

3.3. - Testes [TST]

Unidade: TST	Data: 02.10.2018	Gestor: Tomás Lopes
Versão: 2.0	Status: Baselined	Vice Gestor: Ricardo Gonçalves

Lista de membros desta unidade:

Tomás Lopes	LEI	Coordenador	tomasgabriel1998@gmail.com
Ricardo Gonçalves	LDM	Vice-Coordenador	rgoncalvestlm@gmail.com
João Campos	LDM	Colaborador	jgmmcampos@gmail.com
André Correia	LEI	Colaborador	andrefcorreia88@gmail.com
Guilherme Nogueira	LEI	Colaborador	gnogueira@student.dei.uc.pt

Processos utilizados por esta unidade:

Descrição do processo	ID do Processo
1. Processo para escrita de testes.	MQ_LABSYNC_TST_ESCRITA_TST
2. Processo para averiguar se requisitos estão a ser cumpridos.	MQ_LABSYNC_TST_AVR_REQ
3. Processo para testar código.	MQ_LABSYNC_TST_CÓDIGO
4. Processo para deployment.	MQ_LABSYNC_TST_DEPLOYMENT

Objetivos da unidade de TST:

- 1) Garantir que os requisitos do cliente são implementados e que o código desenvolvido é bem aplicado, tendo em conta os diagramas elaborados pela unidade de requisitos.
- 2) Garantir a qualidade e o bom funcionamento do código implementado.
- 3) Fazer deployment do programa.

3.3.1. - MQ_LABSYNC_TST_ESCRITA_TST

Descrição do processo:

Consiste na escrita do procedimento a ser realizado por um *tester* (pessoa que testa código), em particular as pré-condições, os inputs (texto a ser preenchido e botões a clicar) e o resultado esperado. As pré-condições contêm o estado do sistema antes do teste ser realizado (dados da aplicação e página em que o utilizador se encontra), os inputs referem-se aos dados inseridos pelo tester e que pretendem alterar o sistema (exemplo, criar uma nova entidade) e o output diz respeito ao estado final do sistema e resultados obtidos.

Para os requisitos funcionais, existe o documento “Testes funcionais”.

São baseados na lista de requisitos. Cada teste tem:

- 1) Objetivo do teste.
- 2) Página de pré-condição.
- 3) Valores de pré-condição.
- 4) Ação a realizar.
- 5) Resultado esperado.
- 6) Resultado obtido.
- 7) Data, realizador e horas despendidas no teste.

Um requisito é aprovado se o resultado esperado coincidir com o resultado pretendido. Caso contrário, é enviada uma mensagem à equipa de implementação de forma a estes corrigirem o programa.

Para os requisitos não-funcionais, existe o documento “Testes não-funcionais”. Estes testes servem para averiguar se os requisitos não-funcionais estão a ser cumpridos, podendo estes ser de natureza diversa. Existem testes de carga e testes de usabilidade.

Cada teste de carga tem:

- 1) Constituição do teste.
- 2) Reação do sistema (a que ponto começa a mostrar alterações).
- 3) Autor e horas despendidas.
- 4) Se as métricas o indicarem, pode ter de haver correção do código.

Cada teste de usabilidade tem:

- 1) Constituição do teste.
- 2) Número de cliques ideal.
- 3) Número de cliques efetuado.
- 4) Tempo ideal.
- 5) Tempo efetivo.
- 6) Tarefa concluída? (o utilizador pode não conseguir completar a tarefa, sendo um forte argumento a favor da dificuldade da utilização do programa).

7) Autor e horas despendidas.

Caso as métricas desejáveis e as métricas obtidas sejam muito diferentes, pode ter de haver um redesign do layout.

Inputs:

- Lista de requisitos
- Lista de mockups

Entry Criteria:

Uma mensagem gerada no Discord, enviada do gestor de implementação, para o gestor desta unidade a informar qual a tarefa a ser implementada.

Task Description:

São distribuídas tarefas pelos vários elementos da unidade. Cada elemento terá que usar, como base, os testes no documentos “Testes funcionais” e “Testes não-funcionais”.

Exit Criteria:

Assim que cada requisito contenha um conjunto de *use cases* suficiente para que seja testado (em média seis testes por requisito), então o processo dá-se como concluído, sendo que será mais tarde utilizado por esta unidade para testar código implementado. Apesar de concluído e em uso num outro processo, este documento pode ser novamente reaberto, para adição/edição de casos de uso.

Métricas de avaliação deste processo:

- Número de testes criados por requisito;
- Qualidade das pré-condições e do output;
- Horas de trabalho semanais por elemento.

Outputs gerados:

Gera-se uma tabela de testes com as etapas a serem realizadas aquando de um teste. Esta tabela será preenchida na verificação de cumprimento de requisitos e de testes de código.

Documentos:

Todos os documentos estão no Git dentro da pasta de testes discriminado para cada Sprint. Estão lá presentes os documentos template para que servem de guia para os testes (em formato word) e posteriormente estão presentes os documentos de resultados (em formato PDF).

Monitorização deste processo pela equipa de Qualidade:

A unidade de qualidade faz uma comparação entre a tabela de requisitos e a tabela de testes para saber se os testes criados testam efetivamente o código implementado, se vão de acordo com os requisitos do cliente.

