a asesores de revisión de seguridad externos que ayuden al equipo de seguridad.

La RFS no es únicamente un examen que se puede aprobar o suspender ni tampoco pretende detectar todas las vulnerabilidades de seguridad que quedan en el software, lo que no sería factible, sino proporcionar al equipo de producto y a la administración superior de la organización una idea global del nivel de seguridad del software y de la probabilidad de que pueda resistir ataques una vez que se haya entregado a los clientes. Si la RFS detecta patrones de vulnerabilidades de seguridad restantes, no bastará con solucionar las vulnerabilidades detectadas, sino que habrá que repetir la fase anterior y tomar las acciones necesarias para tratar los orígenes (por ejemplo, mejorar los conocimientos, mejorar las herramientas).” [64]

3.7 Conclusiones.

La investigación enunciada en el capitulo anterior arrojo como conclusiones que:

- Para realizar la propuesta de procedimiento de seguridad se debe cumplir con todas las etapas descritas en el procedimiento, sin obviar ninguna ya que la inferior depende de la superior.
- El equipo de seguridad debe estar en constante intercambio con el equipo de desarrollo del software.
- El procedimiento propuesto tiene concordancia directa con el ciclo de vida de desarrollo de seguridad que propone Microsoft.

Conclusiones

- Se elaboró un procedimiento que enfoca la seguridad informática en el proceso de desarrollo de Software.
- Se aplicaron estándares internacionales establecidos para la gestión de la seguridad en el Software.
- En la investigación se identificaron las pruebas de seguridad que pueden aplicarse en el proceso de calidad.

Recomendaciones

- Aplicar, validar y dar seguimiento al procedimiento propuesto en los proyectos de la facultad 7.
- Adaptar el procedimiento a otras metodologías de desarrollo como por ejemplo la metodología XP.
- Aplicar las pruebas de seguridad propuestas en la facultad en el proceso que realiza el proyecto de calidad.

Referencia Bibliográfica

1. Asteasuain Fernando, Schmidt Leandro. Universidad Nacional del Sur. Bahía Blanca. Aplicación de la Programación Orientada a Aspectos como Solución a los Problemas de la Seguridad en el Software.

2. Steve Lipner, Michael Howard.” El ciclo de vida de desarrollo de seguridad de Trustworthy Computing”, Unidad tecnológica y empresarial de seguridad Microsoft Corporation, Marzo 2005.

3. Maria Lilia Rodríguez Batista, Propuesta de Procedimiento para el Aseguramiento de la Calidad de Software en los proyectos productivos de la Facultad 7, Junio 2007, Universidad de la Ciencias Informáticas.

4. Doshi Shreyas, “Software Engineering for Security: Towards ArchitectingSecure Software”, Information and Computer Science Dept., University ofCalifornia, Irvine, CA 92697, USA. Abril de 2001.

5. Ídem a la referencia 1.

6. Ídem a la referencia 1.

7. Ídem a la referencia 1.

8. John Viega, J.T. Bloch, Pravir Chandra, “Applying Aspect-OrientedProgramming to Security”, Cutter IT Journal, febrero de 2001.

9. John Viega, J.T. Bloch, Yoshi Kohno, Gary McGraw, “ITS4: A Static Vulnerability Scanner for C and C++ Code”, in Proceedings of the 16th Annual Computer Security Applications Conference (ACSAC 2000). IEEE, 2000.

10. Ídem a la referencia 1.

11. Ídem a la referencia 8.

12. Ídem a la referencia 4.

13. Ídem a la referencia 1.

14. Ídem a la referencia 1.

15. Ídem a la referencia 4.

16. Ídem a la referencia 1.

17. Ídem a la referencia 3.

18. AUNA Fundacion.Análisis y Perspectiva. Notas Noviembre 2005.[Citen Availabe from: www.fundacionauna.org].

19. Ídem a la referencia 3.

20. Estudio Global 2000-2004 sobre Piratería de Software de BSA, www.virusprot.com/Pirateri.html.

21. Ídem a la referencia 9.

