**<DotProject EAP>**

**<Build 1.4 > Plano de Teste**

**Versão <1.4>**

**Histórico da Revisão**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 04/12/11 | 1.0 | Criação do documento e de seu *template* | José Carlos, Guilherme Fay |

**Índice Analítico**

Introdução 4

Finalidade 4

Escopo 4

Terminologia e Acrônimos do Documento 4

Missão de Avaliação

Referências 4

Resumo dos Testes Planejados 4

Casos de teste 5

1.1 Técnicas e Tipos de Teste

1.1.1 Teste da Interface do Usuário 5

1.1.2 Teste de Configuração 5

Necessidades Ambientais 6

Hardware Básico do Sistema 6

Elementos de Software Básicos do Ambiente de Teste 6

Responsabilidades e Perfil da Equipe 7

Pessoas e Papéis 7

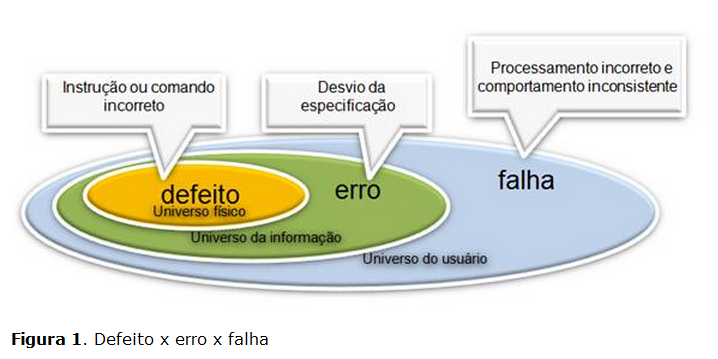
**<Build 1.3> Plano de Teste**

**Introdução**

Teste de software é um processo da engenharia de software. Tem como um de seus objetivos fornecer informações sobre a qualidade de software. Essa qualidade é obtida através da comparação entre os resultados do teste realizado e as especificações pré-definidas para o produto. Os testes de software revelam falhas, as quais precisam ser corrigidas pela equipe de desenvolvimento antes da entrega do produto ao cliente.

Quando falamos em testes de software, é necessário deixar claro alguns conceitos importantes, e que muitas vezes, são confundidos entre sí, são eles:

* Defeito é um ato inconsistente cometido por um indivíduo ao tentar entender uma determinada informação, resolver um problema ou utilizar um método ou uma ferramenta. Por exemplo, uma instrução ou comando incorreto. (IEEE 610, 1990)
* Erro é uma manifestação concreta de um defeito num artefato de software. Diferença entre o valor obtido e o valor esperado, ou seja, qualquer estado intermediário incorreto ou resultado inesperado na execução de um programa constitui um erro.(IEEE 610, 1990).
* Falha é o comportamento operacional do software diferente do esperado pelo usuário. Uma falha pode ter sido causada por diversos erros e alguns erros podem nunca causar uma falha.(IEEE 610, 1990).



Antes de iniciar os testes em um software é necessário elaborar um plano de testes. Elaborar o plano de testes é a primeira atividade da equipe de testes. Nesse planejamento são definidos os objetivos e missões do teste, nível, tipo e técnicas aplicadas ao teste. Além disso, são definidos riscos, recursos e prazos para a realização dos testes. O plano de teste é um dos documentos que permite que os testes sejam controlados e repetidos.

**Finalidade**

Este *Plano de Teste* referente ao Dot Project EAP esta coerente as seguintes estórias:

***4- Excluir qualquer nó, com exceção de nó 0;***

***6- Cada nó deve criar novos filhos, formando uma hierarquia;***

***8- Criar ultimo nível que será de esforço;***

O teste será um teste de usabilidade, no qual objetivo é definir se a interface como um todo está implementada corretamente (como deve ser uma EAP) e usual.

Os recursos necessários são um computador com servidor web, apache, um banco de dados MySQL, e o código fonte.

Os elementos a serem liberados serão scripts de javascript que contém as informações referentes a build 1.4.

**Escopo**

Os testes a serem realizados são os seguintes testes; de interface do usuário e de configuração. O teste se caracteriza por ser um teste de unidade, ou seja, é somente um pequeno pedaço do código principal, além de teste de aceitação, onde um grupo de usuários finais, irão avaliar a interface implementada da EAP. O teste diz respeito à interface para realizar operações com a EAP de um projeto, estar presente e estar corretamente acoplado ao resto do código, esse teste não diz respeito as funcionalidade implementados dentro do menu.

Será feito também um teste de cobertura de código, que mostrará qual porcentagem do código total foi testado.

**Terminologia e Acrônimos do Documento**

|  |  |
| --- | --- |
| **Nome** | **Descrição** |
| **MySQL** | Banco de dados |
| **Apache** | Servidor Web |
| **EAP** | Estrutura analítica do projeto |
| **Nó** | Um nível de hierarquia dentro de uma EAP. |
| **Pai** | Um grupo de trabalho que possui níveis a baixo dele. |
| **Filho** | Um grupo de trabalho que possui níveis acima dele. |

**Referências**

Documento de gerência da Sprint (DPEAP\_BKL\_20111121).

Dentro do documento as tabelas:

*Product Backlog*

*Sprint*

Documento de especificação suplementar (DPEAP\_ESP\_20110925).

