



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
(ICAI)
INGENIERO INDUSTRIAL

**CONSOLIDACIÓN Y ESTANDARIZACIÓN DE
LOS PROCEDIMIENTOS DE PRUEBA DE
LAS APLICACIONES DEL SOFTWARE DEL
HOSPITAL DE LA UNIVERSIDAD DE MIAMI**

Autor: Gonzalo García Ballesteros

Directores: Dra. Susana Ortiz Marcos

Dr. Shihab S. Asfour

Madrid

2014

Consolidación y estandarización de los procedimientos de prueba y desarrollo de aplicaciones de software a través de todos los aspectos de Servicios Corporativos Médicos.

Autor: García Ballesteros, Gonzalo

Directores: Marcos Ortiz, Susana.

S. Asfour, Shihab

Entidad colaboradora: ICAI – Universidad Pontificia de Comillas

UM – University of Miami

RESUMEN DEL PROYECTO

Introducción

El objetivo principal de este proyecto se basa en la mejora del proceso de prueba del principal software del hospital de la Universidad de Miami en Miami, FL, E.E.U.U

Con los informes médicos electrónicos o EMR del inglés, Electronic Medical Record, son muchas las ventajas que tanto pacientes como profesionales del cuidado de salud y empresas relacionadas obtienen. No obstante, estos informes electrónicos han de ser gestionados con un programa informático especializado. Desde los años 1970 aproximadamente, estos EMR han revolucionado la industria hospitalaria. Su uso se ha extendido en todo el mundo de manera rápida. Informes del publicador Kalorama prevén que el mercado de EMR o EHR del inglés, Electronic Health Records, llegará al valor de 22.3 mil millones de dólares (se habla de dólares americanos) durante los años 2012 a 2015 con un tasa de crecimiento uniforme del 5.5%. En 2014 el valor del mercado alcanzaba los 20.9 mil millones de dólares. Además, el uso de este sistema en consultas médicas también ha crecido mucho en estos años, crece a un ritmo del 22% mucho mayor que el hospitalario, en el que ya está más consolidado.

Además se prevé que los hospitales seguirán invirtiendo en actualizaciones del sistema y, que eventualmente, se regulará el uso de estos sistemas obligando a las empresas médicas a actualizarlos regularmente.

Por esto, los hospitales invierten cada vez más en asegurar el correcto funcionamiento de estos programas como se produce en el hospital donde se desarrolla este proyecto.

Se introduce y explica brevemente el funcionamiento de estos programas. En primer lugar, el hospital usa el programa Epic, espina dorsal del funcionamiento diario de todas las funciones que se realizan. Este programa permite a los empleados del hospital desempeñar su trabajo. Epic se alza sobre una base de datos llamada MUMPS. Éste es un lenguaje informático especializado ya que está integrado en la memoria. De modo que los hospitales pueden crear sus propias aplicaciones dentro de Epic si lo desean o cambiar las ya existentes ajustándose a sus necesidades. Es crucial por tanto, que al hacer esto el programa no dé lugar a fallos y es aquí donde nace la necesidad de probar el programa.

Al inicio del proyecto, el hospital contaba con un equipo dedicado a esta tarea. Sin embargo, la manera a proceder se había basado en anteriores experiencias y no se había llegado a plasmar el modo de proceder. Esto daba lugar a errores evitables, pérdida de tiempo y baja productividad.

Dado que no había ningún procedimiento estándar, ni tampoco algún responsable absoluto del proceso, la adquisición de información se vio obstaculizada.

Al final, con la ayuda de varios empleados del hospital y trabajadores del UMIT, la oficina de IT, que en parte se dedicaban a este proceso de prueba se pudo descifrar el modo global de proceder.

Es aquí cuando se empezó a identificar los problemas que habían en el proceso. Los principales fallos que se encuentran en dicho proceso de prueba se basan en la realización de las pruebas independientemente y en algunos casos, su realización a mano y en papel. Para realizar dicho proceso, no se debe olvidar que se rige por una serie de regulaciones que el director de calidad del hospital introdujo, como el almacenamiento de los guiones de prueba. Además para la estandarización del proceso se debían escribir unos *SOP*, del inglés Standard Operating Procedure o Procedimiento Estándar de Trabajo también regulados.

El proceso se basa en el testeo del programa a través de cinco pruebas previamente establecidas para detectar si un cambio, mejora o actualización programada en el sistema está correctamente desarrollada. Ha de indicarse, que es imposible probar al programa por completo. La calidad de la prueba y por lo tanto, la confianza en que el programa esté realmente bien configurado y sea ejecutable por los usuarios del mismo en el hospital, reside en la habilidad y experiencia de los probadores y en el tipo, orden y cantidad de estas cinco pruebas.

Si durante cualquiera de estas pruebas se encontraba un fallo en el programa, el proceso era detenido, los guiones se entregaban a otro equipo, la actualización del sistema no se lanzaba y el tiempo empleado por los trabajadores en dicho proceso se malgastaba.

Por lo tanto se buscó la solución intentando suplir el papel y la des-personificación del proceso por algo definido. De este intento surgió la búsqueda de sistemas de ayuda a dicho proceso y en particular, la sugerencia final que propone la incorporación de un programa de ayuda al testeado en el hospital.

Tras la búsqueda de información de estos programas y sus funciones, se investigó cuáles podrían ser los programas más competitivos a adoptar.

Después de una serie de comparativas se eligió el competidor más asequible y con mayores o semejantes funcionalidades. Este producto es el programa *SpiraTest*.

Este programa ofrece una amplia gama de servicios. Fácilmente se pueden organizar diferentes proyectos con múltiples pruebas y permite ver en una sola pantalla el progreso global de los proyectos, el progreso individual de ellos o el estado de las pruebas que los componen.

También mejora la seguridad de almacenamiento de los guiones y por tanto de los datos de los pacientes y la seguridad del proceso en general. Elimina la necesidad de dedicar espacios al almacenamiento reduciendo costes.

Además los encargados de realizar estas pruebas son capaces de trabajar más eficientemente acortando el tiempo que le dedican a esta tarea. La gestión y documentación de errores ayuda en posteriores pruebas aumentando la probabilidad de éxito en su consecución.

Con la adopción del programa, se elaboró un nuevo organigrama de trabajo por el que se rige el proceso estando siempre dominado automáticamente por *SpiraTest*. De esta manera el proceso queda claro y la des-personificación se elimina teniendo control en todo momento de la autoría de cualquier edición, prueba o eliminación de cualquier documento.

Para corroborar la mejora que supone la implantación de este software, se lleva a cabo un análisis de costes y se muestran las ventajas y desventajas.

Los aspectos negativos que debemos tener en cuenta al adaptar esta solución para atajar el problema de la mala gestión del proceso de prueba, son los siguientes:

- Los empleados necesitan ser entrenados para el uso de estos programas.
- Los datos son vulnerables a *hackers* que pueden llegar a interceptar grandes cantidades de información de una forma más rápida y anónima.
- Requiere una pequeña inversión inicial o mensual, según se desee contratar el servicio.

Además en algunos hospitales, si el programa EMR que el hospital usa se basa en un servidor de internet:

- El coste del servidor a largo plazo puede llegar a ser más caro que el uso antiguo de informes en papel.
- Vulnerabilidad a interrupciones en el servicio de internet.
- La información clínica almacenada junto con otras prácticas de pacientes pueden ser mezcladas.
- Si el negocio quiebra, toda la información desaparece rápidamente.

De estas desventajas las más importantes a tener en cuenta son la necesidad de entrenar al personal y la posibilidad de ser atacados por los piratas informáticos o *hackers*. No obstante la seguridad informática en el hospital es muy elevada de manera que la diferencia de dificultad a la hora de robar los informes físicos frente a la dificultad que supone entrar en el sistema informático del hospital, puede favorecer a esta última.

Por otro lado las ventajas más destacables que se obtienen, no se mencionan todas, son:

- Ahorro de tiempo: un 44% en la administración de errores y un 60% en su resolución. En particular, se estima un ahorro del 23.5% en el tiempo empleado por los encargados de realizar los procesos de prueba de un proyecto (20 días).
- Ahorra de dinero: Al menos, un 24.3% en el proceso de prueba. (40.329USD).
- Se aumenta la productividad de la empresa. Además, al usar menos recursos se mejora la eficiencia.
- La seguridad aumenta. Mayor seguridad en el cumplimiento de las regulaciones de las leyes de protección de datos.
- El control de calidad mejora, el tiempo de reacción ante fallos es menor y el modo de atajarlos más efectivo.
- Se tiene un mayor control financiero en esta tarea.
- Se disminuye el impacto ambiental haciendo el hospital más ecológico.

Como se ve las ventajas compensan con creces a las desventajas por lo que se recomienda de igual modo la adquisición del programa.

[SANG08] SOP Workshop: Workshop in a Book on Standard Operating Procedures for Biotechnology, Health Science, and Other Industries

[PATT05] Software Testing

[MADI05] Process Mapping, Process Improvement and Process Management

[BEAV05] The Practical Guide to HIPAA Privacy and Security Compliance

[KANE01] Lessons Learned in Software Testing: A Context-Driven Approach

[BECK02] Test Driven Development

[CART01] Electronic Medical Records: A Guide for Clinicians and Administrators

www.socra.org SoCRA 20th Annual Conference Session Descriptions.

www.brampton.com What is a Standard Operating Procedure (SOP)?

www.hhs.gov SUMMARY OF THE HIPAA PRIVACY RULE

www.emrsimplyput.com Tag Archives: electronic medical records;

www.ahd.com American Hospital Directory

Consolidating and standardization of Application and Software Development testing procedures across all aspects of Medical Corporate Services.**Author: García Ballesteros, Gonzalo**

Directors: Marcos Ortiz, Susana.

S. Asfour, Shihab

Collaborating institution: ICAI – Universidad Pontificia de Comillas

UM – University of Miami

Summary of the Capstone's Project

Introduction

The main objective of this project is to improve the testing process of the main software used in the hospital of the University of Miami in Miami, FL , USA

With EMR or Electronic Medical Records systems, many are the advantages to both patients, health care professionals and related companies obtain. However, these electronic reports must be managed with specialized software. Since around 1970 , these EMR has revolutionized the hospitality industry. Its use has become spread worldwide quickly. Publisher Kalorama forecasts that the EMR or EHR (Electronic Health Records) market, will worth \$22.3 billion during the years 2012 to 2015 with a uniform growth rate of 5.5 %. In 2014 the net worth of this market reached 20.9 billion dollars. Furthermore, during the past few years, the use of this system has been growing relentlessly. It has been growing at a rate of 22%, much higher than the hospital market, which is now more consolidated.

Also, it is forecasted that the hospitals will keep investing money in the upgrades of the system and that eventually, the government is going to regulate the minimum performance of this programs by forcing the businesses to upgrade their applications.

So, it is supposed that hospitals will keep investing money in upgrades, and also in ways to ensure the proper functioning of this EMR system. This is what is happening in the hospital where this Project takes place.

It is introduced and briefly explained how this programs are being operated. First, the hospital uses the Epic program, the backbone of daily operations of all functions performed at the hospital. This program allows hospital employees to perform their jobs. Epic is perched on a database called MUMPS. This is a specialized computer language that has built-in memory. So that hospitals can create their own applications

within Epic or change existing ones adjusting them to their needs. It is crucial therefore that in doing so the program does not lead to failure and this is where the need to test the program was born.

When the project began, the hospital had a team dedicated to this task. However, the way to proceed was based on previous experience and had not come to shape the way forward. This resulted in avoidable mistakes, wasted time and low productivity.

Since there was no standard procedure, nor any responsible for the process, the acquisition of information is hampered.

Finally, with help of several hospital employees and workers of the UMIT and IT office, partly engaged in this testing process, the overall course of action was decoded.

This is when the problems in the process got identified. The main failures that are in the process of testing are based on the performance of tests independently and in some cases, the completion by hand on paper. For this process, do not forget that is governed by a set of rules that the director of hospital quality introduced as storing test scripts. In addition to the standardization of the process had an SOP, Standard Operating Procedure, is needed and also regulated.

The process is based on the testing program through five previously established steps. The goal is to detect whether a change or scheduled update is properly developed. It is to be noted that it is impossible to test the program thoroughly. The quality of the test and therefore the confidence in the reliability of the program lies in the skills and experience of the testers and the type, order and number of these five tests.

If during any of these tests a bug is found in the program, the process is stopped, the scripts are delivered to another team, the system update is not launched and the time spent by employees in this process is wasted.

So the solution tries to eliminate paper and to identify the author of any change in the process. From this attempt, the suggestion of looking to a help program is born.

After finding information on these programs and their functions, we investigated what could be the most competitive programs to adopt.

After a series of comparisons with the most affordable and competitive alternative a program got chosen. This product is the *SpiraTest* program.

This program offers a wide range of services. Easily, you can organize various projects, it allows you to do multiple tests on a single screen where you can see the

overall progress of the projects, the individual progress of them or the state of the tests they contain.

It also improves the safety of scripts storage and therefore data of patients and the overall process safety. It eliminates the need to dedicate space to storage and reduces costs.

Also, the responsible of performing these tests are able to work more efficiently shortening the time they devote to this task. The error management and documentation helps subsequent tests to get a higher probability to succeed.

With the adoption of the program, a new workflow is designed for the process being always automatically dominated by *SpiraTest*. Thus the process is clear and the de-personification is removed at all times having the control of the authorship of any publication, testing or document removal.

To corroborate the improvement achieved by the implementation of this software, a cost analysis is performed and the advantages and disadvantages are shown.

The negative aspects that we must consider by adapting this solution to tackle the problem of poor management of the testing process, are:

- Employees need to be trained to use these programs.
- Data are vulnerable to hackers who can get to trap large amounts of information more quickly and anonymously.
- Requires a small initial or monthly investment, as service option desire.

Also at some hospitals, if the EMR program that the hospital uses is based on an internet server:

- The cost of long-term server can be more expensive than the former use of paper reports.
- Vulnerability to disruptions in internet service.
- The clinical information stored along with other practices of patients can be mixed.
- If the business goes out, all information disappears quickly.

The most important disadvantages to consider are the need to train staff and the possibility of being attacked by hackers. However computer security at the hospital is very high so it is probably easier to be stolen rather than hacked.

On the other hand the most significant advantages obtained, not all listed, are:

- Time saving: 44% in the error handling and 60% in resolution. In particular, a saving of 23.5 % is estimated to be the time taken by those conducting the testing process of a project (20 days).
- Money savings: At least 24.3% in the testing process. (\$40,329).
- The business productivity is increased. In addition, using fewer resources efficiency is improved.
- Security increases. Increased confidence in meeting the regulations of the data protection laws.
- Quality control improves, the reaction time to failure is shorter and the way to tackle them more effective.
- You have more financial control in this task.
- The environmental impact is shortened. The hospital is more eco-friendly.

As seen, the advantages outweigh the disadvantages so likewise the acquisition of the program is recommended.

References

[SANG08] SOP Workshop : Workshop in a Book on Standard Operating Procedures for Biotechnology , Health Science, and Other Industries

[PATT05] Software Testing

[MADI05] Process Mapping , Process Improvement and Process Management

[BEAV05] The Practical Guide to HIPAA Privacy and Security Compliance

[KANE01] Lessons Learned in Software Testing: A Context- Driven Approach

[BECK02] Test Driven Development

[CART01] Electronic Medical Records : A Guide for Clinicians and Administrators

www.socra.org

SOCRA 20th Annual Conference Session Descriptions.

www.brampton.com

What is a Standard Operating Procedure (SOP) ?

www.hhs.gov

SUMMARY OF THE HIPAA PRIVACY RULE

www.emrsimplyput.com

Tag Archives : electronic medical records ;

www.ahd.com

American Hospital Directory



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL



Contenido

Índice de tablas	4
Índice de ilustraciones.....	5
Capítulo 1: Introducción y motivación	7
1.1 Introducción	8
1.2 Antecedentes	10
1.3 Objetivos	12
Capítulo 2: Estado de la cuestión y metodología.....	13
2.1 Estado de la cuestión	14
2.2 Conceptos básicos y teoría de los procesos de prueba de software.	15
Capítulo 3: Desarrollo e implementación de la mejora	23
3.1 Proceso de testeo inicial en el hospital.....	24
3.2 Problemas encontrados en el método de trabajo inicial	33
3.3 Control de calidad	36
3.4 Posibles soluciones.....	37
3.5 Investigación de programas	38
3.6 Producto recomendado: <i>SpiraTest</i> [®]	42
3.7 Matriz de requisitos y características del programa.....	55
3.8 Análisis de costes	56
Capítulo 4: Conclusiones	67
4.1 Mejoras de la IT y beneficios para el hospital.....	68
4.2 Desventajas	71
4.3 Proceso de testeo sugerido con <i>SpiraTest</i> [®]	73
4.4 Conclusión final	76
Capítulo 5: Referencias	77
5.1 Bibliografía	78
5.2 Enlaces web.....	78
Capítulo 6: Apéndices.....	79
6.1 Regulaciones básicas para la utilización de informes electrónicos y firmas electrónicas en la industria.....	80
6.2 Lista de verificación para la regulación HIPPA de protección de datos (parte).	86



Índice de tablas

Tabla 1 - Matriz de requisitos del programa por parte del hospital	55
Tabla 2 - Tiempos estimados de las diferentes pruebas	56
Tabla 3 - Coste inicial del proceso	56
Tabla 4 - Tiempo estimado de las diferentes pruebas con el nuevo software	57
Tabla 5 - Coste del proceso con el nuevo software	58
Tabla 6 - Distribución de Centros de aplicaciones y distancias.....	62
Tabla 7 - Impacto ambiental anual reducido por el hospital	65
Tabla 8 - Impacto ambiental anual aproximado en E.E.U.U. si se realizara un correcto proceso de testeo	66
Tabla 9 - Mejoras en IT y beneficios para el hospital.....	69



Índice de ilustraciones

Ilustración 1 - Organigrama de trabajo del proceso inicial	25
Ilustración 2 - Ejemplar de un guión de la Prueba de Aplicación.....	27
Ilustración 3 - Ejemplar de un guión de la Prueba de Integración	28
Ilustración 4 - Ejemplar de guión de prueba de Final de Usuario	29
Ilustración 5 - Ejemplar de un guion de Prueba de Integración fallido.....	31
Ilustración 6 - Ejemplar de guion de Prueba de Integración fallido 2.....	32
Ilustración 7 - Captura de pantalla de el registro de errores con el metodo inicial	33
Ilustración 8 - Gráfico de errores por aplicación.....	35
Ilustración 9 - Comparativa de las tres alternativas.....	39
Ilustración 10 - Gráfico comparativo de alternativas.....	41
Ilustración 11 - Principales funciones de <i>SpiraTest</i>	42
Ilustración 12 - Características principales del software	43
Ilustración 13 - Ejemplo de la función de Muestra de Características.....	44
Ilustración 14 - Ejemplo de la función de Gestión de Requerimientos.....	45
Ilustración 15 - Ejemplo de la función de Manual de Casos de Prueba	46
Ilustración 16 - Ejemplo de la función de Pruebas Automáticas.....	47
Ilustración 17 - Ejemplo de la función de Compartir casos de Prueba entre diferentes actualizaciones	48
Ilustración 18 - Ejemplo de la función de Seguimiento de errores y/o defectos.....	49
Ilustración 19 - Ejemplo de la función de Rastreo de Incidentes.....	50
Ilustración 20 - Ejemplo de la función de Gestión de documentos	50
Ilustración 21 - Ejemplo de la función de Seguimiento de las versiones de documentos	51
Ilustración 22 - Ejemplo de la función de Soporte para interfaz móvil.....	52
Ilustración 23 - Ejemplo de la función de Integración de email.....	53
Ilustración 24 - Ejemplo de la función de Registro del historial de cambios	54
Ilustración 25 - Gráfico informativo de la reducción de tiempo	58
Ilustración 26 - Gráfico informativo de la reducción de coste	58
Ilustración 27 - Gráfico de comparación de costes de tiempo y dinero	59
Ilustración 28 - Mapa de la disposición de los edificios del hospital y las oficinas	60
Ilustración 29 - Mapa de la disposición de los edificios del hospital y las oficinas (completo) ...	61
Ilustración 30 - Croquis de la planta del edificio principal del UMIT	64
Ilustración 31 - Organigrama de trabajo con la adaptación de <i>SpiraTest</i> ®	73
Ilustración 32 - Desglose del bucle de solución	74



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

Capítulo 1: Introducción y motivación



1.1 Introducción

Hoy en día los hospitales en Estados Unidos son negocios altamente beneficiosos llegando, en el mejor lugar, a cantidades que se acercan según el Directorio Americano de Hospitales ó *American Hospital Directory*, de uso público, a 10 billones de dólares americanos (billones americanos) al año y con capacidad para tratar a 1600 pacientes. Es por tanto que se destinen grandes cantidades de recursos y estudios a mejorar estos centros.

Este proyecto consta de la mejora que ha tenido lugar en el Departamento de Tecnología de la Información, IT, del hospital de la Universidad de Miami en Downtown Miami, Florida.

La Universidad de Miami compró el *Cedars Medical Center* en 2007 y lo transformó en lo que ahora es su hospital principal. Con 560 camas, fue el primer hospital multidisciplinar y de cuidados intensivos poseído por una universidad en la región.

Actualmente, estudiantes de la Escuela *Miller* de Medicina de la Universidad de Miami trabajan en el en colaboración con otros doctores cuidando y tratando a pacientes, haciendo proyectos de investigación y teniendo a disposición las últimas tecnologías.