3.3.2. - MQ_LABSYNC_TST_AVR_REQ&COD

Processo para averiguar se requisitos estão a ser cumpridos

Descrição do processo:

É feito uma série de testes a partir da lista de requisitos. No caso dos requisitos funcionais, estes são testados para averiguar se foram implementados ou não. No caso dos requisitos não-funcionais, estes são testados de acordo com os testes e métricas definidas no documento “Testes não-funcionais”. Os requisitos não funcionais devem ser testados de preferência com pessoas não familiares ao projeto.

Este processo verifica também o trabalho realizado pela equipa de implementação, ou seja, que satisfaça as normas de programação e interação com um utilizador (não permitindo a inserção de valores indesejados, por exemplo: uma expressão num campo numérico. Em suma, testa o código implementado.

Inputs:

- 1) O documento de requisitos enviado pela equipa de testes.
- 2) Versão do programa a testar.
- 3) O código realizado pela equipa de implementação que terá sido disponibilizado no GitHub, pelo gestor da unidade de implementação.

Output:

- 1) Lista de funcionalidades implementadas e por implementar.
- 2) Lista de testes elaborados por esta unidade de modo a testar requisitos não-funcionais, juntamente com as métricas recolhidas pelos diferentes elementos da equipa.
- 3) Horas despendidas por cada elemento em cada tarefa.

Entry Criteria:

O gestor da unidade de implementação envia uma mensagem pelo Discord ao gestor desta unidade de que há uma nova versão a ser testada. Este encarrega-se de elaborar uma lista de testes para requisitos não-funcionais e de distribuir tarefas. O resto da unidade acrescenta outros testes para requisitos não-funcionais caso ache pertinente.

Task Description:

São distribuídas tarefas pelos vários elementos da unidade. Cada elemento terá que averiguar se o sistema funciona corretamente e se está de acordo com os requisitos do cliente. Também deve recolher métricas e testar o programa inclusivamente com indivíduos fora do contexto de testes e implementação de forma a obter dados realistas e demonstrativos do utilizador-alvo. Ao testar o programa são também usados documentos com base nos documentos “Testes funcionais” e “Testes não-funcionais”. Para os testes funcionais, o *tester* deve inserir em cada parâmetro o valor descrito para o mesmo. Por último deve alterar o estado do teste para *passed* ou *failed*, consoante o resultado do teste esteja de acordo com o output descrito no vetor de testes. Caso o teste falhe, o requisito não foi satisfeito na totalidade, sendo gerada uma mensagem à unidade de implementação (via Trello) para que esta possa corrigir o erro/satisfazer o requisito presente na tabela de requisitos. Quando um conjunto de testes para o mesmo requisito se der como concluído, é gerada uma mensagem para a equipa de implementação a informar de que o código passou os testes, ou em caso de falha, é informada de tal, para que a tarefa de implementação de código correspondente ao requisito em causa, seja reaberta e o código corrigido. Para os testes não-funcionais, são realizados por elementos de preferência que estejam (de preferência) fora do contexto de implementação e testes e supervisionados por um elemento da unidade de testes.

Este deve indicar ao utilizador as ações a realizar e recolher métricas e comentários com base na experiência do utilizador. Caso os testes indiquem uma má experiência de utilizador, é enviada uma mensagem ao coordenador de implementação a mostrar o resultado dos testes de forma a este poder corrigir e melhorar a experiência de utilização.

Exit Criteria:

O processo é dado como concluído quando cada elemento tenha realizado vários testes para cada requisito e este cumpra todos os requisitos, sendo posteriormente atualizado o projeto principal. Caso contrário é enviada uma mensagem à equipa de implementação para a sua correção.

Métricas de avaliação deste processo:

- 1) Tabela com requisitos aprovados ou reprovados.
- 2) Número de testes realizados por requisitos.
- 3) Número e taxa de testes aprovados/reprovados por requisito;
- 4) Número de pessoas envolvidas em cada teste;
- 5) Horas de trabalho semanais por elemento.

Outputs gerados:

Gera-se uma tabela de requisitos, que conforme tenham sido cumpridos ou não, recebem uma denominação de “aprovado” ou “reprovado” e também os documentos “Testes funcionais” e “Testes não-funcionais”, agora com o resultado dos testes (aprovado/reprovado), com a data em que foi testado e, com observações ao mesmo

Monitorização deste processo pela equipa de Qualidade:

A unidade de qualidade faz uma leitura sobre o resultado de cada teste e averigua, em caso de falha, se a unidade de implementação foi notificada e se sabe o que deve editar.

3.3.4. - MQ_LABSYNC_TST_DEPLOYMENT

Descrição do processo:

O deployment é feito na própria máquina do tester, estando a aplicação limitada a esse utilizador.

Inputs:

O código desenvolvido pela unidade de implementação, disponível no GitHub.
Versão da aplicação.

Output:

- Aplicação deployed.

Entry Criteria:

Existência de uma versão do produto devidamente testada, para ser feito o deployment.

Validação:

Operação realizada sem erros fatais.

Métricas de avaliação deste processo:

Versão da aplicação;

Outputs gerados:

Número de horas despendidas na tarefa. Versão da aplicação.