INFORME DE AUDITORÍA DEL SISTEMA DE GESTIÓN CARNICERÍAS RIESCO

Trabajo realizado por Manuel Guerrero Moñús

Auditoría, Calidad y Fiabilidad del Software.

UCM, Máster en Ingeniería Informática 2020 – 2021.

Destinatario del informe

“Adrián Riesco”, como cliente, presidente y propietario de la empresa “Carnicerías Riesco”.

Empresa auditada

“CholloSoft”, empresa dedicada la programación e integración de sistemas de software.

Proyecto auditado

El proyecto que se audita es “Sistema de gestión Carnicería Riesco”, un software dedicado a la administración y las ventas a domicilio de los productos que ofrece la Carnicería Riesco.

Aspectos evaluados

1. Estudio de adecuación: análisis y elección de las tecnologías y los protocolos.
2. Plan de desarrollo: definición del plan y tareas críticas; hitos y riesgos del proyecto.
3. Requisitos: análisis y cambios; mantenibilidad y extensibilidad del proyecto.
4. Diseño: revisiones realizadas sobre los diseños elaborados.
5. Implementación: uso de patrones, documentación del código, testing y revisiones.
6. Seguimiento: consecución de los hitos, supervisión y resolución de los riesgos del proyecto.
7. Documentación: control de la documentación y documentos controlados.
8. Métricas: cálculo y uso de métricas en el proyecto.
9. Contrato: detección de ambigüedades y análisis de los factores de calidad del software.
10. Estándares: consideración y uso de estándares para la obtención de calidad.

Comparabilidad

No se disponen de informes de auditoría anteriores con los que poder comparar.

Salvedades

- No se ha podido contactar con los desarrolladores ni con el cliente a la finalización del proyecto, por lo que no ha podido aclararse la instalación del producto y algunas cuestiones relacionadas con el uso de la aplicación detalladas en el manual de usuario.
- Falta la documentación correspondiente a la elección de los protocolos, la realización de las revisiones y el seguimiento de los riesgos.
- Faltan los ficheros con el código funcional del testing.

Incumplimientos

#ID	Objetivo	Sí	No	Parcial	No Aplica
Estudio de adecuación					
1.1	Estudio detallado de tecnologías a utilizar				X
1.2	La elección se tomó de manera adecuada	X			
1.3	Proceso documentado				X
1.4	Estudio detallado de los protocolos disponibles		X		
1.5	La elección se tomó de manera adecuada		X		
1.6	Proceso documentado		X		
Plan de desarrollo					
2.1	Plan de desarrollo definido	X			
2.2	Plan de desarrollo documentado			X	
2.3	Tareas críticas establecidas		X		
2.4	Hitos establecidos	X			
2.5	Riesgos establecidos	X			
Requisitos					
3.1	Requisitos claramente documentados	X			
3.2	Requisitos claramente definidos			X	
3.3	Existen requisitos de mantenibilidad	X			
3.4	Existen requisitos de extensibilidad	X			
3.5	Cambios realizados de forma controlada				X
Diseño					
4.1	Diseño claramente documentado	X			
4.2	Revisiones formales de diseño		X		
4.3	Revisiones por pares				X
4.4	Revisiones por expertos		X		
Implementación					
5.1	Pruebas de unidad		X		
5.2	Testing documentado	X			
5.3	Revisiones por pares		X		
5.4	Uso de patrones	X			
5.5	Código adecuadamente documentado		X		
Seguimiento					
6.1	Hitos seguidos			X	
6.2	Riesgos seguidos		X		
6.3	Riesgos solucionados		X		

Documentación					
7.1	Documentos controlados		X		
7.2	Responsable de documentos controlados		X		
7.3	Documentación para testing			X	
7.4	Documentación de instalación y uso			X	
7.5	Documentación para futuras extensiones				X
Métricas					
8.1	Se calculan métricas		X		
8.2	Se usan métricas		X		
Contrato					
9.1	Se evitan ambigüedades		X		
9.2	Se tienen en cuenta todos los factores de calidad			X	
Estándares					
10.1	Se consideran los estándares necesarios			X	
10.2	Se siguen los estándares necesarios			X	

1.1) Estudio detallado de las tecnologías a utilizar.

Se ha revisado la documentación entregada y no se ha podido encontrar en ningún documento un análisis detallado de las tecnologías que podrían resultar más convenientes para llevar a cabo la realización de este proyecto.

Sin embargo, se ha considerado que, debido a que el modelo del negocio de la aplicación es muy clásico, no se va a considerar necesario la realización de un estudio detallado para la elección de las tecnologías, por lo tanto, este punto no aplica.

1.2) La elección se tomó de manera adecuada.

Teniendo en cuenta el contenido del fichero “Análisis y diseño funcional”, podemos observar en su apartado 4.6 “Entorno de la aplicación”, en la página 30, cuáles han sido las herramientas escogidas para la construcción del producto software.

Se ha escogido “Eclipse Luna”, el framework web “Symfony” y los paquetes de software del “Sonata Project” y “FOSBundle” para el desarrollo del sistema de venta online del cliente. El servidor XAMPP para la ejecución del código PHP y el sistema gestor de bases de datos relacional MySQL. GitHub como sistema de alojamiento del código en la nube para poder ser utilizado con el sistema de control de versiones Git, según se especifica en el apartado 6.1 “Repositorios de versiones” de este mismo documento, en la página 30. También se ha utilizado Google Drive como almacén en la nube para albergar la documentación del proyecto, según se especifica en el apartado “Herramientas para el desarrollo” del documento “Presentación final”, en la página 3.