Tomando como inspiración la capacidad de liderazgo de la actual presidenta de la universidad, Donna Shalala, el hospital se ha esforzado en la seguridad de sus pacientes y es uno de los hospitales más seguros de la nación.

Estudiantes de todas las facultades de la Universidad de Miami, entre otros, realizan sus proyectos y trabajos de final de Carrera en este hospital mejorando semestre tras semestre la calidad del mismo con bajos costes.

En el caso de este proyecto final de carrera, se trabaja en la consolidación y estandarización del proceso de prueba de todos los aspectos de los Servicios Corporativos Médicos utilizados para el desempeño de las actividades en el Hospital, dentro del entorno del programa *Epic*.



- ¿Qué es *Epic*?

Epic junto con *Cerner* y *MEDITECH* es uno de los mayores programas de software EMR (*Electronic Medical Records* o Informes Médicos Electrónicos). Estos programas eliminan la necesidad de utilizar fichas de pacientes en formato de papel. *Epic* es de propiedad privada y ofrece un paquete integrado de software para las empresas de cuidado de la salud basado en una base de datos *MUMPS*.

MUMPS (*Massachusetts General Hospital Utility Multi-Programming System*, más tarde: 'Multi-User Multi-Programming System') o de forma alternativa, *M*, es un lenguaje de programación informático de múltiples usos. Su característica fundamental es su característica de tener la base de datos integrada permitiendo un acceso de alto nivel al almacenamiento en disco usando sencillos programas simbólicos variables y vectores o cadenas de texto similares a las variables usadas por los lenguajes informáticos más comunes para acceder a la memoria.

Este programa, *Epic*, apareció en el año 1979 (diez años después que *MEDITECH*) y revolucionó el sistema de funcionamiento de los hospitales ya que son muchas las ventajas que un informe médico electrónico ofrece. Algunos ejemplos son:

- Los informes son completamente legibles.
- Es más accesible (incluso para los pacientes) y permite hacerlo desde casa.
- Permite acoplar imágenes, documentos digitales o pruebas especiales (rayos X) fácilmente a la ficha del paciente.
- Mejor relación entre los centros clínicos y los departamentos financieros.
- Aumenta la seguridad del paciente a la hora de tener en cuenta características especiales del mismo (alergias, intervenciones pasadas etc...).
- Se mejora el seguimiento de la información obtenida en los laboratorios y su almacenamiento.
- Costes varios disminuidos; espacio en oficinas, informes, empleados, transporte de documentos.
- Mejor solución ante desastres, copias de seguridad.



- Es más fácil realizar un seguimiento, por ejemplo en forma gráfica, de niveles determinados del paciente en el tiempo como la presión arterial.
- Todos los empleados de sanidad que usen este sistema verán la misma información mejorando la calidad de tratamiento del paciente.

1.2 Antecedentes

El proyecto se realizó en el departamento de IT de la universidad de Miami en el Hospital ó *UMIT* (University of Miami Institute of Technology). Es aquí donde se produce el mayor uso de *Epic*. Las aplicaciones de este software ayudan al funcionamiento, especialmente , de grandes hospitales. Sus aplicaciones permiten ayudar a los trabajadores con un gran número de tareas; registrar nuevos pacientes, programar sistemas clínicos, facturar a las aseguradoras, planear los utensilios necesarios en cualquier intervención quirúrgica etc...

Por tanto, es de vital importancia para el hospital que este software funcione en todas las áreas y servicios del edificio y proporcione los resultados esperados en todo momento. Para esto, con las nuevas exigencias que surgen con el paso del tiempo, el programa debe ser actualizado regularmente y es aquí donde el trabajo del *UMIT* toma su papel. Su trabajo conlleva, en la mayoría de los casos, la realización de la actualización del programa y su prueba antes de ser lanzado y utilizado por los usuarios del hospital. En este proceso de prueba es en donde se centra este proyecto.

En el proceso de prueba de *Epic*, hay 5 etapas por las que el equipo debe pasar antes de poder lanzar la actualización del sistema. Estos pasos son 5 pruebas que el equipo ha estimado necesarias y suficientes. Estas pruebas reciben el nombre de: Prueba unitaria o de unidad, Pruebas de las aplicaciones, Pruebas de integración, Pruebas de usuario final y Pruebas de regresión. En cada etapa o paso hay una serie de procesos que el analista debe pasar antes de poder pasar a la siguiente etapa. Además, en cada una de estas etapas cabe la posibilidad de encontrar defectos que impidan el paso a la siguiente etapa. Los defectos encontrados en el sistema tienen que pasar por un proceso diferente para su corrección con el fin de ser solucionado y poder pasar a la siguiente etapa del proceso. El tiempo que se tarda en cada etapa de la prueba varía de un día a 4 semanas, dependiendo de la cantidad de problemas que se encuentren.



UNIVERSIDAD PONTIFICIA COMILLAS

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

INGENIERO INDUSTRIAL

Actualmente, no existe un método estándar y todos los pasos del proceso de pruebas se llevan a cabo por los empleados del equipo. Estos empleados fueron los primeros en realizar las pruebas correspondientes a la primera actualización del sistema, y aunque sigan las recomendaciones básicas del manual de *Epic*, a grandes rasgos el proceso entero se lleva a cabo de memoria. Esto crea problemas debido a que algunos pasos del proceso pueden ser olvidados quedándose fuera, y se propicia la creación de defectos en el sistema. Mediante la implementación de un flujo de trabajo, se puede realizar un seguimiento del proceso e identificar qué etapa del proceso de pruebas crea la mayor cantidad de defectos. También se quiere encontrar la etapa en la que el hospital emplea más recursos ya sean estos económicos o físicos (espacio en oficinas, tiempo) y solucionar los problemas derivados para reducir el coste y mejorar la calidad.



1.3 Objetivos

El objetivo de este proyecto es entender el funcionamiento del proceso de prueba, detectar los errores y solucionarlos ofreciendo un método de trabajo. Para ello se creará una metodología sólida y robusta para dicho proceso de prueba del sistema que se utilizará en todos los distintos departamentos y servicios médicos de la empresa. Se incluirá la creación de un organigrama de trabajo y procedimiento operativo estándar que se utilizará para el proceso de prueba que se le aplica a las mejoras y actualizaciones del sistema.

Los objetivos en particular deseados por el Hospital incluyen:

1. Proceso de Pruebas
 - a. Equipo de Aplicación
 - b. Súper Usuarios
 - c. Usuario final
2. Documentación
 - a. Guiones para las pruebas.
 - b. Colección de problemas.
 - c. Correcciones Formato.
3. Incorporación de las nuevas normas venidas a la Organización por el director de Control de Calidad.
4. Aplicación e inclusión de las normas de los estándares *HIPPA*.
5. Almacenamiento de la documentación.
6. Incorporación de las normas de la FDA.
7. Metodología de la Comunicación.



Capítulo 2: Estado de la cuestión y metodología



2.1 Estado de la cuestión

Para llevar a cabo el proyecto, lo primero es identificar cómo se está realizando el proceso de prueba hasta el momento. Para ello es necesario reunirse con los integrantes del equipo de *UMIT* e identificar quién es el encargado de cada parte del proceso. Al no haber un sistema definido se presentan dificultades a la hora de lograr esto. Después de varias reuniones, se descubre que el proceso empieza con la labor de uno o dos empleados que normalmente se encargan de la **primera etapa** del proceso, la prueba unitaria. Se insiste en que al no haber un proceso definido, no se tiene la certeza de que siempre se realice de la misma manera. En esta primera etapa el encargado dispone de un esquema en papel para hacer la prueba donde se **indica detalladamente el módulo del software que se está probando y cada componente de él, el paso en el que se encuentra, el resultado esperado, el resultado real, las iniciales del empleado y un espacio para comentarios**. A mano, el encargado de realizar este trabajo rellena uno a uno todos los pasos de cada aplicación necesaria dentro del programa de cada paso en el proceso de prueba global (los 5).

De este modo si la prueba es realizada con éxito quien se encargue de esta primera etapa le entrega el documento escrito al siguiente compañero y éste realiza la siguiente prueba de la siguiente etapa y se procede de la misma manera hasta que se han completado las 5 pruebas. Sin embargo, en algunas ocasiones el mismo empleado puede llegar a hacer más de una prueba o incluso un mismo guion de prueba puede volver a alguna persona que se encargó de etapas anteriores perdiendo totalmente la trazabilidad del proceso global.

En el momento en el que en alguna de las etapas se detecta un error, el proceso es detenido y se entregan los escritos a un equipo distinto y ellos se ocupan de arreglarlo. Cuanto más tarde se encuentra el defecto más tiempo es malgastado por los encargados de las pruebas y por lo tanto más dinero pierde el hospital.

De la misma manera, se programaron reuniones con el director de control de calidad y con el equipo responsable de cumplir con las normativas *HIPPA* y de la *FDA*. Para la solución se debe tener en cuenta estos requerimientos.



2.2 Conceptos básicos y teoría de los procesos de prueba de software.

- ¿Qué es una prueba de software?

Se trata de ejecutar el software a probar en un entorno simulado o real usando *inputs* (o entradas) seleccionadas específicamente de la manera que se desee.

- Objetivos de las pruebas.

El objetivo principal de las pruebas es el de asegurar el correcto funcionamiento del software y que se aumente la confianza al trabajar con él. Al probarlo, de encontrarse fallos en el sistema se tratarán para intentar corregirlos.

También es interesante para conocer las propiedades del software, como pueden ser:

- Confiabilidad
- Rendimiento
- Uso de memoria
- Seguridad
- Usabilidad

La seguridad del programa es una característica muy a tener en cuenta cuando se trata del software de un hospital.

- Dificultades a la hora de probar un software

Una de las dificultades principales a la hora de probar un software es determinar qué tipo de prueba hay que asignar en cada prueba al software. No en todas las pruebas hay que usar el mismo tipo de prueba o tampoco es necesario probar de la manera más completa al programa.



Otros ejemplo típicos:

- Determinar si el resultado esperado es correcto o no.

Normalmente, se aceptan frecuentemente los resultados porque son plausibles más que correctos. Es difícil determinar qué resultados son correctos por varios motivos:

- En primer lugar el software ha sido escrito artificialmente para computar una respuesta.
 - La cantidad de resultados posibles es tan grande que es imposible validarlos todos por completo.
 - No hay un resultado visible.
- Comparar los resultados que se dan internamente en el programa con los que se esperan.

En la mayoría de los casos, para esto se hará uso de la consola de depuración de errores. Nótese el gran número de líneas de código que un programa de este tipo tiene (del orden de decenas de millón).

- Contrastar el tipo de prueba que se ha realizado y si es la correcta.

Se corroborará con la etapa del proceso a la que se empareja y sus necesidades.

· Determinar lo que se puede afirmar acerca del programa cuando las pruebas han finalizado.

Al no saber con certeza qué porcentaje del programa se ha probado o si al probarlo algunas partes dan errores que permiten proseguir con el desarrollo normal del programa por no ser catastróficos, pueden pasar desapercibidos.

- Medir el rendimiento del programa.
- Comparar las diferentes estrategias seguidas.



Dado que en la mayoría de los casos siempre hay más de un modo de probar el software, la experiencia de la persona que realiza la prueba es determinante a la hora de utilizar la mejor estrategia. Se puede intuir si una prueba más corta puede hacer la misma función que otra prueba análoga de mayor duración.

- Comparar el tiempo requerido con el tiempo estimado.
- Identificar el porcentaje de prueba realizado al programa.
- Tipos y etapas en el proceso de prueba.
- Pruebas a pequeña escala

Pruebas unitarias o 'Unit testing'

Pruebas de integración o 'Integration testing'

- Pruebas a gran escala

Pruebas del sistema

Pruebas de principio a fin

- Pruebas unitarias o 'Unit testing'

En estas pruebas se prueban las líneas de código individuales ejecutables más pequeñas. Normalmente estas pruebas son llevadas a cabo por los programadores. Los casos de prueba son normalmente basados en las líneas de código, las especificaciones y como se indicaba anteriormente, según la intuición y habilidad o experiencia del responsable.

Las herramientas de las que puede disponer son;

Un guion de prueba

Un indicador del rango que dichas líneas de código abarcan.

Un generador automático de casos de prueba.



- Pruebas de integración o 'Integration testing'

En este caso, lo que se comprueba son las interacciones entre dos o más componentes unitarios como se explicaba en el apartado anterior. Estas pruebas también son llevadas a cabo por los programadores normalmente. Pone especial atención en las interfaces.

Algunos de los problemas derivados de este tipo de prueba:

- Orden en el que las unidades están combinadas. ¿ Qué implica este orden?.

Si es de partes superiores a partes inferiores en el conjunto del código, los niveles superiores se probarán repetidamente y se necesitará un *stub*. Un *stub* se refiere a un trozo de código usado como sustituto de alguna otra funcionalidad para simular el comportamiento de algún código existente o simular el comportamiento de otro código aún no desarrollado. Si al probar un código necesitas que el programa te dé una respuesta de uno de los módulos que se encuentra en los niveles inferiores pero éstos no están disponibles todavía, los *stubs* sirven de módulos de respuesta temporales.

Si es de partes inferiores a partes superiores se testearán repetidamente los niveles inferiores y se necesitará un *driver*. Un *driver* actúa del mismo modo que un *stub* pero se necesita cuando la parte del código que se prueba es la inferior y se necesitan los módulos que se encuentran en la parte superior.

Si se testean las partes críticas primero sin orden aparente se necesitarán ambos *drivers* y *stubs* y estas unidades críticas se probarán repetidamente.

- Modo de asegurar la compatibilidad y/o corrección de los componentes del programa que son suministrados externamente (por el usuario).

Los problemas que se derivan de este tipo de testeo son más peligrosos cuando no se lleva a cabo una adecuada prueba unitaria, el planeamiento y organización del testeo de integración es inadecuado o se usa documentación o suministros externos incorrectos. (El usuario que introduce el *input* requerido no es válido).



- Pruebas de sistema

En este tipo de pruebas se testea la funcionalidad del sistema entero. Este tipo de pruebas son desarrolladas normalmente por probadores o 'testers' profesionales.

Realidades en las pruebas de sistema:

- Aunque el programa se testee enteramente, nunca todos los problemas serán encontrados. Como se dice anteriormente, puede haber fallos que no perjudiquen a la consecución del programa y por lo tanto no sean identificados. Los recursos para los testeos son limitados. Las especificaciones normalmente son ambiguas o no están claras además de ser cambiantes. Los programas suelen ser siempre demasiado largos como para permitir que los casos de prueba sean seleccionados basados en las líneas de código que interesan.

Otras realidades:

- Las pruebas exhaustivas no son posibles y están relacionadas con la imaginación del *tester* y son difíciles.
- Uno de los mayores objetivos del testeo es la prevención de errores y fallos catastróficos.
- Las pruebas han de ser siempre planeadas.
- Los encargados de las pruebas han de ser independientes de los desarrolladores del programa, de este modo se encontrarán más fallos y se evitarán cruces de intereses.

- Pruebas aleatorias

En este tipo de pruebas, la selección de qué probar está basada en una distribución de probabilidad. Típicamente se usa la distribución uniforme donde los casos de prueba son elegidos con igual probabilidad, o la distribución operacional que se basa en el registro histórico de los datos de pruebas pasadas.



UNIVERSIDAD PONTIFICIA COMILLAS

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

INGENIERO INDUSTRIAL

La parte positiva de este tipo de pruebas es que se pueden generar automáticamente permitiendo probar muchos más casos que en modo manual siendo más económico o más productivo.

La parte negativa es que el código deberá estar bien estructurado y si no se usa una distribución operacional, el testeado no se aproximará al comportamiento del usuario típico y por lo tanto esos errores no serán depurados con una adecuada efectividad.

- Pruebas de regresión

En este caso se aprovechan las versiones del programa modificadas anteriormente y se testean solamente los nuevos cambios introducidos en la última actualización. De este modo se pretende asegurar que dicha actualización no ha introducido errores nuevos en el programa. Normalmente esto se lleva a cabo por los probadores o *testers*.

El principal problema en este caso es cómo elegir una prueba efectiva desde una versión probada anteriormente.

- Pruebas de aplicación

Para este tipo de pruebas, dentro del programa en su globalidad, se desea probar las aplicaciones que hay dentro de él. Por ejemplo en el caso de *Epic*, al hacer esta prueba se podría probar por ejemplo el módulo de Ambulatorio, el de Facturación a aseguradora, concertación de una cita etc...

En este caso se suelen usar plantillas en donde paso a paso se sigue el proceso que se ha de realizar.

- Pruebas del usuario final

Este es el caso más sencillo. En este tipo de pruebas, se le presenta la actualización a uno de los empleados que van a usar el programa. Se le pide que desempeñe su trabajo con



normalidad y registre el comportamiento del programa para su aplicación o aplicaciones de uso cotidiano.

En general, se pueden encontrar muchos más tipos de testeo (pruebas de rendimiento, confianza, carga, estrés...) Para este proyecto con conocer estos tipos es suficiente para entender la consecución de la memoria. Sin embargo se explica a continuación cómo elegir el orden de las pruebas para entender el orden elegido para *Epic*.

- Estrategias para la selección de pruebas

Todas las estrategias sistemáticas para la realización de las pruebas han de verse como una manera de hacer divisiones del trabajo. Esta división del trabajo puede basarse en las características del código, las especificaciones de los detalles, la estructura del código, el análisis de riesgo que se desee, la calidad esperada del programa y su seguridad etc...

Según la división que se quiera hacer aparecen las desventajas de cada una. Si las divisiones se quieren hacer según las características del código, en este caso sólo se puede probar con el testeo unitario o 'Unit testing' y el precio que esto conlleva suele ser excesivamente alto. Si se basa en las especificaciones se encuentra que raramente se dan unas especificaciones estrictas y formales y es difícil asegurar que todas ellas han sido tenidas en cuenta a la hora de seleccionar los tipos de pruebas. Estas además hay que ir actualizándolas con el paso del tiempo y las actualizaciones del sistema. Tampoco se puede asegurar que el programa esté libre de fallos aunque se hayan probado todas las funcionalidades del mismo.

- Priorización de los casos de prueba

Una vez que el entorno de prueba se ha definido se ha de elegir el orden de las pruebas según un criterio. De modo, que aunque no todas las pruebas se pueden realizar por falta de tiempo o personal, al menos, los testeos más importantes se aseguran.



Bases para la priorización del tipo de pruebas:

Entradas más frecuentemente usadas.

Funciones críticas del programa.

Entradas individuales más frecuentes.

Cobertura de brechas o fallos.

Cobertura o aseguramiento de funciones básicas deseadas.

Fallos potenciales.

- Conclusiones de la prueba de software

¿Por qué automatizar el proceso de prueba?

La mayor ventaja de la automatización de las pruebas se encuentra en el ahorro de tiempo de los empleados y por lo tanto ese tiempo se puede dedicar bien a otras tareas o se puede prescindir de algunos trabajadores por algún tiempo. El testeo es repetitivo por lo tanto si es llevado a cabo por personas puede dar lugar a mayor cantidad de fallos ya que se pierde interés. Los guiones finales son valiosos, de ellos se puede sacar información que los desarrolladores pueden necesitar y ahorrarán tiempo a posteriori.

¿Por qué ha de realizarse un buen testeo del software?

En primer lugar es fundamental el correcto funcionamiento del software en cualquier empresa, si bien es crítica esta necesidad cuando se trata de un hospital.

La reducción de los errores también y el coste suele ser significativa. Por ejemplo, en América el informe NIST publicado en 2002 indicaba que el coste total para la economía de Estados Unidos del inadecuado proceso de testeo de software suponía al año 59 mil millones de dólares frente a los 22 mil millones que potencialmente llegaría a costar el mismo testeo de software al realizar las mejoras más probables. El guardado electrónico ofrece muchas ventajas frente a robos, pérdidas, localización y difusión.

Por todo esto el proyecto se basa en proporcionar el método sólido y robusto que el hospital demanda basándose en un software de calidad que mejore el servicio en el hospital, reduzca costes, disminuya errores y ahorre tiempo y/o dinero.



Capítulo 3: Desarrollo e implementación de la mejora



3.1 Proceso de testeo inicial en el hospital.

Al principio, el proceso de trabajo se basaba en unos guiones plantilla predefinidos que podían ser electrónicos o en formato papel. Éste es el caso de las pruebas de Integración y de Usuario Final.

Cuando se terminaban de realizar las pruebas, las versiones electrónicas eran impresas para, junto con las versiones ya en papel entregarlas a los siguientes compañeros que se encargaban de la siguiente prueba.

Al terminar todas las pruebas los documentos eran almacenados físicamente en varios cuartos dentro del hospital.

Si durante las pruebas se producía algún error, estos documentos eran llevados al coordinador o en su defecto al director de control de calidad y éstos se encargaban de revisar el proceso. Si todo parecía estar bien estos documentos eran transportados al centro de ayuda como se explicará más adelante.

El organigrama de trabajo inicial se presenta en la siguiente hoja para facilitar la comprensión, en rojo se marcan las principales fuentes de error y pérdida de calidad y eficiencia.