22. IBM Internet Security Systems Executive Brief, IBM Internet Security Systems X-Force® 2006

Trend Statistics, Enero 2007, and www.iss.net.

23. Ídem a la referencia 22.

24. www.alertaantivirus.es/seguridad/info_vulns.html?menu=si. Abril 2008.

25. www.owasp.org/index.php/Main_Page. Abril 2008.

26. Ídem a la referencia 25.

27. http://www.ibiblio.org/pub/linux/docs/LuCaS/Manuales-LuCAS/doc-unixsec/unixsec-html/node332.html. 8 de Agosto del 2003.

28. Ídem a la referencia 27.

29. http://www.elpais.com. Abril 2008.

30. Ídem a la referencia 29.

31. http://www.macroseguridad.net/. Abril 2008.

32. http://www.deloitte.es/. Abril 2008.

33. http://www.digiware.com.co/. Abril 2008.

34. http://www.grippo.com.ar/empresas/?e=0068790. Abril 2008.

35. http://www.symantec.com/es/es/index.jsp. Abril 2008.

36. Ídem a la referencia 27.

37. Ídem a la referencia 3.

38. Ídem a la referencia 3.

39. http://www.iso.org/iso/home.htm. Abril 2008.

40. Ídem a la referencia 39.

41. Ídem a la referencia 1.

42. Ídem a la referencia 3.

43. Alina Ruiz (2007), La UCI y la Industria Cubana del SW, Presentación Informática 2007.

44. Ídem a la referencia 43.

45. Ídem a la referencia 3.

46. Ídem a la referencia 43.

47. http://www.one.cu/. Abril 2008

48. Ídem a la referencia 47.

49. Ídem a la referencia 3.

50. Ídem a la referencia 43.

51. Ídem a la referencia 3.

52. Rolando Alfredo Hernández León, Sayda Coello González. EL PARADIGMA CUANTITATIVO DE LA INVESTIGACIÓN CIENTÍFICA, UCI, Noviembre 2002.

53. Ídem a la referencia 52.

54. Ídem a la referencia 3.

55. http://sgsi-iso27001.blogspot.com/. Abril 2008.

56. Ídem a la referencia 55.

57. Ídem a la referencia 2.

58. Ídem a la referencia 2.

59. Ídem a la referencia 2.

60. Ídem a la referencia 2.

61. Ídem a la referencia 2.

62. Ídem a la referencia 1.

63. Ídem a la referencia 2.

64. Ídem a la referencia 2.

Bibliografía

1. Asteasuain Fernando, Bernardo Ezequiel Contreras, “Programación Orientada a Aspectos: Análisis del Paradigma”, Tesis de Licenciatura en Ciencias de la Computación. Universidad Nacional del Sur, noviembre de 2002.

2. Asteasuain Fernando, Schmidt Leandro. Universidad Nacional del Sur. Bahía Blanca. Aplicación de la Programación Orientada a Aspectos como Solución a los

Problemas de la Seguridad en el Software.

3. AUNA Fundacion.Análisis y Perspectiva. Notas Noviembre 2005.[Citen Availabe from: www.fundacionauna.org]

4. Alina Ruiz (2007), La UCI y la Industria Cubana del SW, Presentación Informática 2007.

5. Bart De Win, Bart Vanhaute, Bart De Decker, “Security through Aspect Oriented Programming”, in Proceedings of the 1st working conference on Network Security, noviembre de 2001.

6. Bedsy Guevara Mojena, Procedimiento Propuesto para medir la Calidad en la Gestión de Requisitos, Julio 2007, UCI.

7. Bianca María Torres Vega, Jenny Galindo Plasencia, PROCEDIMIENTOS PARA EL DISEÑO DE CASOS DE PRUEBAS DE SITIOS WEB EN LA UCI, Junio 2007, UCI.

8. Doshi Shreyas, “Software Engineering for Security: Towards ArchitectingSecure Software”, Information and Computer Science Dept., University ofCalifornia, Irvine, CA 92697, USA. Abril de 2001.

9. Estudio Global 2000-2004 sobre Piratería de Software de BSA, www.virusprot.com/Pirateri.html.

10. http:// www.alertaantivirus.es/seguridad/info_vulns.html?menu=si

11. http://www.deloitte.es/

12. http://www.digiware.com.co/

13. http://www.elpais.com.

14. http://www.grippo.com.ar/empresas/?e=0068790

15. http://www.ibiblio.org/pub/linux/docs/LuCaS/Manuales-LuCAS/doc-unixsec/unixsec-html/node332.html. 8 de Agosto del 2003.

16. http://www.iso.org/iso/home.htm.

17. http://www.macroseguridad.net/

18. http://www.sgsi-iso27001.blogspot.com/.

19. http://www.symantec.com/es/es/index.jsp

20. http://www.one.cu/.

21. http://www.owasp.org/index.php/Main_Page.

22. IBM Internet Security Systems Executive Brief, IBM Internet Security Systems X-Force® 2006 Trend Statistics, Enero 2007, www.iss.net.

23. Jacaboson, I., Booch, G., Rumbaugh J., El Proceso Unificado de Desarrollo de Software, Addison Wesley 2000.

24. John Viega, J.T. Bloch, Pravir Chandra, "Applying Aspect-OrientedProgramming to Security", Cutter IT Journal, febrero de 2001.

25. John Viega, J.T. Bloch, Yoshi Kohno, Gary McGraw, "ITS4: A Static Vulnerability Scanner for C and C++ Code", in Proceedings of the 16th Annual Computer Security Applications Conference (ACSAC 2000). IEEE, 2000.

26. Maria Lilia Rodríguez Batista, Propuesta de Procedimiento para el Aseguramiento de la Calidad de Software en los proyectos productivos de la Facultad 7, Junio 2007, Universidad de la Ciencias Informáticas.

27. Pressman, R, Ingeniería del Software: Un enfoque práctico, McGraw Hill 1997.

28. Premkumar T. Devanbu, Stuart Stubblebine, "Software Engineering for Security: a Roadmap", in Proceedings of the conference on The Future of Software Engineering. 2000.

29. Rolando Alfredo Hernández León, Sayda Coello González. EL PARADIGMA CUANTITATIVO DE LA INVESTIGACIÓN CIENTÍFICA, UCI, Noviembre 2002.

30. Steve Lipner, Michael Howard." El ciclo de vida de desarrollo de seguridad de Trustworthy Computing", Unidad tecnológica y empresarial de seguridad Microsoft Corporation, Marzo 2005.

31. Yadira Marrero Machín, Perla Mailen Tabasco Reyes. Procedimiento para el control y aseguramiento de la calidad en los flujos trabajo Modelación del Negocio y Requerimientos de los proyectos de software de la Facultad 2 de la Universidad de las Ciencias Informáticas. Junio 2007

32. Yelenys Delvoys Marchante, Mireilys Martínez Miranda, Propuesta de Procedimiento para la Revisión y Evaluación de un Producto de Software Terminado, Julio 2007, UCI.

Anexo

Anexo 1 Encuesta sobre la seguridad en el software en los Proyectos Productivos de la Facultad 7.

Nombre del Proyecto_____ Facultad_____

Rol del Encuestado_____ Tamaño del equipo desarrollo_____

Objetivos del Proyecto_____

Área Temática _____

1. ¿Se dedican solo a la producción del software? Si___ No___ ¿Qué otras actividades realizan?

2. ¿Cuántos software han realizado?_____

3. La seguridad de los productos que realizan es E__ B__ R__ M__

4. ¿Qué usted entiende por seguridad del software?_____

5. ¿Es necesario para usted garantizar la seguridad de los productos fabricados? Si__ No__ ¿Por qué?

6. ¿Los clientes se sienten satisfechos con los productos que le entregan? Si__ No__ No se__

7. Para conocer a cerca de la satisfacción de los clientes lo hacen a través de:

a. Encuestas___ b. Encuentros Planificados___ c. Encuentros casuales___

d. Correo Electrónico___ e. Vía Telefónica___ f. Visitas al cliente____

g. Por el servicio de Soporte de Software_____ h. Otros_____

8. ¿Utiliza metodología RUP? Si____ No____. Mencione otra en caso de que no sea RUP_____

9. ¿En el proyecto hay algún integrante del equipo que se dedique a la seguridad del producto?____

10. ¿Conoce algún modelo, estándar o paradigma que se utilice a nivel mundial para gestionar la seguridad en el software? Si____ No_____ No se____ Si su respuesta es si:

a. Menciónelo(s) _____

b. Utiliza alguno de estos modelos, estándares o paradigmas en su proyecto. Si___ No___ No se___

c. Menciónelo(s) _____

d. ¿Conocen algún procedimiento para la seguridad en el software? Si_____ No____ No se_____

e. ¿Aplica algún procedimiento para la seguridad en el software? Si_____ No_____ No se____

11. ¿Conoce las vulnerabilidades a las que puede estar expuesto sus productos? Si___ No___

12. El proyecto realiza actividades de control de la seguridad en el software? Si____ No___ Si su respuesta es si diga:

a. Las actividades de control de la seguridad las realizan:

_____un equipo interno del proyecto _____cada integrante individualmente ___entre todos

b. ¿Qué actividades del control de la seguridad realizan?_____.

13. ¿El proyecto realiza pruebas de seguridad? Si____ No____ No se___.

14. ¿Utilizan técnicas de aseguramiento de la calidad? Si___ No____ No se___

Glosario de Términos

Antivirus: Los **antivirus** son programas cuya función es detectar y eliminar virus informáticos y otros programas maliciosos *(a veces denominados malware).*

Ataque: Evento, exitoso o no, que atenta sobre el buen funcionamiento del sistema.

Cortafuego: Un **cortafuegos** (o ***firewall*** en inglés), es un elemento de hardware o software utilizado en una red de computadoras para controlar las comunicaciones, permitiéndolas o prohibiéndolas según las políticas de red que haya definido la organización responsable de la red.

Ingeniería del software: La **Ingeniería de software** designa el conjunto de técnicas destinadas a la producción de un programa de computadora, más allá de la sola actividad de programación. Forman parte de esta disciplina las ciencias computacionales y el manejo de proyectos, entre otros campos, propios de la rama más genérica denominada Ingeniería informática.

Seguridad Informática: La seguridad informática consiste en asegurar que los recursos del sistema de información (material informático o programas) de una organización sean utilizados de la manera que se decidió y que el acceso a la información allí contenida así como su modificación sólo sea posible a las personas que se encuentren acreditadas y dentro de los límites de su autorización.

Sistema de detención de intrusos: Un **sistema de detección de intrusos** (o **IDS** de sus siglas en inglés *Intrusion Detection System*) es un programa usado para detectar accesos desautorizados a un computador o a una red.

Software: El software es el conjunto de instrucciones que permite al hardware de la computadora desempeñar trabajo útil.

Stakeholders: Personas u organizaciones que están activamente implicadas en el negocio ya sea porque participan en él o porque sus intereses se ven afectados por los resultados del proyecto. Pueden ser los propietarios, la dirección, los clientes, los trabajadores, los proveedores, etc.

Virus informático: Un **virus informático** es un programa que se copia automáticamente y que tiene por objeto alterar el normal funcionamiento de la computadora, sin el permiso o el conocimiento del usuario. Aunque popularmente se incluye al "malware" dentro de los virus, en el sentido estricto de esta ciencia los virus son programas que se replican y ejecutan por sí mismos. Los virus, habitualmente, reemplazan archivos ejecutables por otros infectados con el código de este. Los virus pueden destruir, de manera intencionada, los datos almacenados en un ordenador, aunque también existen otros más "benignos", que solo se caracterizan por ser molestos.

Vulnerabilidad: Posibilidad de ocurrencia de la materialización de una amenaza sobre un Activo.