Software Project Management 2019/2020

Bill Splitter

D1.2.2 - Quality Assurance Plan

Autores

- Paulo Dias (dias.pauloalex@gmail.com)
- Sérgio Soares (smcsoares20@gmail.com)
- Rui Mota (ruifilipetmota@gmail.com)
- Ruben Marques (rubenantoniomarques@gmail.com)

Estado

- Revisão

Versões Principais

- V0.1, 15 de Outubro, Ruben Marques, Primeiros rascunhos
- V0.2, 16 de Outubro, Rui Mota, Desenvolvimento Principal
- V0.3, 19 de Outubro, Paulo Dias, Rectificação dos tópicos 3. e 3.1.
- V0.4, 23 de Outubro, Sérgio Soares, Desenvolvimento
- V0.5, 27 de Outubro, Paulo Dias, Atualização da tabela de riscos
- V0.6, 30 de Outubro, Rui Mota, Gestão dos riscos
- V0.7, 30 de Outubro, Rui Mota, Ajustes finais

Versões Publicadas

- V1, 30 de Outubro, Rui Mota, Versão aprovada

Índice

1. Introdução

O Plano de Garantia de Qualidade define quais são as qualidades desejadas para o produto e como é que vão ser avaliadas. Define ainda os métodos standard que permitam que esta qualidade seja assegurada no final do projecto.

Relatório - Avaliação de Qualidade

2. Objetivos de Qualidade (Quality Goals)

Os atributos externos mais importantes para este projecto são os seguintes:

- Confiança - Os utilizadores devem poder confiar na qualidade dos cálculos efetuados pela aplicação.
- Usabilidade - A aplicação deve ser de simples compreensão por parte do utilizador.

A respeito das qualidades internas, os atributos mais importantes são:

- Compreensibilidade - O código deve ser de fácil compreensão.
- Modularidade - O código deverá apresentar independência entre interface e lógica.

3. Revisões (Reviews)

A equipa compromete-se a realizar uma reunião com o cliente semanalmente e a expor os relatórios realizados pelo anotador da equipa no seguinte endereço: Reuniões com cliente

3.1. Inspeções (Inspections)

As inspeções têm como principal objetivo identificar anomalias num ficheiro para que mais tarde o autor possa corrigir para, enfim, ser aprovado. Cada inspeção terá uma ordem de eventos pré-determinados tais como: planeamento, preparação, reunião, correção , validação e será

realizada por elementos da equipa mais um elemento de outra equipa que obterá o papel de Inspetor/Revisor, segue a lista de papéis:

- **Revisor** - Responsável por ler e apontar possíveis incorreções que serão discutidas com a equipa (exceto o autor) e posteriormente anotado caso a equipa não seja capaz de a esclarecer.

- **Autor** - Elemento da equipa que produziu o documento em causa.

- **Moderador** - Responsável por preparar a reunião e conduzi-la da melhor forma possível. Poderá também assumir a responsabilidade do Anotador. Este papel não poderá ser assumido pelo Autor

- **Anotador** - Responsável por anotar erros/problemas encontrados. Este papel não poderá ser assumido pelo Autor.

Poderá encontrar os relatórios resultantes aqui: Inspeções

3.2. Desk checks

Uma Desk check é uma revisão manual de um pedaço de código de um programa. Consiste na leitura de funções dentro do código e no seu teste manual, envolvendo muitas vezes papel e caneta como forma de registo do processo e output destas. Por exemplo, pode-se seguir uma variável e o respetivo valor do início ao fim da função, correndo o código linha a linha como forma de ajuda à descoberta de erros de lógica ou problemas e ineficiência.

3.2. Walk-throughs

Um walk-through consiste na revisão em grupo de um produto de software (ficheiros de código fonte ou documentos), onde o programador/designer daquele elemento guia as outras partes ao longo de uma análise completa em busca de possíveis erros e problemas.
Os ficheiros em questão devem ser previamente analisados por todas as partes, que devem tomar nota de qualquer detalhe passível de receber atenção, e devem posteriormente indicar esses mesmos detalhes na reunião.