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

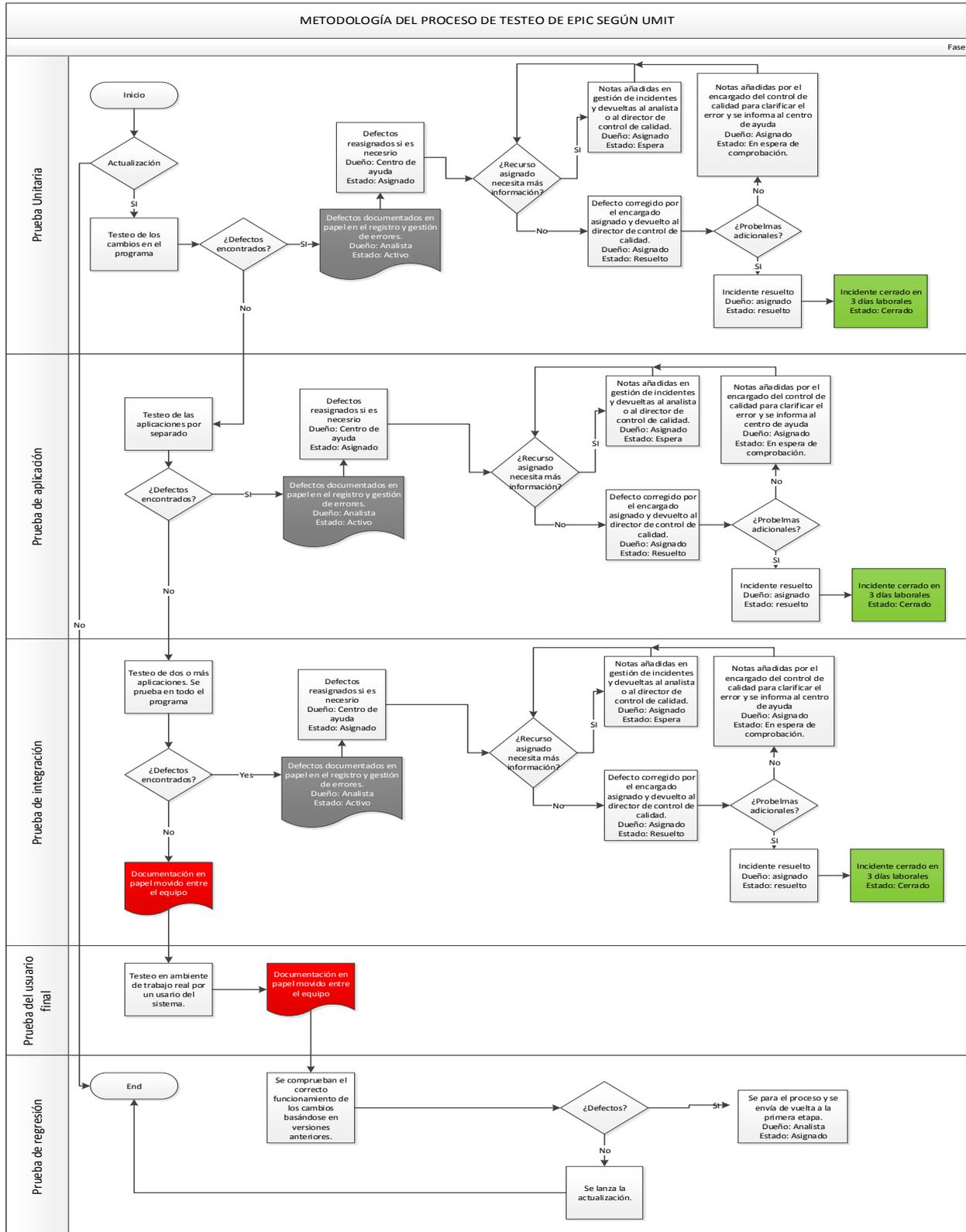


Ilustración 1 - Organigrama de trabajo del proceso inicial



UNIVERSIDAD PONTIFICIA COMILLAS

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

En este organigrama se aprecia el modo de trabajo hasta el momento llevado a cabo por el equipo de testeo del hospital. Se basa como se explicaba anteriormente en 5 etapas; Prueba unitaria o Unit testing, Prueba de aplicación o Application testing, Prueba de integración o Integrated testing, Prueba de Usuario final y Prueba de regresión.

El orden elegido por el equipo fue establecido por el coordinador teniendo en cuenta un proceso lógico. Con el testeo unitario se comprueba que las principales líneas de código ejecutables individualmente no den fallo. De este modo se podrá continuar con pruebas mayores asegurándonos una base sólida y sin errores. Terminada esta fase, la más costosa, se pasa a la prueba de aplicación. En esta etapa se prueba cada aplicación del programa *Epic* con el objetivo de encontrar las aplicaciones que más fallos dan y asegurar que cada aplicación individualmente funciona bien y poder seguir a la siguiente etapa donde se comprueba la interacción entre dos o más de estas aplicaciones, la prueba de integración. Una vez terminadas estas tres etapas se puede proceder a entregar el prototipo de la actualización de *Epic* a un usuario real del sistema. Aquí comprobaremos si aparece cualquier problema en el uso cotidiano del software. Si todo esto está correcto el proceso terminará después de pasar con éxito la última prueba, la prueba de regresión. En ésta se prueba *Epic* por completo basándose en actualizaciones y pruebas anteriores.

Primera etapa, Prueba unitaria o Unit testing:

El flujo de trabajo comienza en la prueba unitaria. En esta etapa el primer encargado testea los principales códigos de *Epic*. Como se muestra en el organigrama esto es realizado a mano, escribiendo sobre una plantilla en el ordenador e indicando el resultado obtenido comprobando si es el esperado o no. De ser así se pasa a la siguiente etapa.



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

Segunda etapa, Prueba de aplicación o Application testing:

Del mismo modo en esta etapa se desarrolla el testeo de las aplicaciones explicado anteriormente. Se anota en una plantilla toda la información que se puede sacar de esta prueba y se comprueban también los resultados obtenidos comparándolos con los esperados.

Overall Status:		Workflow	<Script Title>						
Untested									
Status Graph:		Description							
		Test Patient							

#	Application	User	Step	Variations	Status	Stops Testing?	Issue #	Issue Comments	Additional Info
1	Application Name		Summary instruction for a testing task.		Untested				
2	Application Name	USER LOGIN	First step for this task.		Untested				
3	Application Name	USER LOGIN	Second step.		Untested				
4	Application Name	USER LOGIN	Third step. Delete the rows for any steps you don't need.		Untested				
5	Application Name	USER LOGIN	Fourth step. If you need more steps for a task, insert rows above the last step to maintain proper grouping.		Untested				
6	Application Name		Summary instruction.		Untested				
7	Application Name	USER LOGIN	Step.		Untested				
8	Application Name	USER LOGIN	Step.		Untested				
9	Application Name	USER LOGIN	Step.		Untested				
10	Application Name	USER LOGIN	Step.		Untested				
11	Application Name		Summary instruction.		Untested				
12	Application Name	USER LOGIN	Step.		Untested				

[Introduction](#) |
 [Test Script Tracker](#) |
 [Script Template](#) |
 [Abstraction](#) |
 [Clinical Support Visit](#) |
 [Creating Addenda](#) |
 [Immunizations](#) |
 [Record Request](#) |
 [Lab Orders](#)

Ilustración 2 - Ejemplar de un guión de la Prueba de Aplicación. Fuente: documentación interna del UM Hospital



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

Tercera etapa, Prueba de integración:

Repetitivamente, en esta etapa se lleva a cabo el mismo trabajo, diferente testeado pero de igual modo se usa una tercera plantilla esta vez a mano y en papel y se anota todo de nuevo. Se comprobarán los resultados obtenidos comparándolos nuevamente con los esperados. Si todo es correcto se procede al penúltimo paso.

Workflow: Outpatient PCP Office Visit w/ Lab orders and Imaging (Practice Based)								
Section A: Overview				Section D: Applications / Initials		Section E: Tester Name/ Initials		
Testing Cycle:	Integrated Testing	ADT	Reporting Workbench					
Testing Date:	3/10/2013	Ambulatory	Security/SER					
Test Script ID:	CSV.UCHART.001	ASAP	Welcome					
Version:	4	Benefits Engine	Willow					
		Cadence						
		Clarity						
		EpicCare Link						
		Haiku Canto						
		HB	Legacy					
		HB Claims						
		HIM						
		Identity	Third Party					
		Interfaces						
		MyChart						
		PB	Lab Systems					
		PB Claims						
		Prelude						
		Referrals						
		Note - Please remember to document Order ID #s						
		Note - Please remember to document DOS						
Section F: Application / Workflow Steps								
Step ID	Team	User Name	Workflow Step	Testing Step / Process	Expected Outcome	Actual Outcome	Tester Initials/Date	Comments or Issue #
ADT								
1	Registration	UMESPROCIIV (PRO)	Register Patient	Add patient's demographic information.	Demographic screen opens and you're able to enter all info.			
2	Registration	UMESPROCIIV (PRO)		Add Patient contact and PCP information.	Patient contact screen opens and you're able to enter all info.			
3	Registration	UMESPROCIIV (PRO)		Add guarantor information	Guarantor info screen appears and you're able to enter info.			
4	Registration	UMESPROCIIV (PRO)		Add coverage and click finish	Coverage Wizard appears.			
SCHEDULING								
5	Scheduling	UMESPROCIIV	Schedule appointment	Schedule New patient visit	In Schedule scanner portion. available			

Ilustración 3 - Ejemplar de un guión de la Prueba de Integración. Fuente: documentación interna del UM Hospital



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

Cuarta etapa, Prueba del usuario final:

En esta etapa se introduce la novedad de que el probador o *tester* será un empleado real del hospital que no está o tiene que estar relacionado con el UMIT y se comprueba si puede llevar a cabo su trabajo. Dicho responsable deberá anotar a mano la misma información en una cuarta plantilla diferente y tendrá que ser devuelta al equipo para su análisis.

PATIENT NAME:	Enduser, Patient One	TYPE OF PATIENT:	New Patient Visit
DEPT:	PAC Neurology	INSURANCE:	Medicare
HAR #:		DOB:	10/12/1963
MRN:		SEX:	Female
CSN:		DOS/ADM DT:	
SCRIPT ID:	EUTC001	VERSION:	2.0
<i>PRELUDE</i>			
<i>STEP</i>	<i>WORKFLOW</i>	<i>EXPECTED RESULTS</i>	<i>ACTUAL RESULTS</i>
<i>Workflow: Patient Registration and Appointment Scheduling</i>			
1.	Select REG/ADT>Registration	Patient look up screen appears.	
2.	Enter patient's name, dob, ssn and sex and select Find patient.	If patient does not exist, the no patient found error appears.	
3.	Select Demographic link on interactive face sheet and fill in patient information. Make sure to include an email address. Select Patient Contact link and enter contact information.	All information is entered correctly.	
4.	Select add guarantor and create guarantor account. Select Add Coverage and enter UM AETNA HMO for insurance.	Guarantor and coverage appears.	
<i>CADENCE</i>			
<i>STEP</i>	<i>WORKFLOW</i>	<i>EXPECTED RESULTS</i>	<i>ACTUAL RESULTS</i>
<i>Workflow: Schedule new patient visit in PAC Neurology</i>			
1.	Use the appropriate search options.	Available appointments appear.	
2.	Click schedule.	Provider Schedule appears and/or first available.	
3.	Select Appt Time on Schedule.	Appt Review screen Appears.	
4.	Add PCP and referring provider.	PCP and Referring provider populates.	
5.	Verify patient and guarantor using checklist.	Hyperlink changes to green check mark and status change to Verified.	
<i>INSURANCE VERIFICATION</i>			
<i>STEP</i>	<i>WORKFLOW</i>	<i>EXPECTED RESULTS</i>	<i>ACTUAL RESULTS</i>
<i>Workflow: Insurance Verification</i>			
1.	Locate and open patient on work queue.	Patient appears on work queue.	

Ilustración 4 - Ejemplar de guión de prueba de Final de Usuario. Fuente: documentación interna del UM Hospital



Quinta y última etapa, Prueba de regresión:

Si hasta esta etapa todo ha funcionado correctamente, se procede a este último test.

Aquí se prueba *Epic* por completo ayudándose de pruebas anteriores. Este es el paso más importante ya que si todo funciona como es esperado la actualización del sistema puede ser lanzada y utilizada por el hospital mejorando su rendimiento. Si no, se tiene que volver al principio del proceso e intentar identificar el fallo perdiendo mucho tiempo y volviendo a utilizar empleados en la misma tarea. Es por esto que es importante que los probadores sean lo más experimentados y hábiles posibles para reducir los fallos.

Fallos encontrados:

Si en cualquier etapa se encuentra un fallo o el resultado no es el esperado, se para la prueba por completo y se documenta. El encargado de la prueba le entrega el guion al equipo de UMIT (si no es uno de ellos) para que éstos revisen si la prueba se ha llevado a cabo correctamente. En caso de ser así se le asigna al centro de ayuda donde los desarrolladores de la actualización ya sean ajenos o no deberán hacerse cargo del problema y resolverlo.

Se puede ver un ejemplar de un guion de prueba fallido a continuación. Se aprecia la cantidad de tiempo que lleva a uno de estos probadores realizar una de estas pruebas. En particular este guion tiene 10 páginas con 152 pasos. También se puede ver en la imagen siguiente en qué paso se identifica el fallo (número 67).



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

UOHART OCT2013_SUGAD - INTEGRATED TESTING

64	Ambulatory	UMAMBGEND		Create a progress note in Note Writer	The Note Writer opens in the Sidebar and you are able to navigate to various activities with the note pinned to the Sidebar.	As expected	9:24 OK	
65	Ambulatory	UMAMBGEND		Manually enter a level of service in the Level of service field. Level of service entered: 99213.	The level of service is added.	As expected	9:24 OK	
66	Ambulatory	UMAMBGEND		Close Encounter	You are able to close the encounter.	As expected	9:24 OK	
Willow								
67	Willow	Pharmacist-UMRXPHARM	Orders are written on paper with specific protocols. Meds from Pyxis are pulled by Nurse. Unit has access to both SCT2 and SCT Pyxis units. Pharmacy receives orders via paper and or Physician order entry with combination orders to be dispensed from Pharmacy and pulled from Pyxis.	Enter orders for: -1. Furosemide 20mg in), select and stay -2. Ibuprofen 400mg tab). Make sure no labels print, accept	order composer with dispense location UMHC BMT Pyxis - Furosemide Verify no labels print Ibuprofen: dispense location main Rx	Failed unable to access patient via today's	9/24/13 MOR	Script note-Clarify dispense areas for meds Patient, Patient Station or Manage Orders
68	Willow	Pharmacist-UMRXPHARM	Orders are written on paper with specific protocols. Meds will be dispensed from Main Pharmacy receives orders via paper and or Physician order entry.	Enter orders for: 1. D5 1/2 kcl 20meq/ 1L (41001) select and stay, 2. Neupogen 300mcg (10036) Sign orders	Both orders appear in order composer; Neupogen with dispense location: SCCC Main Pharmacy. D5 1/2 kcl 20meq/mag (mixed in Pharmacy). Labels print for both medications.			Review medications for 2010 change
69	Willow	Nurse: UMAMBGERN	Pull furosemide from Pyxis unit (SCT2 OR SCT)	Pull furosemide	Nurse can access and pull med from Pyxis			
Guidance								
70	Check Out	UMESADTFRO	Check-out patient	From DAB select Check Out	Copy screen appears			
HIM Coding								
71	HIM	HIMCODER/UMHC CODING	Code the account.	From the Account tab in the workqueue list, highlight and double-click the "UMHC HIM - Uncoded #P-#P" workqueue.	The account workqueue activity opens.			
72	HIM	HIMCODER/UMHC CODING		Double-click the account to be coded.	The Coding Info activity opens. A window appears, asking if you want to start coding the account.			
73	HIM	HIMCODER/UMHC CODING		Click Yes.	Admission, discharge, and provider information populates on the ADT Info tab. At the bottom of the Hospital Coding Info form, the Coding status changes to In Progress and the coder is marked as the login user. The coding productivity timer begins.			
74	HIM	HIMCODER/UMHC CODING		Validate and update (if necessary) the admission, discharge, and provider information on the ADT Info tab.	The account information updates appropriately.			

Ilustración 6 - Ejemplar de guion de Prueba de Integración fallido 2. Fuente: documentación interna del UM Hospital

Además al terminar estos documentos, muestren o no fallos, son llevados a una sala donde se almacenan a requerimiento de las normativas *HIPPA 'Health Insurance Portability and Accountability Act'* o en español Ley de Responsabilidad de Portabilidad del Seguro de Salud donde se establecen las normas de uso y divulgación de la información de la salud de los pacientes. Estos ocupan lugar y suponen un problema para el hospital ya que los datos que contienen estos papeles pueden tener carácter confidencial.



3.2 Problemas encontrados en el método de trabajo inicial

Con este proceso de prueba son varios los problemas que se pueden mejorar.

El principal es el tiempo empleado en cada etapa del proceso. Al realizarlo a mano se pierde mucho tiempo escribiendo y cambiando de un medio a otro (papel-PC). Implica también que un probador solo pueda llevar un test al mismo tiempo y al terminar tiene que entregárselo a la siguiente persona y desplazarse hasta encontrarla ya que no se encuentran en la misma oficina. Adicionalmente, el registro de errores era introducido manualmente mediante Excel por una persona extendiendo más y más el tiempo que esta tarea conlleva. A continuación se expone una parte de un ejemplar de uno de estos registros donde se aprecia el tiempo que se se pierde en esto.

Issue ID	Issue Title	Priority	Phase	Application	Assigned To
1	Missing Background Users	1-Critical-Showstopper	Upgrade - Installation / Validation	Security and Access	Hasimi, Dorian
2	Upgrade stopped on step Unpack Globals	1-Critical-Showstopper	Upgrade - Installation / Validation	System Pulse	Hasimi, Dorian
3	UWeb 2012 - Client-Not_Authorized	2-High-Important, not critical	Upgrade - Installation / Validation	Systems	Lages, Jose
4	Inline validation encounter department length limitations	1-Critical-Showstopper	Essential Build and Test	EpicCare Ambulatory	Gonzalez, Maria Lucia
5	Need to change default of Notes in OB navigators	3-Medium-Need to have, not critical	Optimization-Build and Test	EpicCare Ambulatory	White, Jennifer
6	Error in OB/CVN navigator	2-High-Important, not critical	Essential Build and Test	EpicCare Ambulatory	Gonzalez, Maria Lucia
7	IPP 1170 - Close Encounter Print Group Wrapper Has a Bug	3-Medium-Need to have, not critical	Testing-Application/Unit	EpicCare Ambulatory	Dumas, Chantal A.
8	RN 232379 - Visual Change to Pharmacist Interventions	4-Low-Will need in future	Essential Build and Test	Willow Inpatient	DeLaVoe, Saviour
9	RN 267976 - Dispense Multiple Package Sizes to Fill a Mixture Order with an Order Level Dose 2	3-Medium-Need to have, not critical	Essential Build and Test	Willow Inpatient	Johnson, Rutledge, Martha
10	Pharmacist Access - Today's Pils, no Open Chart Button	1-Critical-Showstopper	Essential Build and Test	Willow Inpatient	DeLaVoe, Saviour
11	AVS not printing more than one allergy	2-High-Important, not critical	Essential Build and Test	EpicCare Ambulatory	White, Jennifer
12	Outlook/Outlook - Not working as expected	2-High-Important, not critical	Essential Build and Test	EpicCare Ambulatory	Cuellor, Zack
13	Unable to undo a system of change	3-Medium-Need to have, not critical	Essential Build and Test	Willow Inpatient	DeLaVoe, Saviour
14	Non-Agent does not restart properly after theAlert Upgrade	3-Medium-Need to have, not critical	Upgrade - Installation / Validation	System Pulse	Hasimi, Dorian
15	Activity Menus does not display properly for admin users in RELBLD	1-Critical-Showstopper	Essential Build and Test	Willow Inpatient	DeLaVoe, Saviour
16	REL Upgrade with RA1155 failed two checks	1-Critical-Showstopper	Upgrade - Installation / Validation	System Pulse	Hasimi, Dorian
17	RA1155 Checksum errors in RELBLD	1-Critical-Showstopper	Upgrade - Installation / Validation	System Pulse	Hasimi, Dorian
18	RA1155 BU Installation Stopped in RELBLD	1-Critical-Showstopper	Upgrade - Installation / Validation	System Pulse	Hasimi, Dorian
19	Notification message not appearing in the Undo Event Window.	3-Medium-Need to have, not critical	Essential Build and Test	ADT	Poletti, Angela
20	error after REL Upgrade	3-Medium-Need to have, not critical	Upgrade - Installation / Validation	System Pulse	Hasimi, Dorian
21	Missing SU in REL - after upgrade	1-Critical-Showstopper	Upgrade - Installation / Validation	System Pulse	Hasimi, Dorian
22	RN 235703 [RTF] - replace IPP 27001 with new IPP 154566	2-High-Important, not critical	Essential Build and Test	Preclude	Rivers, Paulina
23	creating encounters thru interface errors on non-clinical type	1-Critical-Showstopper	Testing-Application/Unit	Bridges	Rerecich, Marilyn
24	Scanned documents stored in EMC won't be available in RELBLD.	2-High-Important, not critical	Testing-Application/Unit	Bridges	Rerecich, Marilyn
25	RN 254175 custom encounter - missing with contact number	2-High-Important, not critical	Testing-Application/Unit	Identity	Rivers, Paulina
26	Physic Orders entered by Pharmacist dropping charge without pull	1-Critical-Showstopper	Essential Build and Test	Willow Inpatient	DeLaVoe, Saviour
27	History navigator sections not displaying the correct items	3-Medium-Need to have, not critical	Testing-Application/Unit	EpicCare Ambulatory	White, Jennifer