Después de tener en cuenta las herramientas anteriormente escogidas, se considera que la elección de éstas es correcta debido a que el framework “Symfony” viene preparado para poder construir aplicaciones haciendo uso de una arquitectura MVC, permite el desarrollo de software en base al uso de patrones y la realización de baterías de pruebas, por lo que permite llevar a cabo un desarrollo modular y sostenible. La base de datos MySQL es adecuada, pues el modelo de negocio es clásico y responde ante un modelo de información estructurada, por lo que servirá como almacén de datos de la aplicación. GitHub y Git son conocidos por la mayoría de los desarrolladores para el control del código fuente del proyecto por lo que serán suficientes para esta tarea. Finalmente, en lo que respecta a Google Drive, también se considera adecuado en esta ocasión, ya que la cantidad de documentación generada en el proyecto no es mucha, por lo que puede ser controlada manualmente de forma responsable.

1.3) Proceso documentado.

No hay documentación respecto al estudio de las tecnologías a utilizar, ni tampoco respecto a la elección de éstas; sin embargo, atendiendo a las conclusiones recogidas en la presente auditoría en los puntos 1.1 “Estudio detallado de las tecnologías a utilizar” y 1.2 “La elección se tomó de forma adecuada”, se considera que no aplica la realización de esta documentación.

1.4) Estudio detallado de los protocolos disponibles.

Se ha revisado toda la documentación de la que se dispone para el proyecto y no se ha encontrado ninguna información que haga referencia a los protocolos de comunicación o seguridad disponibles para su uso en la aplicación desarrollada, tampoco se hace referencia a la seguridad con las que la aplicación almacena las contraseñas de usuario.

Se considera, por tanto, que el sistema desarrollado no cumple con este aspecto, debido a una falta de compromiso por parte de la empresa desarrolladora con el análisis y la especificación de los requisitos no funcionales relacionados con la seguridad, que debeían haberse tenido en cuenta.

Recomendaciones:

- Analizar, añadir y cumplimentar nuevos requisitos no funcionales que hagan referencia a la seguridad de las comunicaciones y del sistema software en general, como una forma de comprometerse con el cliente y garantizarle así la confidencialidad con la que la aplicación gestiona la información de su negocio.

1.5) La elección se tomó de manera adecuada.

En cuanto a la seguridad de las comunicaciones, se ha encontrado en el apartado 5.5.1. “Definición del modelo de datos” del documento “Análisis y diseño funcional”, en las páginas 46 – 48, una evidencia en las tablas del modelo del dominio sobre el protocolo de comunicaciones que se ha empleado, este es http, que no https. Esto se ha comprobado mirando los ficheros de código fuente php de la aplicación bajo la ruta /*Carniceria/src*, donde solo hay peticiones de tipo http.

Se concluye pues, que este punto no se ha cumplido, ya que prácticamente se desconocen las decisiones tomadas en estos aspectos y las evidencias de las tomadas, indican malas decisiones.

Recomendaciones:

- Cambiar el protocolo http por https u otros protocolos que permitan el cifrado de los datos para evitar que las peticiones de la aplicación viajen en texto plano, evitando de esta forma que alguna persona pueda interceptar el tráfico de red para hacerse con los datos de los usuarios.

1.6) Proceso documentado.

El estudio de los protocolos disponibles y su proceso de elección no está recogido en ninguno de los documentos de los que se dispone.

Recomendaciones:

- Esclarecer las decisiones tomadas en lo referente la seguridad que utiliza la aplicación para gestionar las contraseñas almacenadas, así como otras medidas de seguridad que se hayan podido tomar, pues dicha documentación es necesaria para tener un mayor control sobre las actividades realizadas en el proyecto y favorecer la mantenibilidad de la documentación del mismo.

2.1) Plan de desarrollo definido.

El Plan de desarrollo está definido dentro del documento “Plan de desarrollo”, en el apartado 4.2.1 “Plan de las fases”, página 9. Allí se exponen de forma detallada las fases de las que se compone el proyecto de desarrollo, la duración de cada una de ellas y el número de iteraciones a realizarse en cada una.

2.2) Plan de desarrollo documentado.

El Plan de desarrollo está documentado dentro del documento “Plan de desarrollo”, en el apartado 4.2.2 “Calendario del proyecto”, páginas 10 - 11. En ellas se explica como será la generación de artefactos en cada fase y un diagrama de Gantt que muestra cómo es la secuenciación en el tiempo de las actividades que componen cada etapa del proyecto de desarrollo.

Recomendaciones:

- Sería adecuado copiar el contenido del apartado 1.2 “Metodología” del documento “Análisis funcional”, página 3, en el documento “Plan de desarrollo” ya que en este último se indica que la metodología es ágil, pero no se matiza cuál se utiliza particularmente (Scrum).

- Añadir información que complemente el diagrama Gantt. Concretamente, la relacionada con la estimación de recursos económicos, de personal y tiempo necesario por persona (esfuerzo) para realizar cada una de las actividades de cada etapa. Para realizar esta tarea se podría añadir información adicional junto al diagrama Gantt o crear una única tabla que combine la información de la estructura de descomposición de tareas del proyecto y la planificación temporal.

2.3) Tareas críticas establecidas

No se ha podido encontrar información en lo referente a las tareas críticas del proyecto en el documento “Plan de desarrollo”, tampoco en ningún otro documento del proyecto.

Recomendaciones:

- Es necesario ser consciente de todas aquellas tareas cuyo retraso supone el retraso de todo el proyecto de desarrollo. Se recomienda pues utilizar algún tipo de análisis para encontrarlas, por ejemplo la elaboración de un grafo PERT, pues éste no solamente permite hallar el camino crítico compuesto por las actividades que no pueden retrasarse, sino que también nos sirve para estimar el tiempo que se va a tardar para finalizar el proyecto o la varianza de la duración de las actividades.

2.4) Hitos establecidos.

Los hitos se encuentran correctamente definidos dentro del documento “Plan de desarrollo” en el apartado 4.2.1 “Plan de las Fases”, página 9. En el se detallan los objetivos que se esperan lograr al acabar cada etapa del proyecto de desarrollo.

2.5) Riesgos establecidos.

Los riesgos del proyecto están recogidos en el documento “Análisis y matriz de riesgo”, apartados 3 “Estrategia de Manejo de Riesgos” y 4 “Respuesta”, páginas 4 - 10.

