

Garantía de calidad Software

v3.0.1

Ingeniería del Software



PiKey Team- P_KT :

Jesús Aguirre Pemán
Enrique Ballesteros Horcajo
Jaime Dan Porras Rhee
Ignacio Iker Prado Rujas
Alejandro Villarín Prieto

26 de Marzo de 2013

Índice general

I. Propósito del SQA	2
II. Documentos de referencia	5
III. Gestión	7
IV. Documentación	9
1. Propósito	9
2. Requisitos mínimos de documentación	10
3. Otra documentación	10
V. Estándares, prácticas, convenciones y métricas	12
1. Propósito	12
2. Estándar de documentación	12
3. Estándar de código y comentarios	13
4. Estándar de verificación y prácticas	13
VI. Revisiones del software	15
1. Propósito	15
2. Requisitos mínimos	15
3. Otras revisiones y auditorías	15
VII. Prueba	17
VIII. Informe de problemas y acción correctiva	19
IX. Herramientas, técnicas y metodologías	21
X. Control de medios	23
XI. Control de proveedor	25
XII. Colección de registros, mantenimiento y conservación	27
XIII. Formación	29

XIV.Gestión del riesgo	31
-------------------------------	-----------

XV.Procedimiento de cambio e historia del plan SQA	33
---	-----------

Parte I

Propósito del SQA

La Garantía de Calidad del Software consiste en un medio de seguimiento de los procesos de ingeniería de software y métodos utilizados para asegurar la calidad del software que se produce en el proyecto.

La SQA abarca todo el proceso de desarrollo de software, incluyendo procesos tales como la definición de requerimientos, diseño de software, programación, control de código fuente, revisiones de código, gestión de cambios, gestión de configuración, pruebas o gestión de versiones.

El plan de Garantía de Calidad del Software define las actividades específicas a llevar a cabo en este proyecto. Contiene una lista de comprobación para las actividades que se deben llevar a cabo para asegurar la calidad del producto y garantizar que KIKE-Hostelería ® cumple los requisitos especificados en la documentación.

La SQA engloba:

- Enfoque de gestión de calidad.
- Tecnologías de IS (métodos y herramientas).
- Revisiones Técnicas Formales.
- Estrategia de pruebas.
- Control de la documentación y de cambios.
- Procedimientos que aseguren ajustes a los estándares de IS
- Mecanismos de medición y generación de informes

Las Revisiones Técnicas Formales serán la base de las correcciones de los documentos, y el principal elemento de garantía del software. En cada iteración del Proceso Unificado se producirán nuevas correcciones de los documentos existentes, nombradas según se ha visto en el apartado "Nombrado" del documento **Gestión de la Configuración**.

Garantía de calidad

Además, al finalizar cada iteración el proyecto pasa por una fase específica de corrección donde todos los miembros de P_KT revisan cada documento.

El control de la documentación ya ha sido comentado en el documento de **Gestión de la Configuración**

Parte II

Documentos de referencia

Co, esto se hace al final cuando este acabado el documento

- Documento Gestión de la Configuración. IEEE Std 730TM -2002

(- Proporciona una lista completa de cualquier documento referenciado en el plan o utilizado en su elaboración.)

Parte III

Gestión

La estructura en la gestión de la calidad en nuestro grupo es lineal, es decir, prácticamente todos los integrantes revisan cada documento. Esto siempre bajo la organización del jefe del proyecto que indica a cada integrante qué parte de cada documento revisa.

En cuanto al documento de SQA, Jesús Aguirre se encarga de la redacción, Iker Prado de la organización de las tareas, Kike Ballesteros de la gestión, Alejandro Villarín de las herramientas y Jaime Porras del mantenimiento.

En la parte de redacción se incluyen la introducción del documento de SQA, la relación de los documentos de referencia, y la colección de registros, mantenimiento y conservación. En la organización se incluyen la asignación de tareas, la creación de los modelos para L^AT_EX y el seguimiento del trabajo. En la gestión se registra la organización del equipo y de las tareas. En las herramientas se escribe la relación de tareas, y en el mantenimiento se garantiza que el plan de garantía de calidad esté actualizado según va avanzando el proyecto.