Shedlock Ticket #	Date Created	Environment	Status	Opened By	Closed By	Date Closed	Item Type	Path
1297528	12/3/2012 14:00	REL	Closed	Hasimi, Dorian		12/3/2012 0:00	Item	sites/EPIC/Lists/UCHart 2012 Upgrade
1297528	12/3/2012 14:00	REL	Closed	Hasimi, Dorian		12/3/2012 14:55	Item	sites/EPIC/Lists/UCHart 2012 Upgrade
1309511	12/14/2012 15:00	REL	Closed			1/25/2013 0:00	Item	sites/EPIC/Lists/UCHart 2012 Upgrade
1317434	12/20/2012 17:00	RELBLD	Closed			1/15/2013 18:00	Item	sites/EPIC/Lists/UCHart 2012 Upgrade
1318383	12/20/2012 15:00	RELBLD	Closed			3/19/2013 15:00	Item	sites/EPIC/Lists/UCHart 2012 Upgrade
1318686	12/24/2012 14:00	REL	Closed			1/15/2013 14:00	Item	sites/EPIC/Lists/UCHart 2012 Upgrade
1323725	1/3/2013 17:00	RELBLD	Pending Vendor Development	Dumas, Chantal A.	Dumas, Chantal A.		Item	sites/EPIC/Lists/UCHart 2012 Upgrade
1331154	1/10/2013 12:00	RELBLD	Closed			2/7/2013 0:00	Item	sites/EPIC/Lists/UCHart 2012 Upgrade
1332171	1/11/2013 18:00	RELBLD	Closed				Item	sites/EPIC/Lists/UCHart 2012 Upgrade
1331079	1/1/2013 14:00	RELBLD	Closed				Item	sites/EPIC/Lists/UCHart 2012 Upgrade
	1/29/2013 11:00	RELBLD	Closed		White, Jennifer	3/19/2013 14:00	Item	sites/EPIC/Lists/UCHart 2012 Upgrade
1355448	2/1/2013 12:00	RELBLD	Closed		Cuellor, Zack	3/20/2013 18:00	Item	sites/EPIC/Lists/UCHart 2012 Upgrade
1355791	2/1/2013 16:00	RELBLD	Closed		DeLaVoe, Saviour	2/12/2013 0:00	Item	sites/EPIC/Lists/UCHart 2012 Upgrade
1360710	2/5/2013 10:00	TEST	Closed		Hasimi, Dorian	2/7/2013 0:00	Item	sites/EPIC/Lists/UCHart 2012 Upgrade
1369816	2/7/2013 16:00	RELBLD	Closed				Item	sites/EPIC/Lists/UCHart 2012 Upgrade
1297382	2/12/2013 12:00	REL	Closed		Hasimi, Dorian	2/12/2013 14:30	Item	sites/EPIC/Lists/UCHart 2012 Upgrade
	2/13/2013 12:00	RELBLD	Closed		Hasimi, Dorian	2/14/2013 14:00	Item	sites/EPIC/Lists/UCHart 2012 Upgrade
	2/14/2013 14:00	RELBLD	Closed		Hasimi, Dorian	2/14/2013 14:00	Item	sites/EPIC/Lists/UCHart 2012 Upgrade
1370399	2/14/2013 15:00	RELBLD	Closed		Poletti, Angela	3/21/2013 0:00	Item	sites/EPIC/Lists/UCHart 2012 Upgrade
1297482	2/14/2013 17:00	REL	Closed		Hasimi, Dorian	2/14/2012 0:00	Item	sites/EPIC/Lists/UCHart 2012 Upgrade
1297582	2/14/2013 18:00	REL	Closed		Hasimi, Dorian	2/18/2013 10:00	Item	sites/EPIC/Lists/UCHart 2012 Upgrade
1376711	2/21/2013 18:00	RELBLD	Closed				Item	sites/EPIC/Lists/UCHart 2012 Upgrade
1376794	2/23/2013 12:00	RELBLD	Closed		Rerecich, Marilyn	3/1/2013 18:00	Item	sites/EPIC/Lists/UCHart 2012 Upgrade
1386292	2/23/2013 12:00	RELBLD	Closed			3/20/2013 0:00	Item	sites/EPIC/Lists/UCHart 2012 Upgrade
1379487	3/1/2013 15:00	REL	Closed				Item	sites/EPIC/Lists/UCHart 2012 Upgrade
1297588	3/7/2013 6:00	RELBLD	Closed		White, Jennifer	3/11/2013 11:15	Item	sites/EPIC/Lists/UCHart 2012 Upgrade

Ilustración 7 - Captura de pantalla de el registro de errores con el metodo inicial. Fuente: documentación interna del UM Hospital



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

En la versión real de esta captura, el empleado debía introducir o, en el mejor de los casos seleccionar, desde la prioridad del error hasta el momento en el que se efectuó el registro manualmente pasando por la indicación de la etapa en la que se encuentra, la aplicación si corresponde, el asignado a la prueba, una dirección para el seguimiento, el estado etc...

Esto incrementa más y más el trabajo manual cuando podría ser archivado automáticamente de una manera fácil y rápida.

Como se introducía antes, este proceso es repetitivo y por lo tanto aburrido para el trabajador lo que introduce errores humanos que perjudican al desarrollo idóneo de la quinta etapa donde se pretende lanzar la actualización. Este proceso, aun estando bajo el control de un supervisor, es arriesgado para el hospital, esta información ha de ser salvaguardada y en este proceso se despersonaliza el trabajo ya que cualquier trabajador puede cambiar cualquier registro del compañero que le pasó el guion anteriormente sin que el supervisor se dé cuenta necesariamente.

Los guiones de prueba al ser físicos y transmitidos mano a mano pueden perderse fácilmente o ser sustraídos.

El impacto ambiental es también mayor utilizando mayor cantidad de papel prueba tras prueba.



A continuación se muestra un gráfico donde se muestra la etapa de prueba en la que se producen más errores. Los datos se han obtenido de la tabla anterior contabilizando el número de errores que se encontraban en cada etapa significativa del proceso (phase).

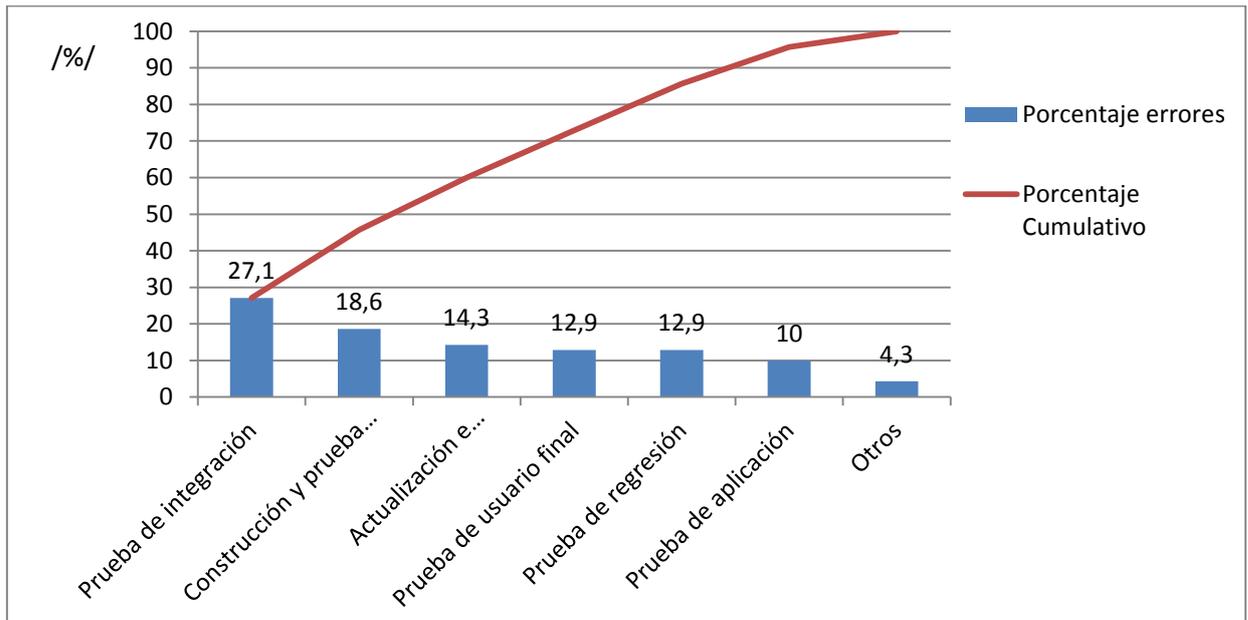


Ilustración 8 - Gráfico de errores por aplicación

Se puede apreciar como la etapa donde se da la mayor cantidad de errores es la es en la prueba de integración. Como se explicaba antes, esta prueba es realizada sobre una plantilla en papel y escrita a mano como se muestra en una de las imágenes.

Otro problema que plantea este sistema es el almacenamiento de los documentos. El director de control de calidad requería una solución tajante para esta brecha ya que en una ocasión un huracán atacó la ciudad de Miami y el hospital y los edificios de oficinas del mismo se vieron afectados. Muchos de estos documentos se perdieron y se vieron ante la posibilidad del cierre del hospital por incumplimiento de la normativa *HIPPA* y el tratamiento de datos personales. Afortunadamente el problema se solucionó y se mejoró el almacenamiento de los documentos pero seguía siendo un almacenamiento físico de los documentos.



De este modo también es difícil calcular cuántos casos de prueba se han llevado a cabo y con qué profundidad, si han sido validados o no.

Si un defecto es encontrado, se dificulta la directa relación con el sistema en la medida de cómo queda afectado por este fallo.

Falta un identificador normalizado para cada una de las pruebas que se realizan.

Los empleados además deberán estar en la oficina para desarrollar estas pruebas en todos los casos y si algún empleado faltara el proceso tendría que esperar hasta su reincorporación o sustitución.

En resumen el hospital tiene unos sobrecostes debido a los salarios por el exceso de tiempo que los empleados le dedican a esta tarea, al almacenamiento y seguridad de los documentos físicos, a la productividad en general de la empresa y al riesgo que se derivan directamente de este proceso de prueba.

3.3 Control de calidad

Un adecuado control de calidad es importante para el proceso de testeo. El control de calidad garantiza que el proyecto se dirige por los caminos adecuados llegando a su finalización habiendo tenido en cuenta todas las especificaciones, funciones y normas requeridas y mitigando errores a la vez que haciendo un análisis del riesgo. Para esto se ha de supervisar el desarrollo del proceso y corregirlo cuando sea necesario.

Para darle solución al problema, se deberá tener en cuenta las siguientes condiciones:

- Se ha de poder llevar un registro de las pruebas
- Debe indicarse quién y cuándo ha editado cada prueba
- Aparecerá las ediciones que cada trabajador haya hecho siendo o no permanentes.
- Registro de tiempo
- Creación de una jerarquía en la que el supervisor o empleado situado en una posición superior pueda restringir el acceso a los demás trabajadores si fuese necesario.



- Almacenamiento seguro.

- Depuración de errores.

Además de estas condiciones las siguientes son deseables:

- Disminución significativa del tiempo

- Guardado automático de los cambios y guiones

- Eliminación del trabajo manual de los empleados

- Posibilidad de soporte portátil

3.4 Posibles soluciones

Al haber estudiado todos los requerimientos, problemas y solicitudes del hospital en cuanto a control de calidad, reglas y control de riesgo se empiezan a presentar una serie de soluciones.

En primer lugar se ha de eliminar el papel en el proceso de testeo del hospital. Esto es costoso, arriesgado y lento.

En segundo lugar se deberá crear un organigrama estructurado en dónde cada empleado vea lo que debe hacer en cada testeo y a qué manos va.

También tiene que ser privado y controlado, si es posible, fuera de las oficinas por si fuera necesario.

Por todo esto se ofrece la adopción de un programa de software que reúna todas o la mayor cantidad de soluciones posibles. Para ello se realiza un estudio de investigación para mostrar los programas informáticos más factibles para abordar las necesidades.



3.5 Investigación de programas

Con la ayuda de internet y algunas revistas de divulgación científica como la *Software Testing Magazine* se encuentran muchos programas de similares características que podrían ayudar a resolver la mayoría de los problemas. Este tipo de programas está empezando a ser cada vez más utilizado y por lo tanto la lista crece rápidamente con el tiempo por lo tanto conviene no comprar la suscripción definitiva y esperar a que la oferta se mantenga y se opte por el mejor software teniendo en cuenta que para cada uno se necesita cierto grado de entrenamiento para los empleados. No obstante, todos son parecidos por lo que habituado a uno de ellos facilitará el rápido aprendizaje del programa definitivo.

Después de investigar y descartar los peores o más simples se presenta una lista con los candidatos más favorables:

- *PassMark*
- *ApTest*
- *QaTrac*
- *TestRail*
- *Zephyr*
- *Testuff*
- *TechExcel*
- *IBM Rational*
- *SpiraTest*
- *TestLodge*
- *HP(Mercury)*
- *MicroFocus*

A continuación se hace un análisis de los mejores posicionados en relación calidad-precio basándose en funcionalidades, precios y opiniones de los expertos. Los tres programas que salen más beneficiados son *SpiraTest*, *TestRail* y *TestLodge*. Los tres de lengua inglesa.



UNIVERSIDAD PONTIFICIA COMILLAS

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

INGENIERO INDUSTRIAL

Import/export formats of the test-artifacts			
	SpiraTest	TestRail	TestLodge
Import	MS Excel; MS Word; MS Project	XML	Google spreadsheet
Export	MS Excel; MS Word; MS Project	XML; CSV; Excel	XML; HTML
	SpiraTest	TestRail	TestLodge
General points			
Trial mode	30 days	30 days	30 days
Licensing	based on the number of users	based on the number of users	based on the number of activities
Requirement management			
CRUD operations	+	only through external RM tool	+
Complexity of adding a requirement	too many fields	only through external RM tool	simple
Structure of requirements	tree-based structure	depends on external RM tool	row-based structure
Traceability matrix	+	+	+
Integration with external RM tools	+	+	+
Test design			
CRUD operations	+	+	+
Complexity of creating a single test case	too many steps	simple method	simple method
Search functionality	global search	limited search	not implemented
Custom fields	+	+	+
Import/export	+	+	+
Documents attachment to the test case	+	+	limits are set in compliance to the selected plan
Test execution			
The execution interface	tree-based structure	tree-based structure	row-based structure
Manual execution	in a browser window	in a pop-up window	in a browser window
Test results tracking	+	+	+
Defect management			
Inbuilt (own) defect management system	+	not implemented	not implemented
Integration with issue trackers	+	+	+

Ilustración 9 - Comparativa de las tres alternativas. Fuente: <http://sysgears.com/articles/tools-battle-spiratest-testrail-and-testlodge/>



UNIVERSIDAD PONTIFICIA COMILLAS

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

En conclusión, podemos apreciar que los tres programas son programas muy completos y organizados de gestión de testeo de software y cubren la mayoría de las necesidades de los usuarios. En particular *TestLodge* y *TestRail* ofrecen una completa y conveniente interfaz de usuario y solucionan la mayoría de las necesidades que proyectos de pequeño a mediano tamaño pueden necesitar. Aunque *SpiraTest* es el más complejo, se posiciona como el más completo y ofrece un amplio abanico de soluciones para las necesidades de los clientes. Es el apropiado para los proyectos de mayor tamaño.

A continuación se muestran los precios para los distintos softwares teniendo en cuenta la dimensión del equipo de UMIT y su posible expansión. (Actualmente 10 personas y con expansión estimada a corto-medio plazo de menos de 15 personas).

Precio mensual		
SpiraTest	TestLodge	TestRail
\$180	\$120	\$500

Salta a la vista el precio elevado de *TestRail* frente a sus otros dos competidores. Por lo tanto sabiendo que ofrece muy similares características y funcionalidades que los otros dos programas y a falta de la opinión del UMIT se descartaría en primer lugar.

Los otros dos programas *SpiraTest* y *TestLodge* tienen un precio muy similar y competitivo por lo que deberá decidirse según las características.



Para facilitar la evaluación de características y asequibilidad de los programas en conjunto se presenta un gráfico relacionando la asequibilidad y capacidad de los programas.

Panorama Competitivo

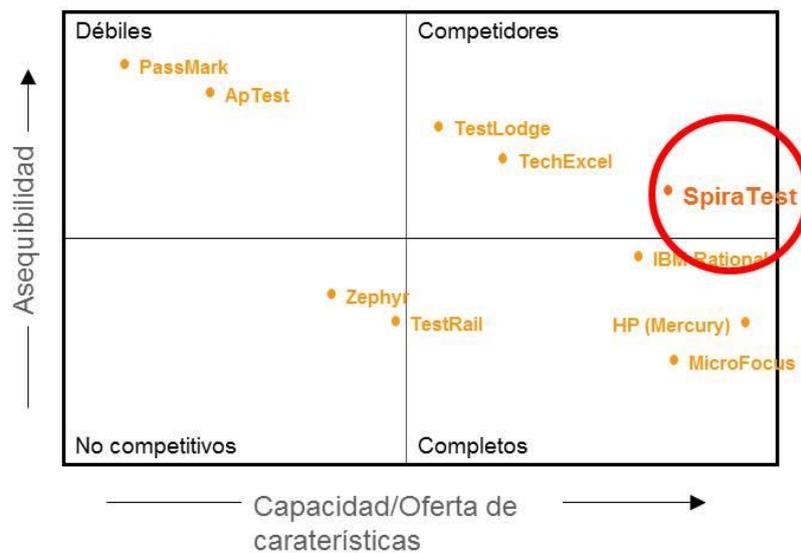


Ilustración 10 - Gráfico comparativo de alternativas. Fuente: <http://www.inflextra.com/SpiraTest/>

Por lo expuesto se sugiere elegir el software *SpiraTest*.

Normalmente para el control de calidad las empresas utilizan las siguientes herramientas para administrar sus procesos de prueba:

- Requisitos almacenados en documentos MS-Word o en hojas de cálculo MS-Excel
- Guiones de prueba para la aceptación de usuarios almacenadas en MS-Word o hojas de cálculo MS-Excel con resultados escritos a mano o páginas de internet independientes que siguen defectos en el sistema.



· El testeo aleatorio se emplea con una variedad de herramientas que dependerán del lenguaje informático que se usa.

Esto lleva a los siguientes retos:

· Los procesos están hechos con un solo objetivo y no son repetibles en diferentes proyectos.

· Es difícil saber cuándo has llegado realmente al punto final del proceso de testeo.

· Se pierde tiempo y es difícil medir el progreso y la productividad durante el proceso de prueba.

3.6 Producto recomendado: *SpiraTest*®

Visión de conjunto del producto *SpiraTest*.

Con este producto el hospital no tiene que comprar por separado los programas para rastrear los defectos, las herramientas para el testeo o los requerimientos de captura de información. Este producto proporciona prácticamente todos los requisitos que el control de calidad requiere en un solo paquete.

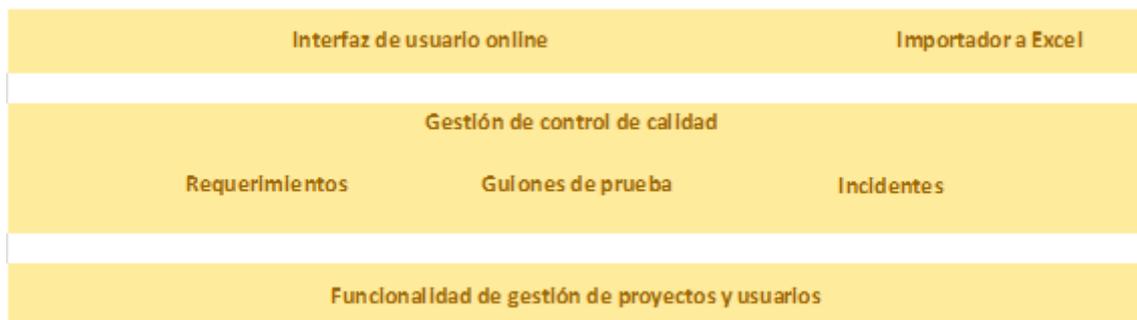


Ilustración 11 - Principales funciones de *SpiraTest*. Fuente: <http://www.inflectra.com/SpiraTest/>



SpiraTest gestiona los requerimientos del proyecto, las pruebas, los defectos encontrados en un único programa con posibilidad de llevar el registro total de cualquier edición.

Características del producto sugerido SpiraTest	Gestión de los requerimientos y características de las pruebas
	Creación, edición y ejecución de casos de prueba
	Seguimiento de las mejoras, los defectos, el riesgo y los problemas.
	Potente y sencillo interfaz del usuario
	Apoyo para guiones de pruebas predefinidos
	Enlaza los defectos a los diferentes pasos o etapas mientras se ejecuta la prueba
	Borradores e informes personalizables
	Gráficos e informes online
	Organigramas y valores de campo personalizables

Ilustración 12 - Características principales del software



- Muestra de las características del programa

SpiraTest puede organizarse en diferentes grupos para diferentes proyectos que pertenezcan a una organización común, cliente o una división. De este modo el software te permite ver rápidamente en una pantalla la suma del estado de todos los grupos al igual que comparar el estado de diferentes proyectos en particular.

En esta imagen se puede observar la cantidad de pruebas que se están llevando a cabo en dos proyectos diferentes así como el estatus de todas ellas. Esto permite rápidamente saber la salud de los proyectos individuales o del sistema en conjunto.