3.1) Requisitos claramente documentados.

Los requisitos se encuentran documentados en el documento “Especificación de requerimientos de software”. Concretamente en el apartado 5 “Requisitos”, páginas 6 - 8.

3.2) Requisitos claramente definidos.

Los requisitos del proyecto están definidos en los subapartados del apartado 5 “Requisitos”, estos son: 5.1 “Actores” y 5.2 “Requisitos funcionales”, 5.3 “Requisitos no funcionales”, 5.4 “Requisitos Tecnológicos” y 5.5 “Requisitos de desarrollo”. Páginas 6 - 8.

Los requisitos de los apartados 5.3 “Requisitos no funcionales”, 5.4 “Requisitos Tecnológicos” y 5.5 “Requisitos de desarrollo” han sido definidos correctamente.

Recomendaciones:

- Los requisitos funcionales indican qué debe hacer el sistema que se pretende crear, pero de una forma excesivamente breve. No aportan información básica como: la prioridad del requisito, la probabilidad de que el requisito cambie, qué datos de entrada tiene, los datos de salida que emitirá, precondición, postcondición, interfaz de usuario donde se utiliza la funcionalidad, usuarios que pueden utilizar dicha funcionalidad, errores que pueden ocurrir, etc.

- Los apartados 5.1 “Actores”, 5.4 “Requisitos Tecnológicos” y 5.5 “Requisitos de desarrollo” se podrían unificar bajo un único apartado denominado “Requisitos del dominio del software” ya que todos ellos vienen dados por las características del contexto en el que se implanta el software.

3.3) Existen requisitos de mantenibilidad.

Lo referente al mantenimiento del producto software puede encontrarse en el documento “Contrato de desarrollo de software”, en el punto VII “Garantía y Soporte”, del apartado “Pactos y estipulaciones”, página 7.

En esta sección se estipula que cada vez que se entregue y valide una parte del programa, se inicia un periodo de garantía del correcto funcionamiento y adecuación a los requisitos de rendimiento y calidad de 6 días laborales continuos, durante los cuales habrá un servicio de mantenimiento de tipo correctivo.

3.4) Requisitos de extensibilidad.

La empresa desarrolladora no realizará ningún mantenimiento del producto que sea de tipo adaptativo y/o perfectivo, según el punto VII “Garantía y Soporte”, del apartado “Pactos y estipulaciones”, página 7.

3.5) Cambios realizados de forma controlada.

En el apartado 1.2 “Alcance”, del documento “Especificación de requerimientos de software”, en la página 3, se indica que el alcance de la especificación solo abarca las etapas del desarrollo de la misma y que no incluirá ningún cambio de requerimientos o evolutivo de la aplicación. Solo se tendrá en cuenta la información suministrada las primeras reuniones con el cliente y las minutas de las mismas, por lo que se considera que no aplica tener un registro para el control de los cambios realizados en los requisitos, pues tampoco se han encontrado evidencias de que el cliente quisiera realizar cambios en los mismos, según lo especificado en el documento “Contrato de desarrollo” en la estipulación IX “General”, apartado “Nulidad o anulabilidad”, página 9.

4.1) Diseño claramente documentado.

Los diseños de la aplicación y su documentación asociada se encuentran dentro del apartado 5 “Diseño Técnico” del documento “Análisis y diseño funcional”, páginas 31 – 47. Concretamente los diseños de los que se dispone son: diseño conceptual de la web, diseños de las vistas de usuario, de los formularios, los mensajes de error y el modelo de datos.

4.2) Revisiones formales de diseño.

Según lo especificado en el apartado 5 “Revisiones” del documento “Plan de calidad”, página 12, el objetivo de dicha sección es definir las revisiones que se realizarán en el proyecto y especificar cómo realizarán.

En el apartado 5.2.4 “Revisiones de diseño”, página 13, en el mismo documento se especifica que se habrán de llevar a cabo estas reuniones para asegurar la consistencia del diseño detallado con la especificación de requerimientos. Sin embargo, no se ha encontrado ninguna información entre los documentos del proyecto que haga referencia a la realización de dichas reuniones.

Recomendaciones:

- Realizar revisiones formales internas al acabar la actividad diseño del proceso de desarrollo de software del proyecto para comprobar que los diseños realizados tienen suficiente calidad como para construir un software que dé soporte a las especificaciones dadas por el cliente.

4.3) Revisiones por pares.

No se ha encontrado información en cuanto a la realización de revisiones por pares, sin embargo, se considerará suficiente con realizar una revisión interna de tipo formal por parte de todo el equipo de desarrollo, tal y como se especificó en el apartado 5.2.4 “Revisiones de diseño”, en el documento “Plan de calidad”, en la página 13, con el objetivo de asegurar la consistencia del diseño detallado en la especificación de requisitos. Por lo que este punto, no aplica.

4.4) Revisiones por expertos.

No se ha encontrado información que haga referencia a la realización de revisiones por expertos en lo referente al diseño, sin embargo, éstas hubieran sido convenientes.

Recomendaciones:

- Realizar revisiones por expertos que sirvan como una orientación y/o corrección de algunos aspectos críticos de la aplicación, en los que tal vez no se tiene mucha experiencia o que simplemente son difíciles de gestionar. Por ejemplo:

    - Elaborar diseños que incluyan patrones de software o bien módulos para implementar controles orientados a la seguridad y así evitar ataques comunes como SQL Injection.

    - Elaborar diseños que permitan indicar conceptualmente cómo han de realizarse las comunicaciones de forma segura, mediante el uso de protocolos de comunicación que sean seguros, o bien delegando la seguridad en sistemas de terceros, en cuyo caso se ha de detallar la comunicación con sus interfaces externas.

5.1) Pruebas de unidad.