Parte IV

Documentación

1. Propósito

Durante el proceso de desarrollo se generará numerosa documentación. Por el momento se han elaborado los documentos **Casos de uso**, **Especificación de requisitos**, **Glosario**, **Plan de proyecto**, que incluye los documentos **Gestión de Riesgos** y **Estimación del proyecto**, **Gestión de la Configuración**, y el actual documento, **Garantía de Calidad**.

Actualmente esta es toda la documentación existente, pero en el futuro se elaborará nueva documentación, como los manuales de usuario o la descripción del diseño del producto.

En la primera entrega, los documentos **Casos de uso** y **Especificación de requisitos** fueron revisados por Gonzalo. La segunda entrega estaba formada por los documentos **Casos de uso**, **Especificación de requisitos**, **Plan de proyecto**, **Gestión de Riesgos** y **Estimación del proyecto** que fueron revisados por otros grupos.

- **Casos de uso:** Revisado por **Grupo Diedral**, a la revisión asistieron Iker Prado y Jaime Dan.
- **Especificación de requisitos:** Revisado por **Nameless**. A la revisión asistieron Jesús Aguirre y Alex Villarín.
- **Plan de proyecto:** Revisado por **Cauchy Team**. A la reunión asistieron Jesús Aguirre, Kike y Jaime Dan.
- **Gestión de riesgos:** Revisado por **Cauchy Team**. A la reunión asistieron Alex Villarín y Jaime Dan.
- **Estimación del proyecto:** Revisado por . A la reunión asistió Iker Prado.

Toda la información referente a dichas revisiones se encuentra en las Actas de las RTF. En el futuro habrá que revisar los nuevos documentos que vayan surgiendo. Serán revisados según disponga el supervisor del proyecto, Gonzalo.

2. Requisitos mínimos de documentación

Para asegurar que la implementación del software satisface los requisitos técnicos, al menos se requiere la siguiente documentación.

- **Especificación de requisitos:** En este documento se explican los requisitos que debe cumplir el programa. Fue elaborado por los miembros del grupo P_KT, con la colaboración del cliente. Se hicieron varias reuniones con el cliente en las que este explicó qué era lo que esperaba del programa. En las siguientes reuniones se le mostraban al cliente las propuestas del grupo P_KT acerca de los requisitos y del funcionamiento del programa, y el cliente aceptaba o rechazaba dichas propuestas. De esta manera se elaboró este documento.

El resto de documentos mencionados en el IEEE 730-2002, es decir Descripción del diseño del software, Plan de verificación y validación, Manual de usuario y Plan de configuración del software se escribirán en las próximas etapas del proceso de desarrollo.

3. Otra documentación

Por el momento no se pueden identificar otros documentos aplicables al desarrollo del producto software.

(- Define toda la documentación que se va a generar durante el proceso de desarrollo. - Lista los documentos que serán revisados o auditados, así como los criterios de revisión.)

Parte V

Estándares, prácticas, convenciones y métricas

1. Propósito

En esta parte, se trata de identificar, como su título indica, qué estándares, prácticas, convenciones y métricas se van a utilizar en el SQAP, así como indicar cómo será monitoreado y asegurado el cumplimiento de los mimos.

2. Estándar de documentación

La documentación del software KIKE- Hostelería ® debe ser adecuada para que otro grupo autónomo y aislado se pueda encargar de continuar desarrollando la aplicación, así como de su mantenimiento. No se debe olvidar, que la aplicación, en un futuro, es muy probable que se amplie para más clientes.

Como ya se ha comentado más veces, la documentación está realizada en L^AT_EX siguiendo plantillas definidas por el propio equipo y los estándares de IEEE. En las plantillas siempre se sigue una estructura común:

- Documento de tipo report, tamaño del papel DIN A4, con tamaño de la fuente a 11 puntos, y disposición en dos lados (twoside, a modo de libro). La fuente es la presentada por defecto en L^AT_EX.
- Portada con título del documento, versión y fecha del mismo, nombre del grupo y componentes.
- Índice general, con el contenido del documento.
- Documento dividido en partes, que a su vez se dividen en secciones, que a su vez se dividen en subsecciones, y algunas de estas en subsubsecciones.

- Encabezado: Según el lado en el que cae (páginas pares o impares), en un extremo nombre de la sección y en el otro el nombre del documento del que se trate. Además aparece una línea de 0.4 puntos para separar el encabezado del texto.
- Pie de página: Número de la página centrado. También aparece una línea de 0.4 puntos que separa el pie de página del resto de la página. En cuanto a la numeración, las primeras páginas aparecen numeradas con números romanos en minúscula (*roman*), hasta que empieza la primera parte. A partir de ese momento, se numera desde 1 con numeración arábiga (*arabic*).