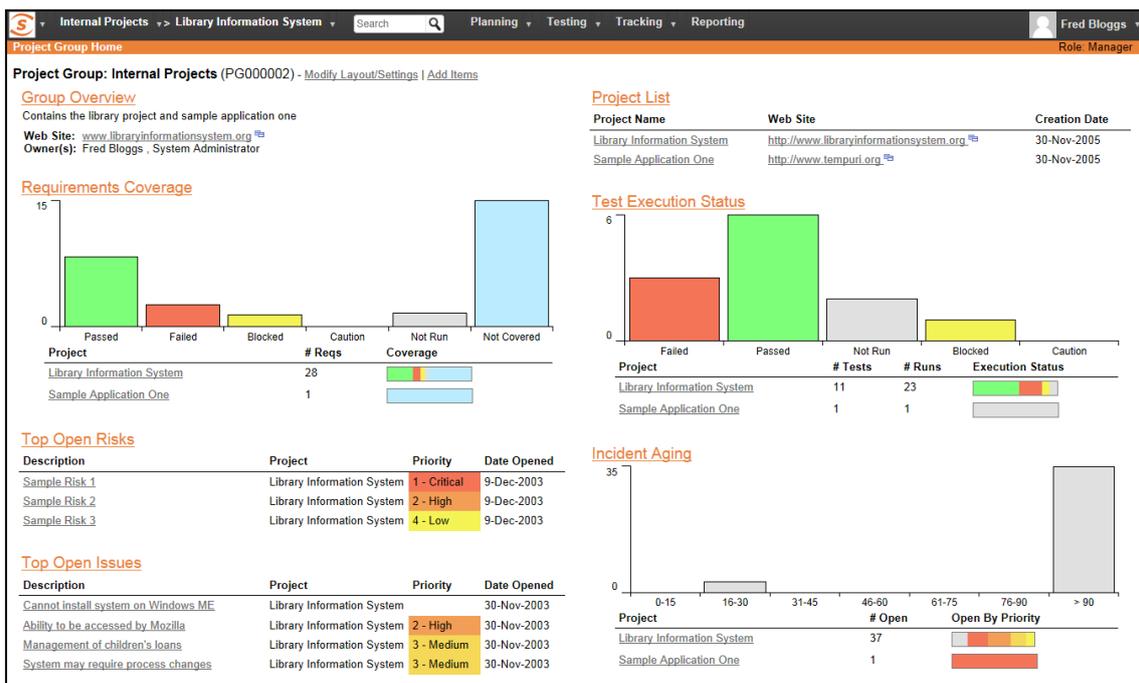


Ilustración 13 - Ejemplo de la función de Muestra de Características. Fuente: <http://www.inflectra.com/SpiraTest/>



- Gestión de requerimientos

Este software permite crear editar o gestionar el objetivo de un proyecto así como sus requerimientos en una organización piramidal. De esta manera puedes crear una matriz con tus objetivos asignándoles una prioridad o importancia.

Name	Test Coverage	Importance	Status	Author	Release	ID	Difficulty	Edit
Functional System Requirements	<div style="width: 100%; height: 10px; background-color: green;"></div>		In Progress	Fred Bloggs		RQ000001		<input type="button" value="Edit"/>
Online Library Management System	<div style="width: 100%; height: 10px; background-color: green;"></div>		In Progress	Fred Bloggs		RQ000002		<input type="button" value="Edit"/>
Book Management	<div style="width: 100%; height: 10px; background-color: green;"></div>	1 - Critical	In Progress	Fred Bloggs		RQ000003		<input type="button" value="Edit"/>
Ability to search by ISBN	<div style="width: 0%; height: 10px; background-color: green;"></div>	1 - Critical	Planned	Fred Bloggs	1.1.0.0.0001	RQ000029		<input type="button" value="Edit"/>
Ability to add new books to the system	<div style="width: 100%; height: 10px; background-color: green;"></div>	1 - Critical	Completed	Fred Bloggs	1.0.0.0.0001	RQ000004	Moderate	<input type="button" value="Edit"/>
Ability to edit existing books in the system	<div style="width: 100%; height: 10px; background-color: green;"></div>	1 - Critical	Completed	Fred Bloggs	1.0.0.0.0001	RQ000005		<input type="button" value="Edit"/>
Ability to delete existing books in the system	<div style="width: 100%; height: 10px; background-color: green;"></div>	1 - Critical	Completed	Fred Bloggs	1.0.0.0.0002	RQ000006		<input type="button" value="Edit"/>
Ability to associate books with different subjects	<div style="width: 100%; height: 10px; background-color: green;"></div>	1 - Critical	Completed	Fred Bloggs	1.1.0.0.0001	RQ000007		<input type="button" value="Edit"/>
Ability to associate books with different authors	<div style="width: 100%; height: 10px; background-color: green;"></div>	1 - Critical	Completed	Fred Bloggs	1.1.0.0.0001	RQ000008		<input type="button" value="Edit"/>
Ability to associate books with different editions	<div style="width: 100%; height: 10px; background-color: green;"></div>	1 - Critical	Completed	Fred Bloggs	1.1.0.0.0002	RQ000009		<input type="button" value="Edit"/>
Ability to completely erase all books stored in th...	<div style="width: 100%; height: 10px; background-color: green;"></div>	1 - Critical	Completed	Fred Bloggs	1.2.0.0	RQ000010		<input type="button" value="Edit"/>
Edition Management	<div style="width: 0%; height: 10px; background-color: green;"></div>	1 - Critical	In Progress	Fred Bloggs		RQ000011		<input type="button" value="Edit"/>
Ability to create different editions	<div style="width: 0%; height: 10px; background-color: green;"></div>	1 - Critical	In Progress	Fred Bloggs	1.0.0.0.0003	RQ000012		<input type="button" value="Edit"/>
Author Management	<div style="width: 100%; height: 10px; background-color: green;"></div>	2 - High	In Progress	Joe P Smith		RQ000013		<input type="button" value="Edit"/>
Ability to add new authors to the system	<div style="width: 100%; height: 10px; background-color: green;"></div>	2 - High	Planned	Joe P Smith	1.0.0.0	RQ000014		<input type="button" value="Edit"/>

Ilustración 14 - Ejemplo de la función de Gestión de Requerimientos. Fuente: <http://www.inflectra.com/SpiraTest/>

- Gestión de casos de prueba

De forma análoga *SpiraTest* permite también crear, editar y organizar los guiones de prueba en una estructura piramidal flexible de modo que la priorización de los pasos en el proceso de prueba queda a la elección del hospital. Se pueden mover, copiar o filtrar los casos de prueba según el probador o *tester* quiera.



- Manual de casos de prueba

Cada prueba consiste en una serie de pasos que representan las acciones individuales que el probador deberá llevar a cabo para completar el test. De este modo el probador tiene un organigrama predefinido por el sistema y no queda a elección o experiencia del mismo como ocurría al principio. También se pueden añadir otros guiones de prueba definidos por el usuario para conducir al probador por el camino deseado.

The screenshot shows a web application interface for test case management. The main content area displays a test case titled "Ability to create new book [TC:000002]". The test case name is "Ability to create new book". Below the name, there are tabs for "Overview", "Req. Coverage", "Test Runs", "Releases", "Incidents", "Attachments", and "History". The "Details" section is expanded, showing a description: "Tests that the user can create a new book in the system" and a precondition: "condition #1". Below the description, the "Test Steps" section is expanded, showing a table of test steps.

Step #	Test Step Description	Expected Result	Sample Data	Execution Status	ID	Edit
Step 1	Call 'Login to Application' (TC17)			N/A	TS000001	Edit
Step 2	User clicks link to create book	User taken to first screen in wizard		Failed	TS000002	Edit
Step 3	User enters books name and author, then clicks Next	User taken to next screen in wizard	Macbeth, William Shakespeare	Not Run	TS000003	Edit
Step 4	User chooses book's genre and sub-genre from list	User sees screen displaying all entered information	Play, Tragedy	Not Run	TS000004	Edit
Step 5	User clicks submit button	Confirmation screen is displayed		Not Run	TS000005	Edit

Ilustración 15 - Ejemplo de la función de Manual de Casos de Prueba. Fuente: <http://www.inflectra.com/SpiraTest/>



- Ejecución de los manuales de los casos de prueba

SpiraTest permite ejecutar grupos de casos de prueba de manera que el probador puede ver y seguir instrucciones y determinar si el sistema está comportándose de la manera esperada. A diferencia del modo anterior, donde el probador escribía manualmente los resultados que veía.

- Pruebas automáticas

Como indicábamos antes, estas pruebas son más económicas y permiten realizar mayor número de ellas en el mismo tiempo. Con este programa puedes seleccionar testeos automáticos a algunos casos de prueba.

Automation

This section defines the automated test script associated with this test case:

Automation Engine: Rapise

Script Type*: Attached Linked Repository

Filename: EditExistingBook\EditExistingBook.sstest

Document Type: Functional Specification

Document Folder: Root Folder

Version: v 1.0

Test Script: > Edit Parameters

Ilustración 16 - Ejemplo de la función de Pruebas Automáticas. Fuente: <http://www.inflectra.com/SpiraTest/>



- Compartir casos de prueba entre diferentes actualizaciones

Varios casos de prueba en los proyectos pueden asociarse a diferentes lanzamientos de modo que ejecuciones de las pruebas y cobertura de las mismas pueden ser informadas conjunta o separadamente. Esto es especialmente importante en el último paso del proceso de prueba del hospital. En la quinta etapa del proceso, la prueba de regresión se basa en versiones anteriores. De este modo ya están guardadas en el programa y se puede analizar la calidad de la nueva actualización con respecto a anteriores lanzamientos con facilidad.

Test Case: Ability to create new book [TC:000002]

Name: Ability to create new book

Overview | Req. Coverage # | Test Runs # | **Releases #** | Incidents # | Attachments # | History #

Available Releases:

<input checked="" type="checkbox"/>	Name
<input type="checkbox"/>	1.0.0.0 - Library System Relea...
<input type="checkbox"/>	1.1.0.0 - Library System Relea...
<input type="checkbox"/>	1.2.0.0 - Library System Relea...

Mapped Releases:

<input checked="" type="checkbox"/>	ID	Name	Status
<input type="checkbox"/>	RL000001	1.0.0.0 - Library System Relea...	Active
<input type="checkbox"/>	RL000002	1.0.1.0 - Library System Relea...	Active
<input type="checkbox"/>	RL000003	1.0.2.0 - Library System Relea...	Active
<input type="checkbox"/>	RL000004	1.1.0.0 - Library System Relea...	Active
<input type="checkbox"/>	RL000017	1.1.0.0.0001 - Iteration 001	Active
<input type="checkbox"/>	RL000018	1.1.0.0.0002 - Iteration 002	Active
<input type="checkbox"/>	RL000019	1.1.0.0.0003 - Iteration 003	Active

The release coverage box indicates the releases that are currently mapped against the test case.
To associate releases with this test case, choose from the list above and click [Add].
You can use the [Remove] and [Remove All] buttons to remove releases that are no longer covered by the test case.

Ilustración 17 - Ejemplo de la función de Compartir casos de Prueba entre diferentes actualizaciones. Fuente: <http://www.inflectra.com/SpiraTest/>



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

- Seguimiento de errores y/o defecto

El programa permite crear, editar, asignar, rastrear, gestionar y cerrar incidentes que aparecen durante las pruebas de la mejora en desarrollo.

Name	Type	Status	Priority	Detected By	Owner	Closed On	Operating System	ID	Edit
Ability to generate customized reports	Enhancement	Resolved	1 - Critical	Joe P Smith	Joe P Smith			IN000025	Edit
Ability to associate multiple authors	Enhancement	Duplicate	1 - Critical	Fred Bloggs				IN000029	Edit
Section 508 compliance	Issue	Closed	1 - Critical	Fred Bloggs	Joe P Smith	2-Dec-2003		IN000035	Edit
Test Training Item	Training	Assigned	1 - Critical	Fred Bloggs	Fred Bloggs			IN000040	Edit
Test System Limitation	Limitation	Open	1 - Critical	Fred Bloggs				IN000045	Edit
Test System Limitation	Limitation	Assigned	1 - Critical	Joe P Smith	Fred Bloggs			IN000046	Edit
Test System Limitation	Limitation	Duplicate	1 - Critical	Joe P Smith	Fred Bloggs			IN000050	Edit
Test Change Request	Change Request	Closed	1 - Critical	Fred Bloggs	Joe P Smith	9-Dec-2003		IN000055	Edit
Sample Risk 1	Risk	Open	1 - Critical	Fred Bloggs				IN000059	Edit
Page not found bog.	Bug	Assigned	1 - Critical	Fred Bloggs	Joe P Smith		Windows 7	IN000062	Edit
Bug on book list page.	Bug	New	1 - Critical	Fred Bloggs			Windows 2000	IN000064	Edit
Editing the date on a book is clunky	Bug	Assigned	2 - High	Joe P Smith	Fred Bloggs			IN000008	Edit
Quote handling issues throughout	Bug	Resolved	2 - High	Fred Bloggs	Fred Bloggs			IN000012	Edit
The homepage hangs whilst loading	Bug	Not Reproducible	2 - High	Joe P Smith	Fred Bloggs			IN000016	Edit
Support for IBM DB2	Enhancement	Open	2 - High	Fred Bloggs				IN000019	Edit

Ilustración 18 - Ejemplo de la función de Seguimiento de errores y/o defectos. Fuente: <http://www.inflectra.com/SpiraTest/>



- Rastreo de incidentes

A diferencia de una herramienta de seguimiento de errores independiente, con esta funcionalidad se puede rastrear los defectos una vez acabado el caso de prueba y el requisito del sistema que las generó. Así el director del proyecto puede analizar la calidad "en proceso" de un sistema durante su ciclo de vida.

Incident List for Test Run Step: [RS:000003]

ID	Incident Name	Type	Status	Priority	Severity	Owner	Creation Date	Detected By
IN000007	Cannot add a new book to the system	Bug	Assigned	1 - Critical	3 - Medium	Joe P Smith	4-Nov-2003	Joe P Smith

ID	Test Step Description	Actual Results	Expected Results	Pass/Fail
RS000001	User logs in to menu screen			
RS000002	User clicks link to create book	User taken to first screen in wizard	TC000002 / TS000002	Passed
RS000003	User enters books name and author, then clicks Next	User taken to next screen in wizard	Macbeth, William Shakespeare TC000002 / TS000003	Failed

Console Output

Runner Name: - Assert Count: -
Automation Host: - Test Name: -
Message: -

Ilustración 19 - Ejemplo de la función de Rastreo de Incidentes. Fuente: <http://www.inflectra.com/SpiraTest/>

- Gestión de documentos

Todos los documentos y enlaces web, como se necesita, están disponibles en el programa. El hospital podrá organizar los documentos como desee, categorizarlos o permitir a más de un único usuario que busque los documentos que necesiten si necesidad de encontrar el documento físico original.

Internal Projects -> Library Information System -> Planning -> Testing -> Tracking -> Reporting

Requirements | Releases | Documents

Document Folders | Refresh

- Root Folder
- CreateNewAuthor
- CreateNewBook
- Design Documents
- EditExistingAuthor
- EditExistingBook
- Misc Documents
- Screen Captures
- Specifications
- Test Results
- Test Scripts

Tag Cloud

- book
- management
- functional specification
- libraries
- testing
- user interface
- wireframe

Displaying 1 - 5 out of 5 document(s) in the current folder.

Document Name	Type	Size	Edited By	Edited On	Author	ID
Sequence Diagram for Book Mgt.pdf	UML Diagram	35 KB	Joe P Smith	9-May-2006	Fred Bloggs	DC000007
Graphical Design Mockups.psd	Screen Layout	1009 KB	Joe P Smith	30-Apr-2006	Joe P Smith	DC000013
Use Case Diagram.vsd	UML Diagram	43 KB	Fred Bloggs	22-Apr-2006	Fred Bloggs	DC000012
Author Management Screen Wireframe.vsd	Screen Layout	533 KB	Fred Bloggs	1-Apr-2006	Joe P Smith	DC000008
Book Management Screen Wireframe.ai	Screen Layout	392 KB	Fred Bloggs	31-Mar-2006	Joe P Smith	DC000011

Show 15 rows per page | Displaying page 1 of 1

Ilustración 20 - Ejemplo de la función de Gestión de documentos. Fuente: <http://www.inflectra.com/SpiraTest/>



- Seguimiento de versiones de documentos

Como se requería, con este programa se lleva un registro de todos los documentos con un número que indica la versión de cada edición de los mismos.

Esto es uno de los requisitos de la normativa *HIPPA* y es requerimiento para la estandarización y creación de un documento *SOP* (Standard Operating Procedure) ya que requiere de un número identificativo. Un *SOP* es un procedimiento específico para una operación que describe las actividades necesarias para completar tareas de acuerdo a las regulaciones de la industria, leyes regionales o los propios estándares propios que, por ejemplo, es hospital dicta para realizar su negocio.

Active	Version	Version Comments	Size	Author	Upload Date	Operations
<input checked="" type="checkbox"/>	Version 1.2	Sequence diagram in UML format that provides additional detail surrounding the book managment use-case / test case	35 KB	Fred Bloggs	9-May-2006	
<input type="checkbox"/>	Version 1.1	Sequence diagram in UML format that provides additional detail surrounding the book managment use-case / test case	30 KB	Fred Bloggs	4-May-2006	> Make Active Delete
<input type="checkbox"/>	Version 1.0	Sequence diagram in UML format that provides additional detail surrounding the book managment use-case / test case	28 KB	Joe P Smith	2-May-2006	> Make Active Delete

Upload New Version

Filename:

Description:

Version*: Make this the active version

Ilustración 21 - Ejemplo de la función de Seguimiento de las versiones de documentos. Fuente: <http://www.inflectra.com/SpiraTest/>



- Soporte para interfaz móvil

Otra de las funcionalidades es la capacidad de trabajar fuera de las oficinas del hospital. Los empleados podrán trabajar fuera de sus puestos de trabajo con las aplicaciones para smartphones y tablets.

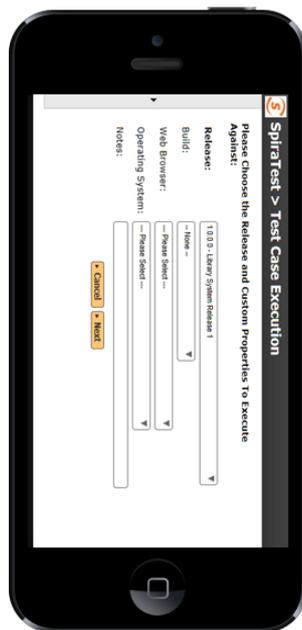
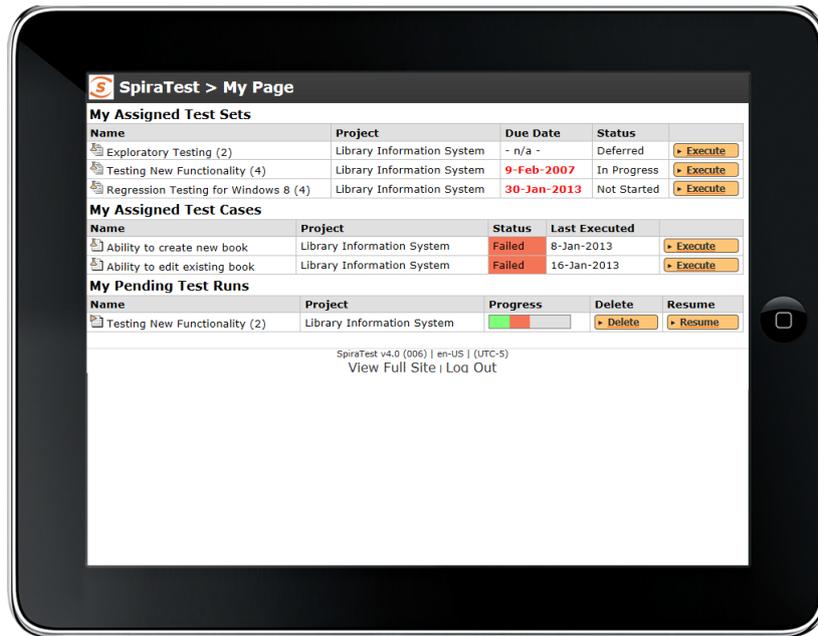


Ilustración 22 - Ejemplo de la función de Soporte para interfaz móvil. Fuente: <http://www.inflectra.com/SpiraTest/>



- Integración de email

SpiraTest posee la funcionalidad de notificar a los usuarios por email cualquier evento deseado. Esta característica resuelve los problemas de comunicación y encuentro entre los trabajadores. La comunicación interdepartamental se podrá realizar por email integrado en el programa si es necesario.

