

VISIBILIDADE DE PROJETO CONTÍNUA (VISPROJ)

Plano de Testes

Versão 1.0

Abril / 2013

Histórico de Revisões

Data	Versão	Descrição	Autor
<8/abril/13>	<1.0>	Release Inicial	Artur Pascualote Santos

ÍNDICE

1. INTRODUÇÃO.....	4
2. MASSA DE DADOS	4
3. FERRAMENTAS.....	5
3.1 TestLink.....	5
4. ESTRATÉGIA DE TESTE	6
4.1 Teste Estrutural	6
4.1.1 Teste de Unidade.....	6
4.2 Teste Funcional	6
4.2.1 Teste de Aceitação	6
4.2.2 Teste de Integração.....	7
5. TEMPLATE DE CASOS DE TESTE.....	8

1. INTRODUÇÃO

Esse documento do Plano de Testes do Visibilidade Contínua. compõe-se dos seguintes objetivos:

- Identificar informações de projeto existentes e os componentes de software que devem ser testados.
- Listar os Requisitos a Testar recomendados (alto nível).
- Recomendar e descrever as estratégias de teste a serem empregadas.
- Identificar os recursos necessários e prover uma estimativa dos esforços de teste.
- Listar os elementos resultantes do projeto de testes.

2. MASSA DE DADOS

A preparação de massa de dados é uma das etapas que mais pode consumir esforços e recursos e certamente define a qualidade dos testes funcionais. Vamos usar <http://www.generatedata.com/> que é uma ferramenta gratuita e de código aberto escrito em JavaScript, PHP e MySQL, que permite gerar e exportar nos mais diferentes tipos de formatos(CSV, Excel, JSON, SQL, HTML e XML) rapidamente grandes volumes de dados personalizados em uma variedade de formatos para uso em testes de software. Existe também a versão livre, GNU-licenciado disponível para download.

3. FERRAMENTAS

3.1 TestLink

TestLink é uma ferramenta case de gerenciamento de teste. Essa ferramenta pertence e é mantida pela comunidade de teste de software, em todo o mundo. TestLink é utilizado para a criação e elaboração de Casos de teste e cenários de teste. Nessa ferramenta também é possível vincular os casos de teste de forma hierárquica e a um mesmo grupo de teste. TestLink não é uma ferramenta que executa caso de teste, essa ferramenta é utilizada para documentar os testes que serão executados por outras ferramentas ou manualmente.

4. ESTRATÉGIA DE TESTE

Nesta seção estão descritas os tipos e técnicas de teste a serem utilizados na cobertura dos artefatos.

4.1 Teste Estrutural

4.1.1 Teste de Unidade

Teste de unidade objetiva testar a menor unidade funcional de um código, no caso de projetos orientados a objetos a menor unidade é o método de uma classe. Os testes unitários deverão ser realizados por cada desenvolvedor, ou seja, cada componente de software que este desenvolver deverá conter também os testes unitários. Para esse projeto foram eleitos os dois principais componentes do sistema: Control e o Service. São eles que realizam as regras do negócio. O critério de aceitação é que possua cobertura de código de 80%.

4.2 Teste Funcional

Teste funcional é aquele onde o objetivo é verificar se uma dada implementação está de acordo com a sua respectiva especificação. Também é conhecido como teste “Caixa Preta”. Um tipo de teste funcional é o chamado teste formal. Nele, tanto as especificações como os procedimentos de geração de casos de teste são formalizados. Os casos de teste devem ser capaz de verificar a conformidade com os requisitos. Os casos de teste serão mantidos na ferramenta TestLink fornecida no ambiente da Fábrica de Software. Técnicas que serão utilizadas serão:

- Análise de Valores Limites.
- Partição por Equivalência.

4.2.1 Teste de Aceitação

Tem por função verificar o sistema em relação aos seus requisitos originais, e às necessidades atuais do usuário.

Teste de Aceitação formal:

- Executado pelo Usuário final ou por grupo de teste independente
- Funções e recursos devem ser conhecidos
- Pode ser automatizado
- Deve ser monitorado e medido

Teste Alfa:

No teste Alfa, os procedimentos para executar o teste não são definidos com tanto rigor como no teste de aceitação formal. As funções e as tarefas de negócios a serem exploradas são identificadas e documentadas, mas não há casos de teste específicos para seguir. O testador individual determina o que fazer. Essa abordagem de teste de aceitação não é tão controlada como o teste formal e é mais subjetiva do que o tipo formal.

Teste Beta:

O teste beta é o menos controlado das três estratégias de teste de aceitação. Nesse tipo de teste, a quantidade de detalhes, os dados e a abordagem adotada são de inteira responsabilidade do testador individual. Cada testador é responsável por criar o próprio ambiente, selecionar os dados correspondentes e determinar as funções, os recursos ou as tarefas a serem exploradas. Cada um deles é responsável por identificar os próprios critérios que o levarão a aceitar ou rejeitar o sistema no seu estado atual.

O teste beta é implementado por usuários finais, geralmente com pouco ou nenhum gerenciamento por parte da organização de desenvolvimento (ou outra que não seja do usuário final).

4.2.2 Teste de Integração

Verifica basicamente se as unidades testadas de forma individual executam corretamente quando colocamos juntas. A interação pode ser executada de uma forma não incremental, onde as unidades são combinadas e o programa completo é testado como um todo. (Interação big bang). Uma maneira mais eficiente seria integrar as unidades de forma incremental, as unidades

gradativamente testadas e implementadas. A integração incremental mostra-se como uma técnica mais apropriada, pois é mais fácil isolar a causa dos erros quando se testa por partes menores.

5. TEMPLATE DE CASOS DE TESTE

CASO DE TESTE	É o nome exclusivo usado para identificar este caso de teste.
CASO DE USO	Indica o caso de uso correspondente ao caso de teste.
PRÉ-CONDIÇÕES	Indica o estado do sistema para executar o caso de teste.
PROCEDIMENTOS	Os passos necessários para chegar ao resultado esperado.
DADOS DE ENTRADA	Dados utilizados para a realização do caso de teste.
DADOS ESPERADOS	O que se espera pela execução do caso de teste.
DADOS OBTIDOS	Dados obtidos com a realização do caso de teste.