3. Estándar de código y comentarios

No se utiliza ningún estándar en concreto para ésta parte, sólo se aplican los conocimientos obtenidos en el estudio de la programación y los algoritmos. Se debe recordar que estamos trabajando con Java, un lenguaje imperativo orientado a objetos. Algunas ideas a tener en cuenta para el estilo de código y comentarios son:

- Cada clase debe tener una descripción, tanto de su función como de sus atributos. Incluirá también nombre del fichero, autor, versión y fecha.
- Se deben detallar todas las relaciones de herencia que se den.
- En cada clase, primero aparecerán los métodos y al final los atributos.
- Cada método debe tener una breve explicación sobre sus parámetros de entrada y, cuando proceda, el valor devuelto.
- Todas las partes del código que sean confusas o complejas deben estar bien documentadas.
- Sólo los métodos necesarios deben ser públicos. Por supuesto, salvo en situaciones contadas, los atributos serán privados.
- Los comentarios en una línea se escriben así: `// Comentario...`
- Si un comentario ocupa más de una línea se escribirá así:

```
/*  
 * Línea 1 comentario...  
 * Línea 2 comentario...  
 .....  
 * Línea n comentario...  
*/
```

4. Estándar de verificación y prácticas

Para la verificación y validación, se utiliza el plan propuesto por IEEE. Éste es el IEEE Std. 1012-2004 Standard for Software Verification and Validation Plans.

Parte VI

Revisiones del software

1. Propósito

La intención de las revisiones del software es detectar lo antes posible los fallos que existan en el software que estamos desarrollando, con el fin de ahorrar costes en la corrección de estos fallos. Nos centraremos en las revisiones técnicas formales, y se las haremos a los documentos que hemos producido hasta ahora, es decir, al plan de proyecto, al documento de casos de uso, a la especificación de requisitos, y al documento de gestión de riesgos. Además de detectar posibles errores, las revisiones nos permitirán asegurarnos de que nuestros documentos se ajustan a los estándares y de que se cumplen los requisitos especificados.

2. Requisitos mínimos

Mediante las revisiones nos aseguraremos de que no haya faltas de ortografía ni errores de expresión en ningún documento, y de que se respetan los estándares elegidos para cada documento.

3. Otras revisiones y auditorías

Por ahora, además de las revisiones técnicas formales, realizaremos revisiones internas llevadas a cabo por los mismos integrantes del equipo de desarrollo; y Gonzalo por su parte realizará varias revisiones independientes.

Parte VII

Prueba

No se ha realizado ninguna prueba. (- Identifica todas las pruebas no incluidas en el plan de verificación y validación.)

Parte VIII

Informe de problemas y acción correctiva

- Hasta el momento hemos llevado a cabo entregas y revisiones por parte del profesor de cada documento que íbamos realizando. Sobre cada documento nos proporcionaba ciertos problemas que había encontrado y que debíamos corregir. Hasta el momento hemos recibido la corrección de los documentos: Casos de Uso, Especificación de Requisitos Software, Plan de Proyecto y Gestión de Riesgos. En todos ellos hemos realizado las modificaciones pertinentes, siempre siguiendo lo establecido en el Plan de Gestión de Configuración Software. En último lugar hemos realizado una Revisión Técnica Formal con los distintos grupos de clase, en la cuál detectamos otros problemas y también vimos como los demás grupos han afrontado el proyecto. A partir de la revisión se realizó un informe de los errores que nuestro proyecto contenía y de como debíamos resolverlos. Todo ello está incluido en un acta acerca de cada reunión que se llevó a cabo. Actualmente ya hemos corregido todos aquellos problemas y se realizarán más revisiones con el fin de solucionar todos los posibles problemas.
- El trabajo fue repartido entre los distintos integrantes del equipo, de modo que cada uno se encargaba de una parte del proyecto, intentando siempre que fuese equitativo. Para la revisión y posterior corrección actuamos del mismo modo, ya que cada uno conoce mejor su trabajo y es capaz de detectar mejor los fallos y corregirlos. De este modo todos los miembros se centraron en su sección de los distintos documentos, aunque la posterior modificación fue realizada de distinto modo. En particular este fue el reparto en cada documento:
 - Casos de uso: Enrique Ballesteros, Jaime Dan Porras, Ignacio Iker Prado.
 - Especificación de Requisitos: Jesús Aguirre y Alejandro Villarín.
 - Plan de proyecto: Jesús Aguirre y Enrique Ballesteros.
 - Riesgos: Jaime Dan Porras y Alejandro Villarín.
 - Estimación: Ignacio Iker Prado.