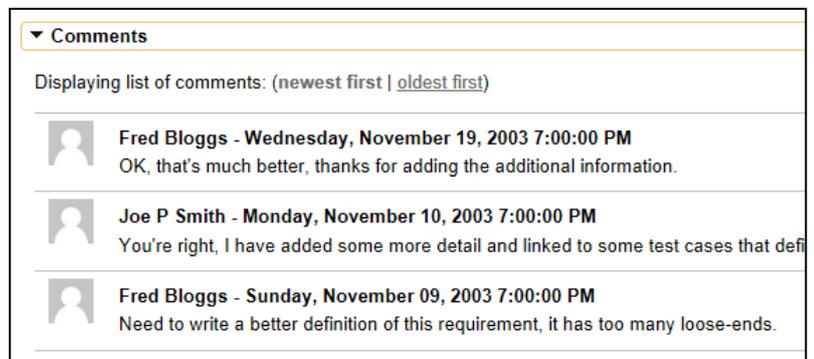
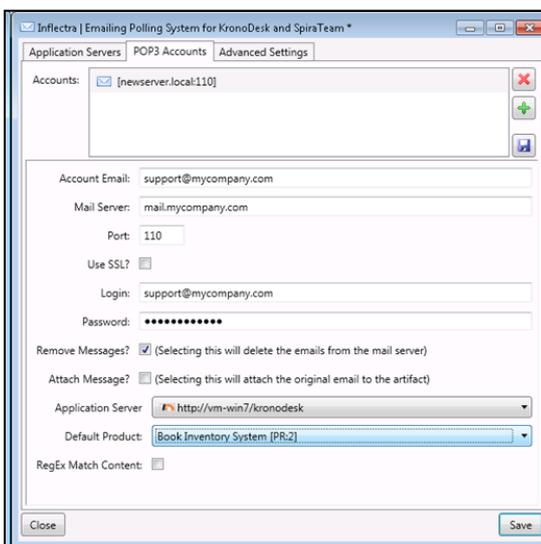


Ilustración 23 - Ejemplo de la función de Integración de email. Fuente: <http://www.inflectra.com/SpiraTest/>



- Registro del historial de cambios

Para eliminar la despersonalización anterior en los procesos de prueba, *SpiraTest* guarda todos los cambios realizados en cualquier documento para mayor seguridad.

Además todos los usuarios tienen una clave necesaria para entrar en el programa lo que garantiza la seguridad de los datos. El jefe del equipo siempre tendrá el mayor control sobre los demás trabajadores permitiéndoles o no el acceso a determinadas funciones, proyectos, documentos, ediciones etc... De la misma manera, todos los documentos y pruebas quedan almacenados electrónicamente de manera que no se pierden, es más difícil que sean robados y no ocupan prácticamente espacio.

The screenshot shows the 'History' tab in the SpiraTest application. It displays a table with 7 columns: Change ID, Change Date, Field Name, Old Value, New Value, Changed By, and Change Type. The table contains 6 rows of change records. At the bottom, there is a pagination control showing '15 rows per page' and 'Displaying page 1 of 1'.

Change ID ▲▼	Change Date ▲▼	Field Name ▲▼	Old Value ▲▼	New Value ▲▼	Changed By ▲▼	Change Type ▲▼
137	4-Feb-2013	Importance	1 - Critical	2 - High	System Administrator	Modified
137	4-Feb-2013	Release	1.0.0.0.0001	1.0.0.0.0003	System Administrator	Modified
85	15-Jan-2013	Description	The ability to add new books into the system, comp...	This is a completely new requirement, but xyz.	Fred Bloggs	Modified
82	15-Jan-2013	Requirement Type		Business Feature	Fred Bloggs	Modified
12	2-May-2006	Status	In Progress	Completed	Fred Bloggs	Modified
4	4-Mar-2005	Status	Requested	In Progress	Joe P. Smith	Modified

Ilustración 24 - Ejemplo de la función de Registro del historial de cambios. Fuente: <http://www.inflectra.com/SpiraTest/>



3.7 Matriz de requisitos y características del programa

Con el software propuesto se consiguen todos los requisitos del hospital.

Requisitos del hospital	Características del programa
Registro de pruebas	Ofrecido
Registro de edición y autoría	Ofrecido
Registro de tiempo y fecha	Ofrecido
Creación de jerarquía	Ofrecido
Almacenamiento seguro	Ofrecido
Depuración de errores	Ofrecido
Disminución del tiempo	Ofrecido
Guardado automático de los cambios	Ofrecido
Eliminación del papel y el trabajo manual	Ofrecido
Posibilidad de soporte portátil	Ofrecido

Tabla 1 - Matriz de requisitos del programa por parte del hospital

Además se proporcionan funcionalidades adicionales:

- + Evaluación rápida del progreso de los proyectos
- + Instrucciones para los empleados dedicados a ejecutar las pruebas
- + Mejora de la prueba de regresión al tener guardadas electrónicamente versiones anteriores
- + Implementación directa con los emails de los usuarios
- + Rastreo de incidentes
- + Gestión de documentos (los documentos se pueden almacenar en la nube).



3.8 Análisis de costes

Mediante el uso del software sugerido, se ha realizado un análisis de costes para llegar a determinar la cantidad de dinero que se podría ahorrar el hospital escogiendo aplicar este programa *SpiraTest*. Actualmente hay 10 trabajadores dedicados al proceso de testeo del sistema, *Epic*, que cobran anualmente una media de 50,000 USD. Para las diferentes etapas del proceso de prueba estos son los datos del tiempo que se necesitan:

Proyecto actual	Tiempo estimado
Prueba unitaria o Unit testing	2 semanas
Prueba de aplicación o Application testing	4 semanas
Prueba de Integración o Integrated testing	4 semanas
Prueba de usuario final o End User testing	4 semanas
Prueba de regresión o Regression testing	3 semanas
TOTAL	17 semanas / 85 días (5 días laborales)

Tabla 2 - Tiempos estimados de las diferentes pruebas

Como sabemos el salario medio de los empleados se llega fácilmente a lo que cobran a la hora y esto es 24.04 USD/hora, 193.32 USD/día o 961.60 USD/semana.

Debemos saber también la cantidad de tiempo que lleva crear un guion de prueba. Estos datos son muy variables pero después de preguntar a todos los 10 empleados se llega a la conclusión de que si trabajasen todos en conjunto en cada prueba, la media son 10 horas por proyecto. Con esto tenemos que para el proceso actual el coste total del proyecto son 165,876 USD como se indica a continuación.

Etapa del proceso	Semanas	Días	Salario/Emplead	Salario total	Creación de guiones
Unitario	2	10	\$1,923.20	\$19,232.00	
Aplicación	4	20	\$3,846.40	\$38,464.00	
Integración	4	20	\$3,846.40	\$38,464.00	
Usuario final	4	20	\$3,846.40	\$38,464.00	
Regresión	3	15	\$2,884.80	\$28,848.00	\$2,404.00
Total	17	85	\$16,347.20	\$163,472.00	\$165,876.00

Tabla 3 - Coste inicial del proceso



Incorporando el software propuesto para el proceso de prueba, se obtendrían unas reducciones de coste y tiempo significativas. El precio del programa son 179.99 USD (\$180 mencionados antes) al mes. Esto significa que al año el hospital deberá realizar una inversión de 2,160 USD o 41,54 USD a la semana.

Mediante el uso de este software el hospital podrá deshacerse por completo de los gastos económicos y de tiempo que conlleva crear los guiones de prueba. Además eliminando los procesos manuales de las etapas 3 y 4, estas son, la prueba de integración y la de usuario final, el tiempo de trabajo se reduciría a la mitad. Como estas dos etapas requieren un tiempo mayor que las demás porque han de ser impresas y pasadas de unos a otros, el programa daría la posibilidad a todos los trabajadores de acceder electrónicamente a las pruebas por igual.

Por lo tanto la Prueba de Integración y la Prueba del Usuario Final se reducirían de 4 a 2 semanas. En general, el proceso completo se reduciría de 17 semanas a 13 semanas con la adaptación del programa.

Aplicación	Tiempo estimado
Prueba unitaria o Unit testing	2 semanas
Prueba de aplicación o Application testing	4 semanas
Prueba de Integración o Integrated testing	2 semanas
Prueba de usuario final o End User testing	2 semanas
Prueba de regresión o Regression testing	3 semanas
TOTAL	13 semanas / 65 días (5 días laborales)

Tabla 4 - Tiempo estimado de las diferentes pruebas con el nuevo software

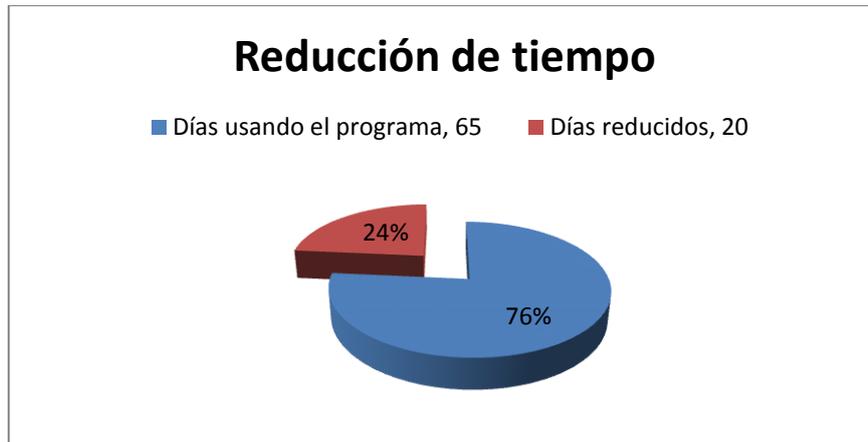


Ilustración 25 - Gráfico informativo de la reducción de tiempo

El coste de los 10 empleados se reduciría también a 125,008 USD debido a una rebaja en el tiempo de prueba. El coste global del proyecto en 13 semanas sería de 125,547 USD. El coste del programa son aproximadamente 540 USD por 13 semanas.

Etapa del proceso	Semanas	Días	Salario/Emplead	Salario total	Plus del software	
Unitario		2	10	\$1,923.20	\$19,232.00	
Aplicación		4	20	\$3,846.40	\$38,464.00	
Integración		2	10	\$1,923.20	\$19,232.00	
Usuario final		2	10	\$1,923.20	\$19,232.00	
Regresión		3	15	\$2,884.80	\$28,848.00	\$539.97
Total		13	65	\$12,500.80	\$125,008.00	\$125,547.97

Tabla 5 - Coste del proceso con el nuevo software

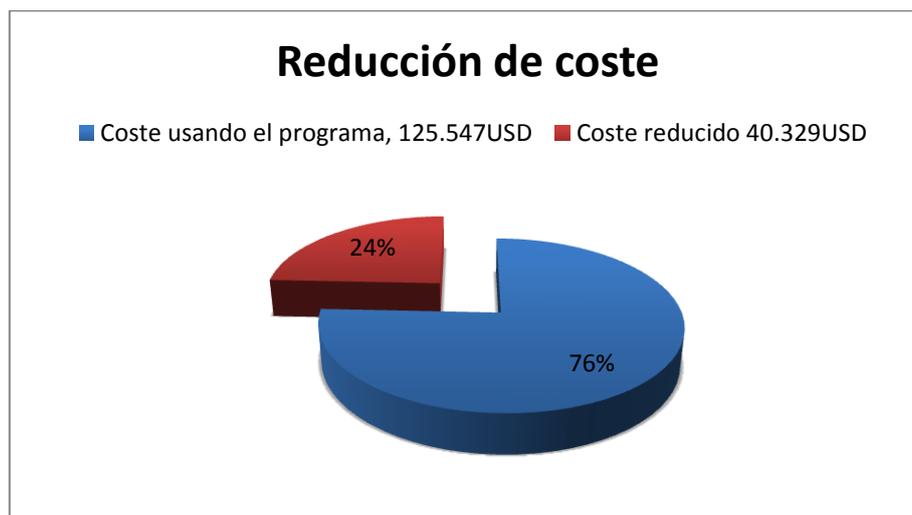


Ilustración 26 - Gráfico informativo de la reducción de coste



Con la mejora propuesta, el hospital ahorraría 40,329 USD lo que supone un 24.3%. Además se ahorrarían 20 días de trabajo entre los empleados que podrían dedicarse a otra labor o bien, anularse. Esta reducción supone también un 23.52% de tiempo de empleados.

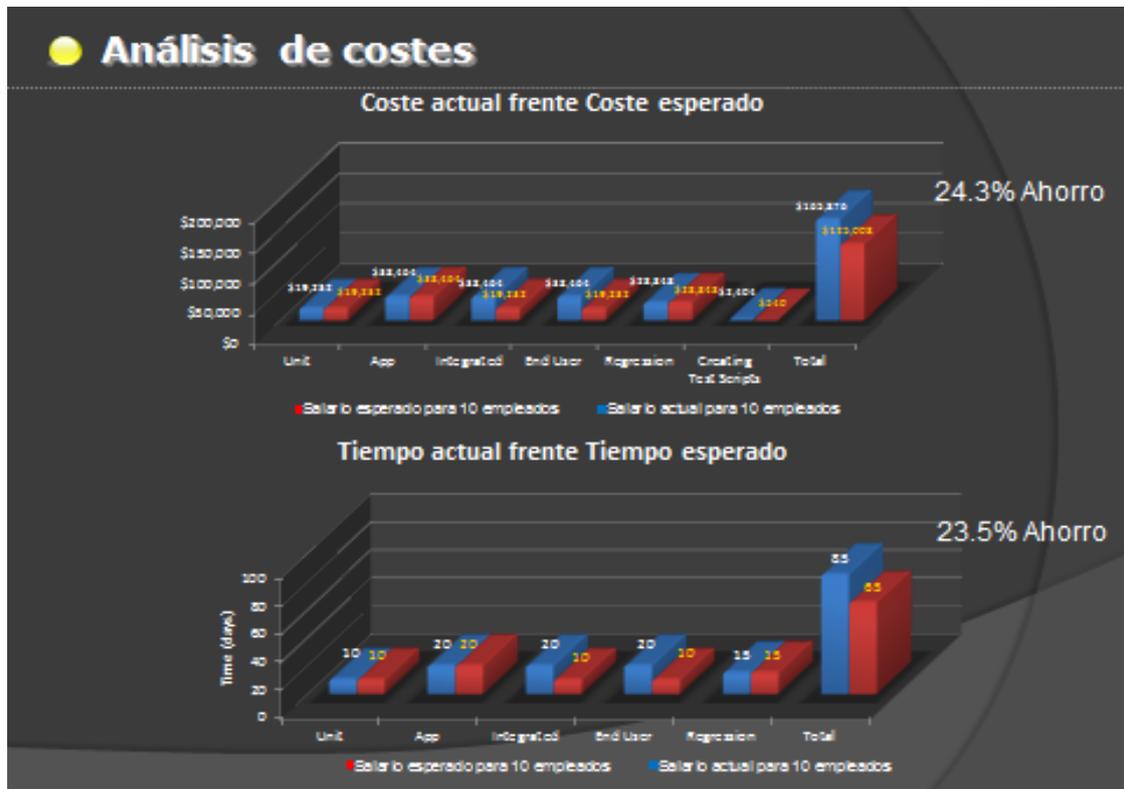


Ilustración 27 - Gráfico de comparación de costes de tiempo y dinero

Además, aunque no se incluya en los análisis, esta solución elimina la necesidad de que los *testers* tengan que moverse, caminar y encontrar a los compañeros que trabajan en el proceso en las siguientes etapas. En la mayoría de los casos, las pruebas estaban destinada a empleados en concreto para aumentar la experiencia en cada tipo de prueba y minimizar los errores, de modo que 2 empleados se encargaban de las pruebas unitarias, otros 2 empleados se encargaban de las pruebas de aplicación otros 3 se encargaban de las pruebas de integración y 3 trabajadores más para la de regresión. En las pruebas de usuario final, los trabajadores tenían que imprimir los guiones y entregárselos a usuarios finales o reales para que simulasen su trabajo normal y viesen si todo estaba correcto. Además, como este proceso no estaba estandarizado las oficinas de estos trabajadores no se habían planificado según estas necesidades ya que no sólo se dedican a esta tarea.



UNIVERSIDAD PONTIFICIA COMILLAS

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

Se adjunta un croquis aproximado de la planta donde trabajan estos 10 empleados. Se distribuían en tres plantas distintas en el edificio lo que llevaba a mayores desplazamientos. Las oficinas en azul, verde y naranja corresponden a tres plantas diferentes, además los usuarios finales pueden encontrarse en sitios completamente distintos dentro del hospital o incluso en edificios próximos ya que el hospital se encuentra situado cerca de edificios de oficinas, como se muestra en la **foto**, también propiedad de la universidad y compartían oficinas según tipo de trabajo.

Todo esto hacía que los empleados perdiesen tiempo en traslados lo que conlleva a una pérdida de dinero en salarios y productividad.

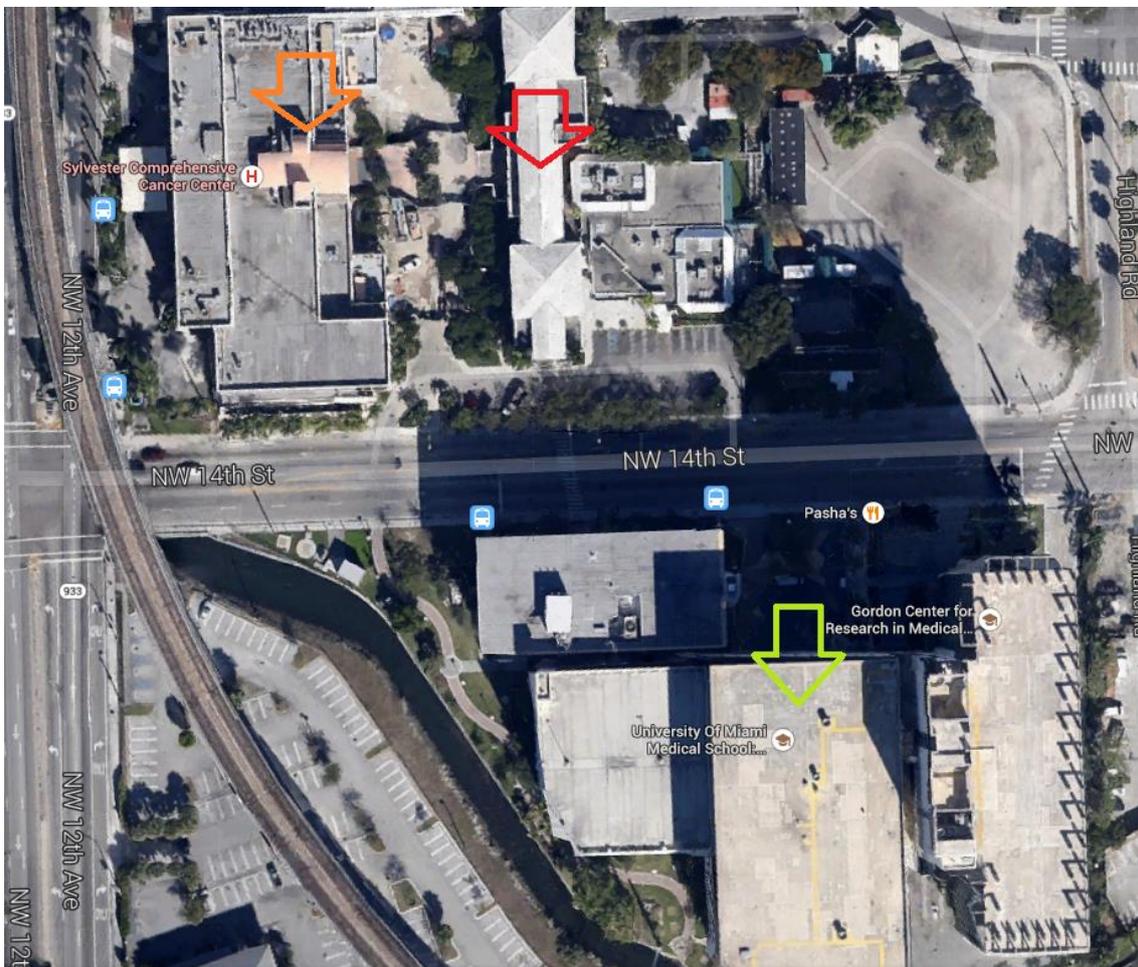


Ilustración 28 - Mapa de la disposición de los edificios del hospital y las oficinas. Fuente: Google Earth

Para las pruebas de usuario final los empleados se movían entre las tres flechas naranjas (Silvester Comprehensive Cancer Center), roja (oficinas del UMIT cuyo croquis del



UNIVERSIDAD PONTIFICIA COMILLAS

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

INGENIERO INDUSTRIAL

plano se adjunta) y verde (University of Miami Medical School que son las oficinas principales). Además en la siguiente foto se aprecia un cuarto edificio el University of Miami School of Medicine donde los usuarios finales necesarios para las pruebas podían también encontrarse.



Ilustración 29 - Mapa de la disposición de los edificios del hospital y las oficinas (completo). Fuente: Google Earth

Teniendo en cuenta que tardaban 17 semanas de media en realizar un proyecto como máximo podrían hacer 3 al año. Teniendo en cuenta que para 4 de las 5 etapas los trabajadores se encuentran en el mismo edificio aunque en distintas plantas y para la quinta tenían que concertar encuentros o simplemente ir sin aviso previo a solicitar a los usuarios finales que llevaran a cabo las pruebas de usuario final, las distancias medias que llegaban a recorrer se pueden aproximar.



Aplicaciones médicas a probar:

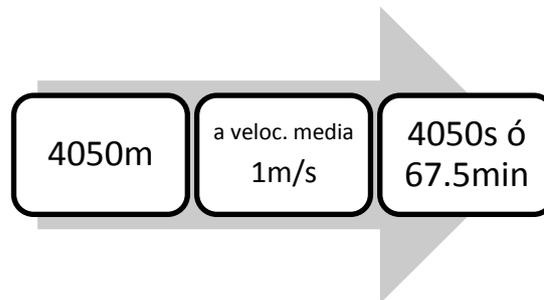
Aplicaciones/ Iniciales - Distancias (metros)			
Ambulatory	200	Security/SER	600
ASAP	200	Welcome	200
Benefits Engine	50	Willow	200
Cadence	200	PB	0
Clarity	50	PB Claims	200
EpicCare Link	0	Prelude	200
Haiku/Canto	50	Referrals	50
HB	100	Legacy	150
HB Claims	300	Third Party	200
HIM	300	Lab Systems	100
Identity	0	MyChart	700
Interfaces	0	Total	4050

Tabla 6 - Distribución de Centros de aplicaciones y distancias. Datos de aplicación de origen interno del hospital.