Se han realizado varias búsquedas en la carpeta “Carnicería”, que contiene el código PHP, que ha sido desarrollado con el framework de Symfony para luego ser ejecutado finalmente en el servidor XAMPP. Pudiera parecer que las pruebas desarrolladas se corresponden con las de los ficheros en la ruta /Carniceria/src/Chollosoft/RiescoBundle/Tests/Controller, pues éstos hacen referencia a la funcionalidad que el sistema debe poder realizar. Sin embargo, éstos no contienen los test unitarios y están en su amplia mayoría comentados e inacabados, por tanto, no se ha encontrado la información pertinente a la implementación de las pruebas unitarias del proyecto de software.

Recomendaciones:

- Esclarecer la situación de las pruebas unitarias, ya que existe información acerca de las los resultados obtenidos de su ejecución en el documento “Matriz de pruebas”, apartado 1 “Matriz de pruebas”, páginas 1 – 9, pero no se tienen sus ficheros para verificar que realmente el equipo de desarrollo ha realizado las actividades estipuladas para poder completar la tercera y última fase del proyecto.

- En caso de que éstas no estén realizadas, llevar a cabo su implementación para poder automatizar la comprobación de las funciones de los distintos módulos del sistema software. De esta forma, se evitará que el resultado final de lo implementado dependa directamente de las habilidades de los programadores, ya que éstos podrían accidentalmente pasar por alto ciertas casuísticas del comportamiento del software que se deberían controlar, lo que lleva a la aparición de errores durante la fase de desarrollo que son propagados al producto final, que es recibido por el cliente.

5.2) Testing documentado.

La documentación relativa al testing realizado en el proyecto de desarrollo se encuentra dentro del documento “Matriz de pruebas”, en el apartado 1 “Matriz de pruebas”, páginas 3 – 9. Por lo tanto, se considera cumplimentado este punto.

5.3) Revisiones por pares.

No hay ninguna información en lo referente a la realización de revisiones por pares para analizar las implementaciones desarrolladas.

Recomendaciones:

- Planificar revisiones por pares en la etapa de desarrollo y testing con el objetivo de que los desarrolladores analicen lo implementado desde diferentes puntos de vista, ello permitirá encontrar fallos tanto en la funcionalidad codificada así como comprobar, de una forma más exhaustiva, si se han cubierto todas las casuísticas de las funcionalidades sometidas a tests.

5.4) Uso de patrones.

La aplicación web ha sido desarrollada con el framework web de Symfony, por lo que esta ha sido creada siguiendo un patrón arquitectónico de tipo Modelo Vista Controlador. Por otra parte, este framework también implementa varios patrones Controller para la recepción y gestión de aquellas peticiones que recibe la aplicación. También usa clases Entity a modo de DAO para la persistencia.

Debido a que el modelo del negocio es demasiado clásico, se considera que no serán necesarios más patrones de software por el momento para lograr la mantenibilidad de la aplicación.

5.5) Código adecuadamente documentado.

Se ha revisado el código implementado en la aplicación, disponible en la ruta /Carniceria/src y se ha revisado el estado de los comentarios de los ficheros de código fuente. El código observado en los ficheros apenas tiene comentarios y los pocos que existen no hace que sea más intuitivo saber lo que el código hace. Se considera pues que la documentación realizada no es adecuada.

Recomendaciones:

- Elaborar la documentación del código fuente programado con el objetivo de favorecer que la aplicación desarrollada pueda ser mantenible a lo largo del tiempo, así como permitir a otros desarrolladores entender de forma sencilla qué es lo que se ha hecho, o bien con fines de aprendizaje o bien con fines de realizar un mantenimiento correctivo.

6.1) Hitos seguidos.

El primer hito se ha cumplido, puesto que los requisitos del producto se recogen en el documento "Especificación de requerimientos de software", apartado 5 "Requisitos", páginas 6 – 8; el plan de desarrollo del proyecto está definido a lo largo del documento "Plan de desarrollo" y los principales casos de uso se identifican en el documento "Análisis y diseño funcional", apartado 4.2. "Diagrama de Casos de uso", página 6.

Se desconoce si se cumplió el segundo hito. Se realizó la entrega del prototipo pero no se ha podido comprobar su funcionalidad debido a la complejidad de su instalación, tal y como se indica en el apartado 7.4 de la presente auditoría. Tampoco hay constancia de que el cliente haya realizado las verificaciones sobre el software en esta entrega.

El tercer hito no se ha cumplido, pues según lo especificado en el apartado 4.2.1 "Plan de las Fases" del documento "Plan de desarrollo", página 9, el hito que marca el fin de la fase de pruebas es la ejecución de todas las pruebas de forma exitosa en la Matriz de pruebas del documento "Matriz de pruebas", páginas 3 – 9. Allí hay varias pruebas que han quedado registradas como fallidas, de éstas, todas llegaron a ser solventadas a excepción de la correspondiente al caso de prueba 12, identificado con el nombre "Aceptar o Rechazar Pedido".

En dicho caso de prueba, se indica que no se cumple positivamente el paso 12.2, lo que supone que no se puede cumplir la condición de éxito indicada en el apartado 4.3.12. "Aceptar pedido" del documento "Análisis y diseño funcional", página 18, la cual indica que: "Las cantidades de los productos del pedido disponibles en el almacén se ven modificadas", mientras que el resultado de la prueba indica: "No se modifican coherentemente algunos valores".

Teniendo en cuenta lo anterior, se considera que los hitos se cumplen parcialmente.

Recomendaciones:

- Se debe incrementar el compromiso del equipo de desarrollo con las revisiones establecidas en el documento “Plan de calidad”, concretamente con lo que se indica en el apartado 3.4.2.1 “Revisar cada entregable generado”, página 8. Allí se dice que ha de verificarse que no queden correcciones sin resolver en los informes de revisión previos, de forma que si se encuentra alguna no resuelta, esta debe ser incluida en la siguiente revisión.

- Ya que el equipo trabaja con Scrum, sería conveniente generar y gestionar un documento “Impediment Backlog” en el que se especifique cuales han sido aquellos motivos que han generado problemas en cada momento del proceso de desarrollo. Este sería utilizado por los responsables del Plan de desarrollo del proyecto para analizar y controlar los problemas en curso, también para reajustar la forma en que se realizan las actividades del proyecto cuando éstas no están dando resultado.