Parte IX

Herramientas, técnicas y metodologías

- Hemos llevado a cabo el proceso de revisiones software para la Garantía de Calidad Software. En particular, nos hemos centrado en las Revisiones Técnicas Formales RTFs o Inspecciones Formales. El objetivo de las Revisiones Técnicas Formales es detectar errores antes de que se conviertan en defectos y garantizar y verificar que el desarrollo del proyecto sigue el camino correcto y es coherente con el resto del proyecto.

Parte X

Control de medios

- A partir de lo establecido en la especificación de requisitos debemos buscar las plataformas que mejor se adaptan a nuestro software. Así, mientras que los datos se deben almacenar en discos duros y sistemas de almacenamiento de datos, la mayor parte del uso de la aplicación se desarrolla desde otros terminales.
- Para garantizar que estos medios físicos son los óptimos para nuestros productos software debemos realizar revisiones ocasionales de los avances que se producen en el desarrollo de nuevos productos y de cómo esos avances pueden sernos útiles. También debemos modificar los medios que vamos a utilizar cuando se produzca un cambio en los casos de uso o en la especificación de requisitos, por leve que sea, pues estos cambios pueden tener importantes repercusiones en los medios físicos que van a ser mejores para nuestra aplicación.

Parte XI

Control de proveedor

De momento el único software que será proporcionado por proveedores externos serán las bases de datos contra las que trabaja KIKE- Hostelería ®. Dado que el proyecto no se ha materializado, no hemos tenido la necesidad de subcontratar a una empresa para la programación de las susodichas bases de datos.

Parte XII

Colección de registros, mantenimiento y conservación

Como mediante la Garantía de Calidad software se mejora la documentación del proyecto, ésta debe mantenerse dentro de la propia documentación, pues será de gran utilidad una vez acabado el proceso si se desea comprobar la fiabilidad y pruebas realizadas al software. En nuestro caso, esto cobra especial relevancia en el campo de la seguridad del programa, puesto que lo único que diferencia a los distintos tipos de empleados es el nombre de usuario y la contraseña. Por tanto, es vital aseverar que la seguridad de KIKE- Hostelería ® es total, y las pruebas que lo demuestran deben salvaguardarse una vez acabado el proceso.

El proyecto, al trabajar on la herramienta Google Code, está almacenado en los servidores de Google Inc., aunque cada miembro de P_KT posee una copia en su ordenador para prevenir un posible fallo de los servidores anteriormente mencionados. Una vez acabado el proceso, se elaborará un documento que contenga toda la información relativa a este, incluyendo todos los documentos que se han producido durante la realización del proyecto.

Los registros de cambios en la documentación se conservarán como testimonio del avance gradual del proyecto, permitiendo a su lector conocer qué pasos se siguieron para el refinamiento de los documentos englobados en la entrega final.

Parte XIII

Formación

La formación que han recibido los integrantes del grupo de desarrollo para satisfacer las necesidades del plan SQA, es la de la asignatura Ingeniería del Software de la Universidad Complutense de Madrid, impartida por Gonzalo Méndez. En ella se les ha enseñado el manejo de varias herramientas de ingeniería del software, como ArgoUML, Microsoft Project o COCOMO II. También han sido instruidos acerca de la realización de un desarrollo de software que se ajuste a los estándares más utilizados, y cómo desarrollar un software de calidad que se ajuste a los requisitos y que satisfaga al cliente.

Como complemento han recibido clases de programación orientada a objetos en Java, lo que permitirá el desarrollo de la aplicación en este lenguaje. (- Identifica las actividades de formación necesarias para satisfacer las necesidades del plan SQA.)

Parte XIV

Gestión del riesgo

Está hecho en un documento aparte.

Parte XV

Procedimiento de cambio e historia del plan SQA

(- Procedimientos de modificación del plan SQA. - Procedimientos de mantenimiento del historial de cambios. - Historial de cambios.)