**Missão de Avaliação**

* Localizar o maior número de erros possíveis;
* Verificar se a implementação atende o estabelecido nos requisitos;
* Verificar se os teste atendem aos critérios de aceitação.

**Resumo dos Testes Planejados**

Os testes que serão executados serão:

* Teste de interface do usuário;
* Teste de configuração.

**Casos de teste**

* **Técnicas e Tipos de Teste**

*Teste da Interface do Usuário*

|  |  |
| --- | --- |
| Objetivo da Técnica: | * Verificar se cada nó do nível 1 cria filhos; * Verificar se ao excluir um pai, todos os seus filhos também serão eliminados. |
| Técnica: | * Dentro da aba EAP, a qual se encontra dentro da janela do projeto, a interface para realização de suas operações deve estar implementada; * 1.1) Verificar se é possível criar um pacote de trabalho no nível 1; * 1.2) Verificar se é possível criar um novo, a partir da criação do nível 1, relacionando-os como pai e filho. * 1.3) Verificar se é possível ao excluir um nó no nível 1, seus filhos a partir do nível 2, também serão excluídos. |
| Estratégias: | * Verificar de forma manual se a interface está implementada da maneira especificada pela técnica. * Será realizado um teste de cobertura de código para verificar a porcentagem do que esta coberto pelos casos de teste |
| Ferramentas Necessárias: | * Para realização do teste de cobertura de código será utilizada a ferramenta *PHP~PHP5\_TestCoverage.*. * Serão utilizados como ferramentas os navegadores especificados no documento DPEAP\_ESP\_20110925 |
| Critérios de Êxito: | * O teste terá êxito se a interface estiver implementada com êxito nas técnicas acima, sem erros, além do grupo de usuários selecionados passarem um *feedback* positivo para a equipe de testes; * As estórias 4, 6 e 8 devem ter sido satisfatoriamente implementadas, e não contendo erros, e passar pelos critérios de aceitação. |

*Teste de Configuração*

|  |  |
| --- | --- |
| Objetivo da Técnica: | * A técnica deve averiguar se o objeto implementado apresenta configurações adequadas aos navegadores especificados pelo documento DPEAP\_ESP\_20110925 |
| Técnica: | * Serão realizados teste de compatibilidade entre os navegadores especificados no documento DPEAP\_ESP\_20110925, e a aba implementada. |
| Estratégias: | * Testar um por vez os navegadores especificados no documento DPEAP\_ESP\_20110925 |
| Ferramentas Necessárias: | * Para realização do teste nenhuma ferramenta de automação será utilizada. * Serão utilizados como ferramentas os navegadores especificados no documento DPEAP\_ESP\_20110925 |
| Critérios de Êxito: | * O teste terá êxito se o software estiver corretamente visível em todos os navegadores testados. |

**Necessidades Ambientais**

**Hardware Básico do Sistema**

São necessários os seguintes elementos de hardware básicos no ambiente de teste deste *Plano de Teste*.

|  |  |  |
| --- | --- | --- |
| **Recursos do Sistema** | | |
| **Recurso** | **Quantidade** | **Nome e Tipo** |
| Servidor de Banco de Dados |  |  |
| —Rede ou Sub-rede |  | A ser definido |
| —Nome do Servidor |  | A ser definido |
| Nome do Banco de Dados |  | A ser definido |
| PCs de Teste Cliente |  |  |
| Inclua requisitos de configuração especiais |  | A ser definido |
| Repositório de Teste |  |  |
| —Rede ou Sub-rede |  | SVN |
| —Nome do Servidor |  | A ser definido |
| PCs de Desenvolvimento de Teste |  | A ser definido |

**Elementos de Software Básicos do Ambiente de Teste**

São necessários os seguintes elementos de software básicos no ambiente de teste deste *Plano de Teste*.

|  |  |  |
| --- | --- | --- |
| **Nome do Elemento de Software** | **Versão** | **Tipo e Outras Observações** |
| MySQL | 5.5.16 | Banco de dados |
| Apache | 2.2.21 | Servidor Web |
| Internet Explorer | 9.0.8112.16421 | Navegador da Internet |
| Mozila Firefox | 7.0.1 | Navegador da Internet |
| Safari - Windows | 5.1.1 | Navegador da Internet |
| Google Chrome | 14.0.835.202 m | Navegador da Internet |
| Windows | 7 SP 1 | Sistema Operacional |
| Tortoise | 1.6.16 | Cliente para *sub-version* |
| NetBeans | 7.0 | IDE de desenvolvimento |
| *SimpleTest* | 1.3.4 | Ferramenta para teste unitário em php |
| *PHP~PHP5\_TestCoverage* | 1.2.25 | Ferramenta para teste de cobertura de código em php |

**Responsabilidades e Perfil da Equipe**

**Pessoas e Papéis**

Esta tabela mostra as suposições referentes ao perfil da equipe do esforço de teste.

|  |  |
| --- | --- |
| **Papel** | **Responsabilidades ou Comentários Específicos** |
| Gerente de Testes | Supervisiona o gerenciamento.  Estas são as responsabilidades:   * Planejamento * Adquirir recursos apropriados * Apresentar relatórios de gerenciamento * Defender os interesses do teste * Avaliar a eficiência do esforço de teste |
| Testador | Implementa e executa os testes.  Estas são as responsabilidades:   * Implementar os testes e os conjuntos de testes * Executar os conjuntos de testes * Registrar os resultados * Documentar incidentes |