4. Testes (Tests)

4.1. Teste Unitários (Unit Testing)

É o processo para testar componentes individuais . Aqui serão testadas todas as classes individualmente de forma a garantir que estas têm um funcionamento correcto que permita que essa unidade seja integrada com as restantes e que tem a performance desejada.

4.2. Teste de Integração (Integration Testing)

Nesta fase, todos os componentes individuais e módulos são combinados um a um e testados em grupo. Tem como objetivo expor defeitos nas interfaces e na integração de componentes com outros.

4.3. Teste de Aceitação (Acceptance Testing)

O principal objetivo dos teste de aceitação é verificar se o projecto satisfaz as condições para ser entregue ao cliente. O software é testado como um todo e se vai de encontro com as necessidades e requisitos impostos pelo cliente. Além dos testes de aceitação, serão também feitos testes de usabilidade realizados por utilizadores com um intuito de garantir melhor qualidade.

Links do plano de testes e do relatório:

Plano de Testes
Relatório

5. Gestão do Risco (Risk Management)

Nesta secção são identificados os riscos e atribuídas classificações a cada um. Depois de identificados deverão ser feitos planos de modo a evitar ou diminuir os riscos onde são depois monitorizados durante todo o desenvolvimento do projecto .

5.1. Identificação e Avaliação do Risco (Risk identification and assessment)

A gravidade de cada um dos riscos será avaliada com base em dois fatores. Probabilidade, que será avaliada de 1 a 5 (baixa a muito alta), e Impacto, que será avaliado em 1, 3 ou 5, sendo 1 tolerável, 3 sério e 5 catastrófico.

Como maneira de obter um valor de avaliação geral, serão multiplicados os valores atribuídos nos dois fatores (Total = P * I).

Lista de Riscos		Prob.	Impacto
1	Problemas de ligação com o servidor	3	5
2	Não obter conhecimento suficiente para concluir a interface a tempo da entrega de GPS	4	5
3	Desistência de um membro de equipa	1	5
4	Avaria dos equipamentos	1	3
5	Falha na estimativa	5	3
6	Indisponibilidade da equipa para se reunir	2	1
7	Desentendimento entre membros da equipa	2	3

Probabilidade de ocorrer - muito baixa(1), baixa(2), media(3), alta(4) ou muito alta(5)

Impacto do risco - tolerável(1), sério(3) ou catastrófico(5)

Lista de Riscos

5.2. Monitoramento e Controlo de Risco (Risk monitoring and control)

Reunião semanal de forma a avaliar o desenvolvimento atual e analisar quais os riscos que se podem manifestar de dentro dos seguintes riscos:

- Problemas de ligação com o servidor
  Risco com P x I = 15

- Não obter conhecimento suficiente para concluir a interface a tempo da entrega de GPS
  Risco com P x I = 20

- Falha na estimativa
  Risco com P x I = 15
  Documento para prevenção: Plano de Riscos

6. Padrões de Programação (Coding Standards)

Serão implementados a maioria dos padrões comuns da linguagem Java. Consultar o seguinte link para informação mais detalhada: Lista de Padrões

7. Métricas de Qualidade (Quality metrics)

- Confiança - resultado dos testes unitários ao módulo da lógica (100% de testes aprovados)
- Usabilidade - serão feitos testes de usabilidade onde vão ser atribuídas classificação à usabilidade da aplicação (0-10)
- Compreensibilidade - relatório de avaliação de desk checks e de inspeção ao código / cumprimento dos coding standards
- Código dividido em módulos (>= 3)
- Satisfação do cliente (>= 85%) - O projecto irá ser avaliado consoante os documentos entregues em cada semana
- Concordância com as datas estabelecidas no EVA (>= 85%) - Será feito um rastreio semanal do documento EVA