Las distancias se han calculado de puerta a puerta de edificio y una aproximación de lo que los trabajadores tienen que hacer dentro de los edificios. En algunos casos como en la aplicación de seguridad se tiene que ir a más de un edificio (en particular en este a los cuatro edificios).Y para *MyChart* relacionado con el equipo de UMIT se ha tenido en cuenta el croquis de la planta del edificio del UMIT.



En total para un proceso en el que se prueban todas o la mayoría de las aplicaciones se recorre un total de 4050 metros.



Cada 17 semanas se ahorrarían 67.5 minutos de desplazamientos, poco más de una hora. Si bien este ahorro no supone ni el 1% del tiempo total de esas 17 semanas y económicamente apenas 30USD, cantidades pequeñas por lo que no se han incluido en el análisis, es una mejora de comodidad y productividad para los trabajadores que se verá convertida en una mejor disposición para realizar las pruebas, en especial la de final de usuario. Ha de tenerse en cuenta además, que en algunos casos no encontraban a los usuarios finales y tenían que volver. Esto no suma cantidades significativas a los cálculos pero al tener que dejar de trabajar más de una vez para encontrar a las personas que necesitan se pierde cada vez más productividad.



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

Se muestra un croquis de la planta del edificio principal del UMIT.

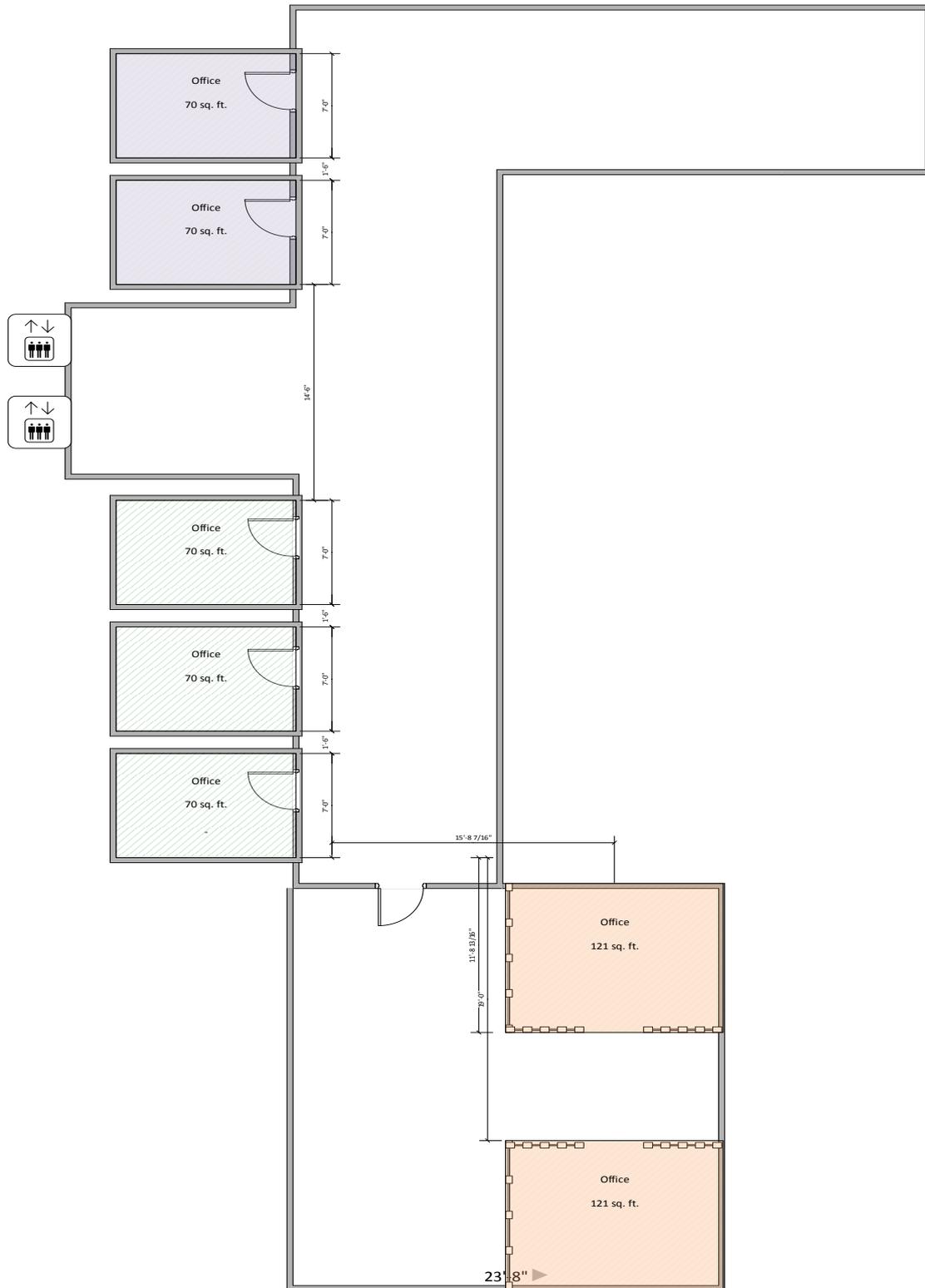


Ilustración 30 - Croquis de la planta del edificio principal del UMIT



También se podría añadir la disminución de papel utilizado. Cada guion de prueba tiene en torno a 10 – 50 páginas DIN A3 y DIN A4, esto llevaría al equivalente de un paquete de 500 folios DIN A4 estándares y 3 paquetes al año. De nuevo estos resultados son poco significativos pero la suma de este tipo de mejoras en todas las empresas marcaría la diferencia.

Ahorros en impacto ambiental:

Impacto ambiental	Cantidad	Analogía
Madera	24 dm ³	0.12 Árboles adultos
Consumo de agua	150 litros	0.75 bañeras
Consumo de combustible	1.05 litros	0.06 depósitos de combustible de moto
Emisión de gases de efecto invernadero	6.75 kilos	Conducir 180km

Tabla 7 - Impacto ambiental anual reducido por el hospital. Fuente: <http://www.araba.ehu.es>

De estos resultados se podría resaltar que en menos de 10 años con esta solución se podría evitar el talado de un árbol. Las emisiones de gases de efecto invernadero en 10 años empezarían a reflejar números algo más llamativos (cerca de 70 kilos) junto con el ahorro de agua (1500 litros).



UNIVERSIDAD PONTIFICIA COMILLAS

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

En modo de aproximación, como sabemos que con este proyecto estamos reduciendo 40,329USD y el informe del NSIT estima un ahorro para Estados Unidos de 37 mil millones de USD relativo a mejoras en el proceso de testeo, si tan sólo la mitad de este ahorro se basara en soluciones similares a este proyecto y se redujera la misma cantidad de papel en cada empresa, la pequeña contribución de cada empresa contribuiría a un gran ahorro en el impacto ambiental como se indica.

Impacto ambiental	Cantidad	Analogía
Madera	110,112 m3	550.56 Árboles adultos
Consumo de agua	688,200 litros	3,441 bañeras
Consumo de combustible	4,817.4 litros	80 depósitos de combustible de coche o 321 de moto
Emisión de gases de efecto invernadero	30,969 kilos	Conducir 825,840 km

Tabla 8 - Impacto ambiental anual aproximado en E.E.U.U. si se realizara un correcto proceso de testeo



Capítulo 4: Conclusiones



4.1 Mejoras de la IT y beneficios para el hospital

Como resultado de la adopción de este programa se quiere señalar algunas de las mejoras más relevantes conseguidas para el beneficio del hospital.

- El hospital mejorará su administración y registro de errores siendo éstos cada vez menores.
- La resolución de problemas será más rápida y efectiva, se podrá consultar problemas anteriores guardados en la memoria del programa adoptando las mismas soluciones o resolviéndolos de manera similar.
- Con el tiempo, los proyectos llegarán siempre a ser exitosos y no se tendrá que volver a reinicializar los procesos de prueba.
- El software, minorará los riesgos ya que es capaz de detectar los empleados menos habituados a las pruebas y con mayor tasa de fallos.
- El hospital provocará menos impacto medioambiental eliminando el uso de papel para los guiones.
- Se eliminará o reducirá al mínimo, al menos, la posibilidad de pérdida de guiones o robo de los mismos. Ante una catástrofe climatológica, el hospital podrá responder con confianza a la hora de mantener los datos de los pacientes a salvo.
- Con la posibilidad de seguir el trabajo de todos los miembros del equipo y las ediciones de los documentos, la transparencia en conjunto del proceso se verá beneficiada.



Mejoras a la IT	Beneficios para el hospital
La administración de errores es mejorada en un 44% y el tiempo de resolución de problemas en un 60%	Se consigue un proceso más robusto y de mayor calidad, reduciendo el fallo de los proyectos.
Se gestiona mejor la relación fallo- <i>tester</i> analizando como los cambios afectan al proceso	Reducción del riesgo y análisis de las habilidades de los empleados
Proporciona una vista general del proceso de testeo y del proyecto informando de la calidad del mismo para el control de la calidad	Aumenta el valor del proyecto reduciendo tiempo y errores.
Se considera que teniendo una descentralización del proceso haciéndolo electrónico se reduce el tiempo de ejecución de las pruebas en un 25%. Transparencia y seguimiento del proceso se estima aumentado en un 30%	Cumplimiento de estándares HIPPA y aumento de la seguridad del hospital. Eliminación de la necesidad de desplazamiento en todos o la mayoría de los casos a la hora de realizar la prueba de final de usuario.

Tabla 9 - Mejoras en IT y beneficios para el hospital



UNIVERSIDAD PONTIFICIA COMILLAS

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

- Mejoras adicionales para el control de calidad del hospital

$$\text{Ecuación de Valor Mayo} = \frac{\text{Resultados} + \text{Seguridad} + \text{Servicios}}{\text{Coste}}$$

Es de común conocimiento que para aumentar el valor de la calidad habrá que, o bien, aumentar el numerador (Resultados, Seguridad, Servicios) o disminuir el denominador (coste).

- ↑Seguridad: En cuanto a seguridad, tenemos la reducción de coste que suponía el almacenaje y supervisión de los documentos físicos. Además la seguridad que proporciona el programa sugerido es mayor que la inicial.
- ↑Servicios: SpiraTest® proporciona y ofrece una gran cantidad de servicios al equipo de UMIT por lo que se le añade valor a este término también aumentado el valor global de la ecuación.
- ↑Resultados: Como se indicaba en la parte de beneficios para el hospital, con la implementación del software se conseguirían muchos resultados. El principal es la disminución de errores en un 44% y de tiempo de resolución de problemas en un 60%.
- ↓Costes: Los costes tienen un disminución directa de 40,329USD en tiempo utilizado por los empleados para esta tarea.

Según esta ecuación de difícil valorización, podemos afirmar que estamos incrementando de las cuatro maneras posibles la calidad del proceso de prueba en el hospital. Otra de las características de control de calidad es el seguimiento de los procesos para la detección rápido de problemas. Nuevamente esto se consigue con las funciones de Seguimiento de errores y/o defectos y Rastreo de incidentes del programa.



4.2 Desventajas

En general son pocas las desventajas que esta solución tiene. Sin embargo es necesario tener en cuenta todas las que se puedan llegar a dar para estar preparados frente a ellas cuando se adapte el programa.

Inconveniencias o desventajas del uso de Informes Médicos Informáticos y del uso de *SpiraTest*[®]:

- Los empleados necesitan ser entrenados para el uso de estos programas. Esta es la mayor inconveniencia ya que requiere trabajo y una pequeña inversión.
- Al contratar nuevos empleados siempre se les tendrá que entrenar o exigir que tengan conocimientos acerca de estos programas. Tanto empleados menos preparados para trabajar con el *EMR* como los que vayan a trabajar en el UMIT. Esto reduce el número de personas contratables. No obstante, los programas *EMR* ya están extendidos extensamente en casi todo el mundo y la gran mayoría de los profesionales dedicados a empresas de cuidado de la salud están ya relacionados con este programa.
- Del mismo modo que con los informes en formato de papel, con este sistema los datos son vulnerables a *hackers* que pueden llegar a interceptar grandes cantidades de información de una forma más rápida y anónima. Si bien esto es cierto, los hospitales cuentan con una elevada seguridad informática como es el caso del hospital de la Universidad de Miami gracias al equipo del UMIT.
- Pequeña inversión inicial o mensual, según se desee contratar el servicio.

Además en algunos hospitales, si el programa *EMR* que el hospital usa es basado en un servidor de internet:

- El coste del servidor a largo plazo puede llegar a ser más caro que el uso antiguo de informes en papel.
- Vulnerabilidad a interrupciones en el servicio de internet.



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

- La información clínica almacenada junto con otras prácticas de pacientes pueden ser mezcladas.
- Si el negocio quiebra, toda la información desaparece rápidamente.



4.3 Proceso de testeo sugerido con SpiraTest®

Mediante la incorporación del software recomendado se presenta el organigrama simplificado para el nuevo proceso de pruebas.

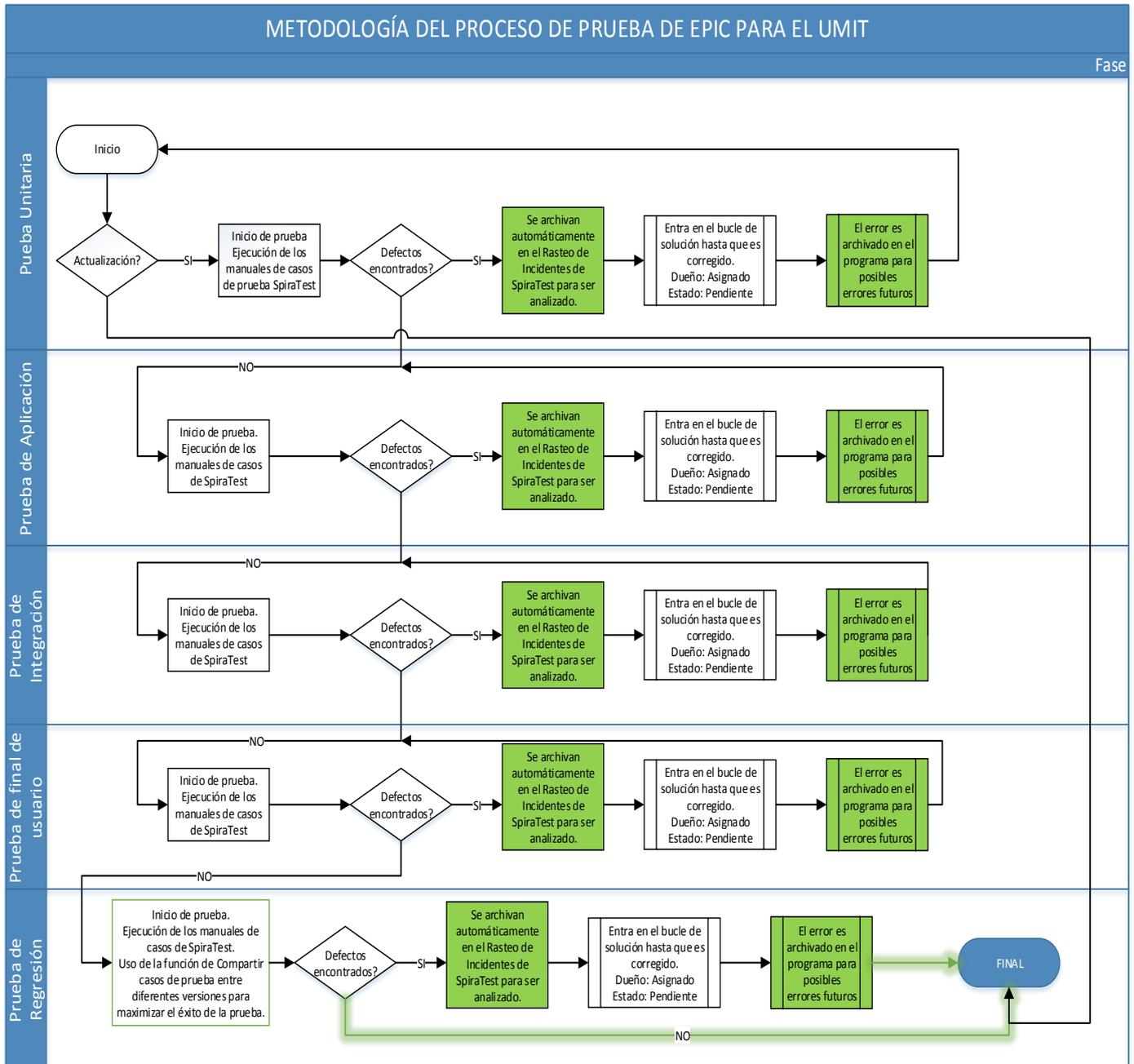


Ilustración 31 - Organigrama de trabajo con la adaptación de SpiraTest®

Con el objetivo de facilitar la interpretación del organigrama, el bucle de solución se sustituye por un subproceso pero a continuación se muestra el desglose de este bucle.

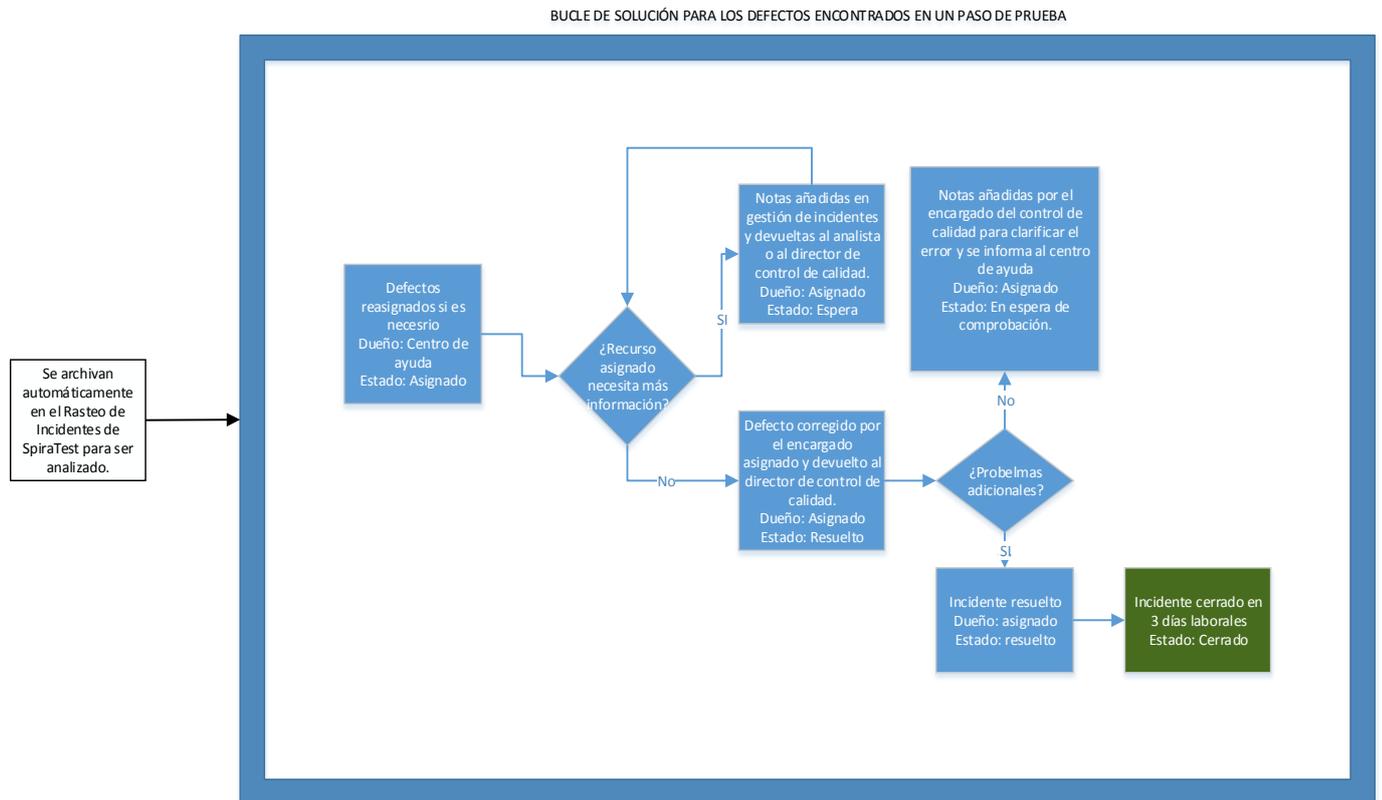


Ilustración 32 - Desglose del bucle de solución

Con la aplicación del nuevo software este sería el nuevo organigrama de trabajo para el proceso de prueba del software.

Al empezar el proceso, se ejecuta la Prueba de unidad como se hacía antes. Sin embargo, ahora se podrá ejecutar la función de manuales de casos de prueba se *SpiraTest* de forma que el modo de proseguir de todos los probadores o *testers* del sistema será el mismo y realizarían las mismas pruebas. Con esta implementación, tanto los guiones de prueba como su relleno serán electrónicos. Si se encuentra algún defecto o fallo en la prueba queda automáticamente guardado en la función de gestión de errores, así como el cuándo y quién. Esta información es importante porque después el coordinador o responsable de calidad



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

puede ejecutar el rastreo de incidentes para ver por qué la prueba ha fallado y valorar el trabajo del probador.