6.2) Riesgos seguidos.

Según se ha especificado en el apartado 4. “Respuesta”, en el documento “Análisis y matriz de riesgo”, página 10, cada una de las respuestas a los riesgos tiene asociado un responsable de ejecución y de seguimiento, además se ha comprobado que los riesgos que tienen un responsable de seguimiento son aquellos que se han decidido controlar, dado su nivel de peligrosidad.

Pese a que el seguimiento de los riesgos fue planificado, no hay ninguna documentación elaborada en lo que respecta a éstos para saber si se han podido controlar sobre la marcha, aplicando planes de mitigación o si bien impactaron en el proyecto y hubo de aplicarse algún plan de contingencia para resolverlo. Se va a considerar, por lo tanto, que no ha habido un seguimiento de los riesgos.

Recomendaciones:

- La empresa desarrolladora debe realizar el seguimiento de los riesgos para identificar en qué momentos del proyecto se dan situaciones que, a la larga, terminan convirtiéndose en algún evento que implica un retraso en la planificación del proyecto o el incumplimiento de lo pactado con el cliente, o ambas cosas, lo cual terminaría acarreando muchas consecuencias adversas. Es por eso por lo que se han de seguir y aplicar sus respectivos planes de mitigación, para alejarse todo lo posible de las circunstancias que terminan llevando el proyecto a una situación de desastre.

6.3) Riesgos solucionados.

No se ha encontrado ningún documento del presente proyecto en el que se informe sobre cuáles de los riesgos analizados en el documento “Análisis y matriz de riesgo” tuvieron lugar y más tarde fueron solucionados. Al no poseerse ninguna información, se considera incumplido este punto.

Recomendaciones:

- La elaboración de documentación relativa al modo en que se solucionaron los riesgos es de vital importancia para la empresa desarrolladora, pues en el futuro será de utilidad para disponer de información histórica que puede ser utilizada para mejorar la eficiencia de los planes de contingencia y así resolver más rápidamente los riesgos que tengan impacto en el proyecto.

7.1) Documentos controlados.

En el documento “Plan de calidad” se estipula en el punto 7 “Estándares de documentación”, en la página 14, cómo la empresa desarrolladora se compromete a seguir un estándar de documentación con el objetivo de garantizar la unicidad de los formatos y estándares de documentación de los documentos para su fácil lectura y revisión.

Los ficheros deben incluir las siguientes cabeceras:

- Información del documento: Nombre del documento, versión del documento, fecha de la versión, elaborado por, revisado por.

- Historial de la versión: Versión, fecha, actualizado por, modificación realizada.

Por una parte, se han revisado los correspondientes documentos del proyecto: “Análisis funcional”, “Análisis y matriz de riesgo”, “Contrato de desarrollo”, “Especificación de requerimientos de software”, “Matriz de pruebas”, “Plan de calidad”, “Plan de desarrollo” y “Manual de usuario”. Todos cumplen lo especificado en el Plan de calidad a excepción del manual de usuario, que no posee la cabecera de “Historial de versión” ni la de “Información del documento”. Tampoco cumple el documento de “Contrato de desarrollo”, ya que no posee la cabecera de “Información del documento” y no llama a la cabecera “Historial de versión” de esa manera, de forma que ahí no se cumplió con la unicidad del formato.

Por otra parte, en el documento “Plan de calidad” se especifica en el apartado 3 “Gestión”, página 5, quiénes son los responsables de la elaboración de cada uno de los entregables generados como consecuencia del proceso de desarrollo del proyecto de software. Se ha comprobado que los responsables de su elaboración coincidan con las personas que se anotaron en el campo del control de versiones “Elaborado por” de cada documento, para comprobar si se ha hecho un uso correcto del control de cambios manual, pero se han encontrado discrepancias entre el responsable designado y el figurante, al menos en los siguientes documentos: “Matriz de pruebas”, “Plan de desarrollo”, “Plan de calidad”, “Contrato de proyecto” y “Especificación de requerimientos del software”. Este análisis también ha permitido averiguar que el documento “Análisis y matriz de riesgo” no tiene asociada una persona responsable en el plan de calidad, sin embargo, en la cabecera de control de versiones de ese documento, sí figura un responsable de su elaboración.

Por último, el plan de calidad tampoco indica explícitamente si el responsable de elaboración también es el mismo que el responsable de supervisión que debe de figurar como responsable de la supervisión del correspondiente documento, pero en caso de ser así, tampoco se ha cumplido ya que se han encontrado discrepancias en los siguientes documentos: “Especificación de requerimientos del software”, “Contrato de desarrollo”, “Plan de desarrollo”, “Análisis y diseño funcional” y “Matriz de pruebas”.

Teniendo en cuenta todos los factores anteriores, se considera que los documentos no han sido controlados correctamente.

Recomendaciones:

- Utilizar un software para el control de la documentación en lugar de realizar el control de los documentos manualmente. Esto permitirá establecer controles más rigurosos en lo referente a los permisos de acceso y a la realización de cambios, así como realizar copias de seguridad y recuperar versiones particulares de los documentos.

7.2) Responsable de documentos controlados.

No se ha especificado nada en lo que respecta a un responsable del control de la documentación en el documento “Plan de calidad”, solamente se han designado responsables de la elaboración de ciertos documentos entregables. Tampoco se ha encontrado información al respecto en el resto de la documentación.

Recomendaciones:

- Nombrar a un responsable de la documentación del proyecto que se ocupe de definir la lista de los documentos que han de ser controlados, especificar como se preparará y aprobará la documentación, decidir dónde se realizará el almacenamiento de ésta y establecer los permisos de cara a su acceso y recuperación.

7.3) Documentación para testing.

La documentación para el testing se encuentra dentro del documento “Matriz de pruebas”, en el apartado 1 “Matriz de pruebas”, páginas 3 – 9. En este documento se exponen las pruebas unitarias realizadas sobre el sistema utilizando la especificación de los casos de uso que han sido recogidos en el documento “Análisis y Diseño”.

La información examinada en el documento es confusa, pues el testing de casos de uso está basado en la comprobación de eventos, por lo tanto este tipo de pruebas serían de caja negra, es decir, sobre el propio sistema, pero sin atender a los detalles de la implementación.

Por otra parte, las pruebas unitarias son pruebas de caja blanca, es decir, aquellas que están relacionadas con el correcto funcionamiento de las funciones individuales que componen cada uno de los módulos de un sistema software. Estas serán responsables de evaluar a un nivel muy fino que cada función cubre todas las posibilidades para las cuales ha sido pensada, y atienden a lo especificado en los requisitos de tipo no funcional para cumplir con los requerimientos de tiempo, disponibilidad, seguridad, etc.

Por la ambigüedad de lo documentado, se considerará que la documentación se ha realizado de forma parcial, pues son necesarias algunas aclaraciones.

Recomendaciones:

- Esclarecer las acciones que son realizadas para evaluar las pruebas, de esta forma se podrá saber claramente qué tipo de testing se ha realizado así como qué y cómo se ha evaluado.

- Separar las pruebas realizadas en función de si son de caja blanca o de caja negra.

    - Las pruebas de caja negra deben recoger como poco la siguiente información: identificador del requisito funcional, nombre del requisito funcional, módulo al que éste pertenece, última fecha en la que se comprobó su funcionamiento, entrada a probar, salida esperada, salida originada y un comentario optativo para esclarecer la salida.

    - Para las pruebas de caja blanca, sería necesario llevar un registro de qué pruebas se están realizando en cada fichero de código asociado al testing, una descripción de lo que se pretende probar en cada una de las pruebas realizadas y los resultados obtenidos al ser ejecutadas por última vez, añadiendo dicha fecha. A nivel de código, es necesario que éstas sean bien documentadas, explicando todas las casuísticas que comprueban y los controles que realizan para cumplir los requisitos no funcionales que correspondan.

7.4) Documentación de instalación y uso.

La información de la instalación se encuentra disponible dentro del documento “Análisis y diseño funcional”, en el apartado 8 “Instalación”, páginas 49 - 50. Allí se explica cómo puede instalarse y configurarse el servidor, así como la base de datos.

También se dispone de la versión definitiva del documento “Manual de usuario”, en el que queda explicado el funcionamiento de la aplicación web que conforma sistema de venta de productos cárnicos de la empresa cliente.

Recomendaciones:

- La documentación de la instalación tiene suficiente entidad como para formar un documento controlado aparte, al igual que el de manual de usuario, pues recoge información de utilidad para el proyecto.

- La información de la instalación debe aumentar mucho más su nivel de detalle, pues no es lo suficientemente comprensible como para poder llevarla a cabo. No se indica donde está el fichero de configuración del servidor apache que se ha de modificar, tampoco especifica nada de la configuración del módulo “RewriteMod” ni detalla como ejecutar el script que carga la base de datos.

- Habrían de aclararse en el documento “Manual de usuario” algunas cuestiones:

    1. En el apartado 3.3.2. “Enviar solicitud”, se dice que después del envío del pedido se genera un mensaje en el cual se indica el código de identificación de la compra, con el cual el cliente puede, si así lo desea, ir al local más cercano para su recogida, pero no se especifica si la generación de este código es posterior a la aceptación del pedido por parte de un usuario administrador, tampoco como se hace para concretar que se desea ir al comercio a pagar el pedido en lugar de esperar a la entrega a domicilio.

    2. ¿Cómo se entera un usuario de si le han cancelado el pedido por alguna razón?

7.5) Documentación para futuras extensiones.

No se ha podido encontrar documentación adicional que haga referencia a futuras extensiones del proyecto de software. Sin embargo, este aspecto no se considerará aplicable debido a un par de consideraciones:

1. En el documento “Contrato de desarrollo” no se especifica la realización de un mantenimiento perfectivo para ampliar la funcionalidad del software y tampoco hay un contrato anexo que lo indique.

2. En el apartado 1.2 “Alcance”, del documento “Especificación de requerimientos de software”, en la página 3, se indica que el alcance de la especificación solo abarca las etapas del desarrollo de la misma y que no incluirá ningún cambio de requerimientos o evolutivo de la aplicación.

Por lo tanto, se entiende que el proyecto ha sido pactado de mutuo acuerdo entre la empresa desarrolladora y la empresa cliente como un único encargo, por lo que solamente ha de elaborarse la documentación correspondiente a aquello a lo que la empresa desarrolladora se ha comprometido a desarrollar para la empresa cliente.

8.1) Se calculan métricas.

Aunque en el apartado 3.2.3. “Responsabilidad del equipo de calidad documento”, del documento “Plan de calidad”, en la página 6, se establece que una de las actividades del equipo para la garantía de la calidad es el análisis de métricas, pero no se ha encontrado ninguna métrica calculada dentro de los documentos entregados, por lo que se considera que no se cumple este punto.

Recomendaciones:

- El equipo para la garantía de la calidad del proyecto debería calcular al menos ciertas métricas básicas, estas servirían para comprobar si el proyecto se está desviando de sus objetivos por motivos económicos, de personal o tiempo. También servirían para saber si la calidad del producto durante las fases de desarrollo, testing y mantenimiento correctivo, es la adecuada, o si bien han de aplicarse ciertas acciones correctivas de cara a lo desarrollado.

8.2) Se usan métricas.

Al no calcularse métricas, éstas tampoco se han utilizado.

Recomendaciones:

- En lo que respecta a métricas del proceso de desarrollo, al menos podría utilizarse el esfuerzo en personas/hora para estimar cada actividad del proyecto de software. Esto permitiría generar información histórica que puede ser utilizada para mejorar la gestión de estos recursos en procesos de desarrollo futuros o para realizar otras estimaciones por analogía.

- En lo que respecta a métricas del producto, al menos podrían utilizarse la cantidad de errores solucionados por fase, la cantidad de fallos detectados por entregable, personas/hora para el arreglo de los fallos detectados, etc.

9.1) Se evitan ambigüedades.

Ambigüedades detectadas en el documento “Contrato de desarrollo”:

- En la estipulación III “Ciclo de desarrollo”, en la página 4, se indica que la descripción de las funcionalidades y requisitos que debe cumplir cada entrega parcial del producto las acordarán las partes en contratos anexos que no aparecen.

- En la estipulación VII “Garantía y soporte”, en la página 7, no se explica con claridad el tiempo de respuesta ante una incidencia y el tiempo de reparación. El tiempo de respuesta ante una incidencia tiene que ser inferior a un intervalo de tiempo que comienza con la notificación de la incidencia, pero no se sabe cuando termina. El tiempo de reparación hay que cuantificarlo en función del esfuerzo técnico, difícil de medir de forma objetiva.

- En la estipulación IX “General”, en el apartado “Contrato completo”, en las páginas 8 - 9, no queda claro qué documentos actúan como anexo al contrato.

- En la estipulación X “Costes”, en la página 9, el término “costes” es ambiguo, se debería hablar de precio del producto. En términos contractuales, el coste se refiere al valor de los equipos, suministros, servicios, etc. que se utilizan para la creación del producto o servicio, mientras que el precio es lo que ha de pagar el comprador por el producto final.

- En la estipulación X “Costes”, en la página 9, se habla de que el pago habrá de realizarse en dos plazos, pero no se especifica las cantidades a abonar en cada uno de los plazos ni la forma de pago.

Errores detectados en el documento “Contrato de desarrollo”:

- En la estipulación III “Ciclo de desarrollo”, en el apartado e, en la página 5, se habla de las condiciones de penalización, pero no aparecen en ninguna parte del contrato.

- En la estipulación IV “Entrega de los productos de software”, en el apartado “Entrega final”, en la página 7, se dice que la fecha de entrega final es el 26/01/2015. Más adelante, se indica que la fecha de finalización de la garantía es el 04/02/2015, por lo que el periodo de garantía sería de 10 días, cuando la ley establece de forma general entre 6 meses y 2 años.

- En la estipulación IV “Entrega de los productos de software”, en el apartado “Nulidad o anulabilidad”, en la página 9, se hace referencia a los cambios en los requisitos que deberían estar en el Anexo I, pero éste no existe.

- En la estipulación VI “Propiedad intelectual”, en la página 6, se especifica que corresponde a la empresa cliente los derechos de explotación derivados de la Ley de Propiedad Intelectual. Sin embargo, la Ley de Propiedad Intelectual establece que los derechos de explotación de un software corresponden al autor o autores. Cuando se compra un software se adquieren derechos sobre la licencia de uso, no sobre el derecho de explotación.

- En la estipulación VII “Garantía y soporte”, en la página 7, no aparece la fecha de la finalización de la garantía, sin embargo, ésta si aparece en la estipulación X “Costes” (04/02/2015), en la página 9, como fecha de pago del segundo plazo.

Teniendo en cuenta que no solamente se han detectado ambigüedades sino también errores, se considera que este punto no se cumple.

Recomendaciones:

- Precisar cuáles son los anexos o bien incluirlos si no se ha hecho.

- Especificar de forma clara el tiempo de respuesta ante incidencias y el tiempo para las reparaciones.

- La estipulación X “Costes” debería tener como título “Precio”. Además se ha de aclarar la cuantía a abonar en cada uno de los dos pagos y la forma de pago.

9.2) Se tienen en cuenta los factores de calidad.

Se ha realizado un análisis de los factores de calidad del software y se ha podido averiguar que han sido tenidos en cuenta los siguientes:

- Factores de operación del producto:
    - Corrección:
        - El tiempo de respuesta del sistema no debe superar los 5 segundos. Esta información puede encontrarse en el requisito no funcional RQN-4 del apartado 5.3 “Requisitos no funcionales”, del documento “Especificación de requerimientos del software”, en la página 8.

- Eficiencia:
  - Según se especifica en el apartado 7. “Requerimientos mínimos”, del documento “Análisis y diseño funcional”, en la página 48. Los requisitos mínimos del sistema están definidos por los requerimientos del gestor de base de datos así como el servidor Web que desee utilizarse, sin embargo, solo se proveerá soporte a sistemas desplegados en los siguientes ambientes: PHP > 5.6, Mysql > 5.6, Apache 2.4 y Google Chrome > 40.0.
- Integridad:
  - En el apartado 5.1 “Actores”, del documento “Especificación de requerimientos del software”, en la página 6, se definen los tipos de usuarios y qué papel desempeñan.
  - En el apartado 5.3 “Requisitos no funcionales”, del documento “Especificación de requerimientos del software”, en la página 8, se indica en el requisito RQN-3, cómo debe de garantizarse la seguridad del módulo de backend a través de usuarios que son identificables por una contraseña.
- Usabilidad:
  - Se ha elaborado un documento “Manual de usuario”. En éste se explica el funcionamiento del sistema tanto a los usuarios comunes como a los usuarios administradores. Éste es uno de los documentos que se le entregará al cliente final para que pueda aprender a usar las funcionalidades del sistema.

- Factores de revisión del producto:
  - Flexibilidad y Reusabilidad:
    - En el apartado 5.5. “Requisitos de desarrollo”, del documento “Especificación de requerimientos de software”, página 8, se especifica que el ciclo de vida para el desarrollo del producto está basado en la construcción de un prototipo evolutivo orientado a objetos, de forma que se pueden incorporar de forma más sencilla cambios y nuevas funciones, así como aprovechar las ventajas de la reusabilidad proporcionada por el paradigma de orientación a objetos.
  - Mantenibilidad:
    - Desde el punto de vista de la implementación, el proyecto posee una arquitectura MVC e implementa patrones de software para su mantenimiento, además ha sido construido de forma modular y se vale de lo flexible y reutilizable que es el desarrollo mediante el uso del paradigma de orientación a objetos.

- Otros factores:
    - Seguridad:
        - En el apartado 5.2.3 “Alertas de Stock de productos”, del documento “Especificación de los Requerimientos de Software” en la página 7, se especifica el requisito funcional RQ-6, en el cual se indica que el administrador debe de poder ver las alertas del stock de su inventario.
    - Manejabilidad:
        - Según lo especificado en el apartado 6.1 “Repositorios de versiones”, del documento “Analisis y Diseño funcional”, en la página 48, se ha utilizado GitHub como sistema de alojamiento del código en la nube para poder ser utilizado con el sistema de control de versiones Git.

Se considera que los factores de calidad del software se han tenido en cuenta de forma parcial, pues es necesario aclarar algunos detalles y añadir algún factor de calidad adicional.

Recomendaciones:

- Deben pulirse los factores de corrección, detallándose las características de las salidas del sistema en el documento “Especificación de requerimientos del software”, en el apartado 5.2 “Requisitos funcionales”, en las páginas 7 - 8, ya que estos requisitos deberían tener un nivel de detalle mucho mayor con información complementaria como los tipos de datos de salida, descripción de lo que representa la salida, mensajes informativos o de error que pueden producirse, etc.

- Ya que los requisitos mínimos del sistema están definidos por el gestor de base de datos (MySQL) así como el servidor (XAMPP), se debería indicar qué prestaciones hardware y software son necesarias para ejecutar ambos, remarcando sobre todo la información en lo referente a las necesidades de la memoria, espacio en disco necesario, sistema operativo, etc.

- En lo que respecta a la confiabilidad, sería adecuado aportar información respecto a la tasa máxima de fallos que tolera el sistema en situaciones bajo presión, por ejemplo cuando se realizan muchos pedidos de forma simultánea o el número de usuarios haciendo uso del sistema es demasiado alto.

- En términos de mantenibilidad, desde el punto de vista de la documentación, faltaría por aportar cierta información del proyecto para garantizar su mantenibilidad, como mínimo la relacionada con algunos aspectos como la seguridad y el testing. Por otra parte, desde el punto de vista de la implementación, falta mejorar la documentación del código fuente del producto, así como el de las pruebas unitarias realizadas.

- En cuanto a la portabilidad del sistema, no se ha especificado nada concreto que indique que la aplicación construida debería de ser portable, sin embargo, esta sí ha sido construida para poder ejecutarse en cualquier sistema Windows o de tipo Unix, ya que emplea un servidor de bases de datos XAMPP para la ejecución del software PHP y el sistema gestor de bases de datos MySQL, según lo especificado en el apartado 4.6. “Entorno de la aplicación”, del documento “Análisis y diseño funcional”, página 30. Teniendo en cuenta esta información, sería conveniente aclarar el factor de portabilidad del sistema como un requisito más.

10.1) Se consideran los estándares necesarios.

En el documento “Plan de calidad”, en el apartado 7. “Estándares de documentación”, página 14, se expone como la empresa desarrolladora ha seguido un estándar de documentación para garantizar la unicidad de formatos y estándares de documentación para su fácil lectura y revisión. En este estándar, se define el formato que debe tener todo documento generado, así como un marco conceptual a seguir para el Plan de desarrollo y la especificación de los requisitos. El estándar seguido parece propio de la empresa desarrolladora, pues no hay evidencias de que esté avalado por el IEEE o la ISO.

Por lo anterior, se considera que los estándares se han tenido en cuenta parcialmente.

Recomendaciones:

- Sería conveniente que la empresa se ajustara a un estándar normalizado como el ISO, que está avalado mundialmente y aporta beneficios, como una mayor competitividad de la empresa y el incremento en la calidad de sus productos.

10.2) Se siguen los estándares necesarios.

La empresa solamente sigue su propio estándar. A pequeña escala, el hecho de tener un estándar propio sería suficiente para pequeños comercios, sin embargo, si se quisiera ampliar el negocio a grandes empresas y multinacionales, sería mejor contar con un certificado ISO, ya que estas empresas así lo demandan. Por lo tanto, se considera que este punto se ha cumplido parcialmente.

Recomendaciones:

- De cara a aumentar la calidad y la competitividad de sus productos, se recomienda obtener un certificado ISO.

Énfasis

Los problemas más graves han sido:

- La falta de compromiso con la seguridad del sistema, que no garantiza la seguridad necesaria para proteger los datos del negocio ni de los clientes.
- La ausencia de cualquier tipo de revisión, lo que repercute en la calidad de los diseños, la codificación del producto, la realización de sus pruebas y la generación de los artefactos.
- La no supervisión de los riesgos del proyecto.
- La ausencia, en general, de comentarios en el código fuente.
- La falta de la consecución del tercer hito, pues no se ha terminado de completar la verificación del funcionamiento del sistema a través de la superación de todas las pruebas, por lo que hay constancia de que el sistema desarrollado no es completamente funcional.

Informe de gestión

En esta auditoría no hemos tenido acceso a documentación sobre la gestión.

Resumen

Se han encontrado incumplimientos en el seguimiento del proyecto, la realización de las revisiones y en la elaboración de la correspondiente documentación del proyecto. Estos incumplimientos han tenido un impacto significativo en la calidad y lo completo que está el producto, y también la tendrá en la mantenibilidad del proyecto, desde el punto de vista de la documentación y la codificación.

Por tanto, se recomienda elaborar la documentación que falta, así como mejorar el estado del producto a partir de la realización de nuevas revisiones, con el objetivo de favorecer la sostenibilidad del proyecto de cara a futuro.

Resultado

DESFAVORABLE, la gestión del proyecto es incorrecta y la calidad del producto no es aceptable.

Fecha y firma

Madrid, 20/01/2021

Manuel Guerrero Moñús