De la misma manera que antes de la adopción del programa sugerido, cuando un fallo es encontrado entra en el bucle de solución que se muestra arriba. Aquí, cuando el fallo es encontrado después de la documentación y análisis por parte del analista responsable es enviado al centro de ayuda, que sigue siendo ajeno al UMIT. Si el centro de ayuda necesita más información se saca de la gestión de incidentes y es devuelto al centro. En el momento en el que no se necesita más información (lo habitual) se resuelve el problema y se devuelve al responsable del control de calidad. Si no hay problemas adicionales el incidente se da por resuelto en un máximo de tres días laborales y queda archivado en el registro para posterior información.

Este proceso es el mismo para las dos siguientes etapas, la Prueba de Aplicación y la de Integración. En estas cobra mayor importancia el programa. En la prueba de Aplicación con la función de pruebas automáticas se puede probar mayor cantidad de aplicaciones individuales en el mismo tiempo. También el registro de incidentes ayuda al éxito de esta prueba.

Como se muestra anteriormente, la Prueba de Integración también se ve beneficiada por este programa ya que era la prueba con mayor porcentaje de errores y se ven disminuidos. El soporte electrónico agiliza el proceso y la misma función de rastreo de incidentes aumenta la cantidad de pruebas exitosas.

En la Prueba de Usuarios finales el proceso es el mismo pero con la mejora de que los empleados del UMIT no tienen, en la mayoría de los casos, que desplazarse para encontrar a los usuarios reales.

En la última etapa el proceso es el mismo que para la primera prueba. Este último paso se ve favorecido por el programa con la función de Compartir casos de prueba. Se aumenta de nuevo los casos exitosos al poder comparar esta prueba a pruebas anteriores.



4.4 Conclusión final

Con todo lo expuesto en esta memoria, se considera que la adaptación de un programa de apoyo al proceso de prueba del software principal del hospital, *Epic*, en torno al cual se desarrollan prácticamente todas las funciones rutinarias es esencial para la salud, tanto financiera como legal del mismo. Además, si no se aleja esta adaptación del periodo de tiempo en el que se desarrolló este proyecto, se recomienda que dicho programa de apoyo sea el mencionado y analizado *SpiraTest*.

Como conclusión final, se resumen los principales y mayores beneficios aportados por el programa a la empresa:

- Ahorro de tiempo: un 44% en la administración de errores y un 60% en su resolución. En particular, se estima un ahorro del 23.5% en el tiempo empleado por los encargados de realizar los procesos de prueba de un proyecto (20 días).
- Ahorra de dinero: Al menos, un 24.3% en el proceso de prueba. (40.329USD).
- Se aumenta la productividad de la empresa. Además, al usar menos recursos se mejora la eficiencia.
- La seguridad aumenta. Mayor seguridad en el cumplimiento de las regulaciones de las leyes de protección de datos.
- El control de calidad mejora, el tiempo de reacción ante fallos es menor y el modo de atajarlos más efectivo.
- Se tiene un mayor control financiero en esta tarea.
- Se disminuye el impacto ambiental haciendo el hospital más ecológico.



Capítulo 5: Referencias



5.1 Bibliografía

- [SANG08]** SOP Workshop: Workshop in a Book on Standard Operating Procedures for Biotechnology, Health Science, and Other Industries
- [PATT05]** Software Testing
- [MADI05]** **Process Mapping, Process Improvement and Process Management**
- [BEAV05]** The Practical Guide to HIPAA Privacy and Security Compliance
- [KANE01]** Lessons Learned in Software Testing: A Context-Driven Approach
- [BECK02]** Test Driven Development
- [CART01]** Electronic Medical Records: A Guide for Clinicians and Administrators

5.2 Enlaces web

- www.socra.org SoCRA 20th Annual Conference Session Descriptions.
- www.brampton.com What is a Standard Operating Procedure (SOP)?
- www.hhs.gov SUMMARY OF THE HIPAA PRIVACY RULE
- www.emrsimplyput.com Tag Archives: electronic medical records;
What is EMR? Server-Based EMRs
Web-Based EMRs Benefits of EMR
- www.ahd.com American Hospital Directory



Capítulo 6: Apéndices



6.1 Regulaciones básicas para la utilización de informes electrónicos y firmas electrónicas en la industria.



3109 W. DR. MARTIN LUTHER KING, JR. BLVD., SUITE 250 • TAMPA, FL 33607 USA
TEL: 813/960-2105 • FAX: 813/264-2816 • WWW.ISPE.ORG

Risk-Based Approach to 21 CFR Part 11

The 21 CFR Part 11 regulation is a comprehensive piece of legislation that outlines the controls necessary for the regulated industry to utilize electronic records and electronic signatures.

Without careful interpretation, however, the requirements can lead to over-engineered solutions that adversely impact the productivity of the industry without providing added benefit to patient health. The goal of this paper is to provide the philosophy necessary to apply risk management, and to encourage manufacturing innovation and technological advances. This philosophy is based on the ideas in the new FDA cGMP initiative. We believe that this approach is equally applicable to all FDA regulated industries.

We are currently working on more detailed material based on the philosophy presented here, covering all sections of 21 CFR Part 11 but focusing on key areas. This material includes more detailed definitions, specific processes for defining risk and identifying appropriate controls, and implementation examples. We will present this material for your consideration in the near future.

The key areas of the regulation which require attention are:

- The definition of an electronic record
- Audit trails
- Electronic copies for inspection
- Retention and maintenance of records
- Hybrid and procedural solutions
- Application of electronic signatures

The use of a risk-based approach to Part 11 would allow the regulated industry to analyze their processes, identify GxP records, and implement appropriate controls to mitigate risks.

The suggested risk-based approach has the following steps, that cover both the scope and selection of appropriate controls:

- User firms identify and define GxP electronic records and signatures, based on the predicate rules, criticality of the process, and risk to product safety, efficacy and quality.



ISPE – Risk-Based Approach to 21 CFR Part 11

- User firms implement controls commensurate with the criticality of the electronic record, and risks identified for that record. These controls should be documented and justified with reference to the identified risks.

This is a top-down approach (“is it a GxP record?”), rather than a bottom-up approach (“is it an electronic record?”). This approach focuses on the critical records as opposed to all electronic records created by a firm.

Definition of Electronic Record

Reference: 21 CFR Part 11.1 (b).

The current interpretation of what is in scope is too broad. This leads to a potential stifling of innovation, and draws focus away from the most critical areas. This is not in the spirit of a risk-based approach.

User firms should identify and define the high impact GxP electronic records and signatures, based on the predicate rules, criticality of the process, and risk to the quality, safety, identity, purity, or strength of the product.

The focus of effort should be on records that have a high impact, i.e. those records upon which quality decisions are based. Examples of high impact records are batch records and laboratory test results. Examples of records with low impact include environmental monitoring records not affecting product quality, training records, and internal computerized system information such as setup and configuration parameters.

Existing security measures and established validation measures are more than adequate to ensure the integrity of lower impact records – additional Part 11 controls are cumbersome and add very little value.

Internal system information not identified in the predicate rules is low impact. The integrity of this information can be assured by system validation, change control, configuration management, and routine security features. These may be controlled by a suitable established procedure – additional Part 11 controls are not required. Paper records of such events are acceptable.

Software should not be considered as being GxP electronic records with regard to Part 11. Industry and FDA have worked for many years on developing approaches for dealing with hardware and software in the GxP environment based on validation of systems, configuration management, change control, and adequate procedures and plans for maintaining the validated state. These approaches have been widely adopted and very successful in meeting GxP requirements. Considering software as GxP electronic records has little practical benefit, as well as discouraging firms from adopting innovative technological solutions. It should be noted that not considering software as a GxP



ISPE – Risk-Based Approach to 21 CFR Part 11

electronic record is in conflict with CPG 7132a.11 “CGMP applicability to hardware and software”.

Copies of Electronic Records

Reference: 21 CFR Part 11.10 (b).

The user firm should provide an investigator with reasonable and useful access to records during an inspection. The provision of electronic copies under Part 11 can bring significant technical challenges, especially for complex systems. We propose that user firms should be able to meet the requirement for accurate and complete copies by, in order of preference:

- Using industry standard portable formats where possible, if the use of such formats brings more benefits than disadvantages.
- Utilizing established automated conversion or export methods where available, to make copies in a more common format (e.g. PDF or paper copies).
- Allowing inspection and review of records on the firm’s site, using the firm’s hardware and software, following the firm’s established procedures and techniques for accessing those records.

The key concepts are that the content and the meaning of the GxP records are preserved during copying, and that these GxP records are available in human readable form.

Retention and Maintenance of Records

Reference: 21 CFR Part 11.10 (c).

Archiving of electronic records can be problematic in two main ways: volume and processability.

If the definition of Part 11 scope is clarified, and interpreted according to a risk-based approach, then volume ceases to be such a significant problem. For example, an encapsulation machine with a check-weigher can measure two million capsule weights for a single batch. The batch record may only use these values to generate a curve that shows the batch weight distribution falls within validated norms. Only the weight distribution curve is meaningful and the individual data points need not be archived, provided the generation process of the distribution curve is validated. This meets the expectation of the predicate rule 21 CFR 211.188, yet considerably eases the record-keeping burden.

Archived records must be retrievable; it is not, however, always necessary to reprocess electronic records. The content and meaning of retrieved records must be evident.



ISPE – Risk-Based Approach to 21 CFR Part 11

Archiving to non-electronic media such as microfilm, microfiche, and paper, is an acceptable alternative to electronic media in some cases, as long as the content and meaning of the electronic record is not compromised, and the archiving process is validated.

Printing copies of records held electronically, and reviewing and subsequently approving them by means of a handwritten signature, to produce the master regulatory record for long term keeping can be effective and efficient. The user firms must ensure that any contextual and background information required to interpret the records is also kept. For high impact records it may be essential that electronic copies of the record are also retained – although not necessarily in reprocessable form.

Reprocessability is an additional challenging requirement placed on user firms when using electronic records. It should be the exception, rather than the rule. Those records needing reprocessability should be clearly identified and defined by the user firm, based on risk. For clinical records, reprocessability may be important, but not for manufacturing records where the main requirement is to support integrity and reliability of the records.

In most cases the value of the ability to reprocess records decreases as time passes. When a product is relatively new to the market there may be a host of issues not well understood, such as unusual drug interactions or environmental factors. The ability to analyze clinical data for new drug products based on new information gleaned from post-market reporting is important. However, as the product is used for several years without major problems, a profile begins to emerge of the drug as safe under normal conditions of use.

A risk-based approach should be used in deciding whether to migrate older records to a new format. Based on risk assessment data, a decision would be made whether it is important to retain processability of older data. If risk is sufficiently low, the firm may choose to write the records to a common, simple, permanent format for archival (PDF, microfiche, paper, etc.). The GxP records and their context and meaning are retained, and the key elements such as the audit trail can be captured and retained. The principal compromise accepted here is sacrifice of the ability to *easily* search and manipulate the records.

Audit Trails and Data Security

Reference: 21 CFR Part 11.10 (e).

Currently the audit trail requirement is being applied too widely. Audit trails should be applied only in cases where operator actions create, modify, or delete high impact GxP records.

If risk analysis shows that adequate data integrity can be achieved through rigorous security controls, user firms should be expected to take a justified decision that addition of audit trails does not provide significant benefit. Audit trails would still be important



ISPE – Risk-Based Approach to 21 CFR Part 11

for systems in which the users are expected to modify data as a routine part of the business process.

We realize and agree that security and integrity of data is of key importance. Audit trails are only one mechanism for ensuring data integrity, together with other physical, logical, and procedural security measures. Data integrity is also supported by established validation methods that prove system operation and data integrity features, and robust change control that maintains the control established by validation.

In many cases where the data should not be modified, the ability to do so can be restricted such that audit trails are not required. For example, if an HPLC data system writes to a secure directory, and validated system controls do not allow overwriting of files, an automated audit trail adds little to the integrity of this data.

Current industry security practice fulfils many of the Part 11 requirements for data integrity. User firms should be encouraged to follow existing international standards, such as ISO/IEC 17799 *Code of practice for information security management*, and their existing information security policies, as well as generally accepted good Information Technology practice.

Procedural and Hybrid Solutions

Reference: 21 CFR Part 11.10, 21 CFR Part 11.70, and others.

The continued use of hybrid systems, where electronic and paper records and signatures co-exist, does not increase the risk to the product or patient, if adequate procedural controls are established. Such procedural controls are the basis of current GxPs, and should be acceptable in this area also.

Recent draft FDA guidance on 21 CFR Part 11 has concentrated on complex technical methods of linking records to each other and to handwritten signatures. We suggest that in many cases procedural controls are sufficient, and effective. Procedural links are extensively used elsewhere in the GxP environment. For example, records, documents, and materials are related to each other by means of a batch number or component code. Such procedural controls are indeed the bedrock upon which the GxPs are based. The user firm should ensure that procedural controls are established and effective, and that validated system features also mitigate the risk to stored electronic records of unauthorized, or uncontrolled, change, copying, or loss.

Electronic Signatures

Reference: 21 CFR Part 11.100.

User firms should define where they need to apply signatures, based on predicate rule requirements, criticality of the process, and risk to the product. User firms should define whether these should be handwritten or electronic.



ISPE – Risk-Based Approach to 21 CFR Part 11

Where signatures are currently used with paper records, it may not always be appropriate to apply an electronic signature. With paper records, handwritten signatures are applied in two different cases:

1. As a legally binding signature, when there is a regulatory expectation
2. As a convenient way of identifying a person

In the second of these cases, the appropriate electronic equivalent would be the system logging the user-id, or an entry in the audit trail, rather than an electronic signature. Automation of the process may, in some circumstances, remove the need for identification, because sufficient assurance is provided through validation. Signatures should only be mandatory when explicitly required by a predicate rule.

Conclusion

This paper has attempted to demonstrate that applying a risk-based approach to electronic records in a GxP context can lead to controls appropriate to the criticality and impact of those records.

This allows measures aimed at a high degree of integrity to be established for records that directly impact the quality, safety, and efficacy of the product, while permitting a less rigorous approach for records of lower criticality.

This overall philosophy, based on the ideas in the new FDA cGMP initiative, would encourage manufacturing innovation and technological advances without increasing risk to the patient or to product quality.



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

6.2 Lista de verificación para la regulación HIPAA de protección de datos (parte).

HIPAA / HITECH Audit Checklist			
#	Control Name	Definition	Testing Procedure
Administrative 164.308(a)			
1	Risk Analysis (R)	Conduct an accurate and thorough assessment of the potential risks and vulnerabilities to the confidentiality, integrity, and availability of electronic protected health information held by the covered entity.	1.1 Run internal and external network vulnerability scans at least quarterly and after any significant change in the network (such as new system component installations, changes in network topology, firewall rule modifications, product upgrades) 1.2 Perform external and internal penetration testing at least once a year and after any significant infrastructure or application upgrade or modification (such as an operating system upgrade, a subnetwork added to the environment, or a web server added to the environment) 1.3 Verify that the penetration test includes network layer and application layer testing. 1.4 Verify that noted exploitable vulnerabilities were corrected and testing repeated.
2	Risk Management (R)	Implement security measures sufficient to reduce risks and vulnerabilities to a reasonable and appropriate level to comply with [164.306(a)].	2.1 Verify that the information security policy includes an annual risk assessment process that identifies threats, vulnerabilities. 2.2 Verify that the information security policy includes an annual risk assessment process that results in a formal risk assessment. 2.3 Review risk assessment documentation to verify that the risk assessment process is performed at least annually.
3	Sanction Policy (R)	Apply appropriate sanctions against workforce members who fail to comply with the security policies and procedures of the covered entity.	3.1 Verify that appropriate sanctions are detailed throughout policies and procedures against workforce members who fail to comply with the security policies and procedures of the covered entity.
4	Information System Activity Review (R)	Implement procedures to regularly review records of information system activity, such as audit logs, access reports, and security incident tracking reports.	4.1 Verify that the security awareness program requires personnel to acknowledge (for example, in writing or electronically at least annually that they have read and understand the company's information security policy. 4.2 Obtain and examine security policies and procedures to verify that they include procedures to review security logs at least daily and that follow-up to exceptions is required. 4.3 Through observation and interview verify that regular log reviews are performed for all system components.
5	Assigned Security Responsibility (R)	Identify the security official who is responsible for the development and implementation of the policies and procedures required by this subpart for the entity.	5.1 Verify that information security policies clearly define information security responsibilities for all personnel. 5.2 Verify the formal assignment of information security to a Chief Security Officer or other security-knowledgeable member of management. 5.3 Verify that responsibility for creating and distributing security policies and procedures is formally assigned. 5.4 Verify that responsibility for monitoring and analyzing security alerts and distributing information to appropriate information security and business unit management personnel is formally assigned. 5.5 Verify that responsibility for creating and distributing security incident response and escalation procedures is formally assigned. 5.6 Verify that responsibility for administering user account and authentication management is formally assigned. 5.7 Verify that responsibility for monitoring and controlling all access to data is formally assigned.
6	Authorization and/or Supervision (A)	Implement procedures for the authorization and/or supervision of workforce members who work with electronic protected health information or in locations where it might be accessed.	6.1 Confirm that privileges are assigned to individuals based on job classification and function (also called "role-based access control" or RBAC).
7	Workforce Clearance Procedure (A)	Implement procedures to determine that the access of a workforce member to electronic protected health information is appropriate.	7.1 Confirm that access rights for privileged user IDs are restricted to least privileges necessary to perform job responsibilities.
8	Termination Procedure (A)	Implement procedures for terminating access to electronic protected health information when the employment of a workforce member ends or as required by determinations made as specified in paragraph (a)(3)(ii)(B) of this section.	8.1 Custom application accounts, user IDs and/or passwords are removed when the employment of a workforce member ends 8.2 Select a sample of users terminated or transferred in the past six months, and review current user access lists to verify that their IDs have been deactivated or removed. 8.3 Verify that inactive accounts over 90 days old are either removed or disabled. 8.4 Verify that any accounts used by vendors to access, support and maintain system components are disabled, and enabled only when needed by the vendor.
9	Isolating the Clearinghouse Funct. (R)	If a health care clearinghouse is part of a larger organization, the clearinghouse must implement policies and procedures that protect the electronic protected health information of the clearinghouse from unauthorized access by the larger organization.	9.1 Verify the health care clearinghouse policies and procedures that protect the electronic protected health information of the clearinghouse from unauthorized access by the larger organization.
10	Access Authorization (A)	Implement policies and procedures for granting access to electronic protected health information, for example, through access to a workstation, transaction, program, process, or other mechanism.	10.1 Verify that each user is authorized to use the system according to company policy by obtaining and examining an authorization form for each user ID
11	Access Establishment & Modification (A)	Implement policies and procedures that, based upon the entity's access authorization policies, establish, document, review, and modify a user's right of access to a workstation, transaction, program, or process.	11.1 Confirm that documented approval by authorized parties is required (in writing or electronically) for all access, and if it must specify required privileges. 11.2 Confirm that access controls are implemented via an automated access control system.
12	Security Reminders (A)	Periodic security updates.	12.1 Compare the list of security patches installed on each system to the most recent vendor security patch list, to verify that current vendor patches are installed. 12.2 Examine policies related to security patch installation to verify they require installation of all critical new security patches within one month. 12.3 Interview responsible personnel to verify that processes are implemented to identify new security vulnerabilities. 12.4 Verify that processes to identify new security vulnerabilities include using outside sources for security vulnerability information. 12.5 Interview responsible personnel to verify that processes are implemented to identify new security vulnerabilities, and that a risk ranking is assigned to such vulnerabilities.
13	Malicious Software (A)	Procedures for guarding against, detecting, and reporting malicious software.	13.1 Verify that anti-virus software is deployed, if applicable. 13.2 Verify that all anti-virus programs detect, remove, and protect all known types of malicious software (for example, viruses, Trojans, worms, spyware, adware, and rootkits). 13.3 Verify that the anti-virus software is current. 13.4 Verify that all anti-virus software is capable of generating logs, log generation is enabled, and logs are retained in accordance with HIPAA. 13.5 Obtain and examine the policy and verify that it requires updating of anti-virus software and definitions. 13.6 Verify that the anti-virus software is enabled for automatic updates and periodic scans.
14	Login Monitoring (A)	Procedures for monitoring log-in attempts and reporting discrepancies.	14.1 Verify all valid and invalid logical access attempts are logged. 14.2 Verify that any vendor remote access accounts are monitored while being used. 14.3 Verify use of identification and authentication mechanisms is logged.
15	Password Management (A)	Procedures for creating, changing, and safeguarding passwords.	15.1 In addition to assigning a unique user ID, employ at least one of the following methods to authenticate all users:



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

Verification by configuration review	Verification by related documentation	Verification through interviewing	Verification through observation of processes and actions	Findings and Recommendations
	X	X	X	
	X	X	X	
	X	X	X	
	X	X	X	
	X		X	
	X		X	
	X			
	X		X	
	X		X	
		X	X	
	X		X	
	X		X	
	X		X	
	X		X	
	X		X	
	X		X	
	X		X	
	X		X	
	X		X	
	X		X	
X	X		X	
	X		X	
	X		X	
	X		X	
	X		X	
	X		X	
	X		X	
		X		



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL
