**Gerador Burndown**

**Plano de Teste de <Iteração/Mestre>**

**Versão 1.0**

**Histórico da Revisão**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| *12/11/2014* | *1.0* | *Release Inicial* | Ana Carvalho,  Caroline Lopes, Naiara Silva e Vinícius Silva |

**Índice Analítico**

[1. Introdução](http://www.wthreex.com/rup/webtmpl/templates/test/rup_tstpln.htm#1.%20%20%20%20%20Introduction)

[1.1 Finalidade](http://www.wthreex.com/rup/webtmpl/templates/test/rup_tstpln.htm#1.1%20%20%20%20%20Purpose)

[1.2 Escopo](http://www.wthreex.com/rup/webtmpl/templates/test/rup_tstpln.htm#1.2%20%20%20%20%20Scope)

[1.3 Público-alvo](http://www.wthreex.com/rup/webtmpl/templates/test/rup_tstpln.htm#1.3%20%20%20%20%20Intended%20Audience)

[1.4 Terminologia e Acrônimos do Documento](http://www.wthreex.com/rup/webtmpl/templates/test/rup_tstpln.htm#1.4%20%20%20%20%20Document%20Terminology%20and%20Acronyms)

[1.5 Referências](http://www.wthreex.com/rup/webtmpl/templates/test/rup_tstpln.htm#1.5%20%20%20%20%20References)

[2. Missão de Avaliação e Motivação dos Testes](http://www.wthreex.com/rup/webtmpl/templates/test/rup_tstpln.htm#2.%20%20%20%20%20Evaluation%20Mission%20and%20Test%20Motivation)

[2.1 Fundamentos](http://www.wthreex.com/rup/webtmpl/templates/test/rup_tstpln.htm#2.1%20%20%20%20%20Background)

[2.2 Missão de Avaliação](http://www.wthreex.com/rup/webtmpl/templates/test/rup_tstpln.htm#2.2%20%20%20%20%20Evaluation%20Mission)

[2.3 Motivadores dos Testes](http://www.wthreex.com/rup/webtmpl/templates/test/rup_tstpln.htm#2.3%20%20%20%20%20Test%20Motivators)

[3. Itens de Teste-Alvo](http://www.wthreex.com/rup/webtmpl/templates/test/rup_tstpln.htm#3.%20%20%20%20%20Target%20Test%20Items)

….

**Plano de Teste de <Iteração/Mestre>**

**1.**     **Introdução**

**1.1**     **Finalidade**

A finalidade do Plano de Teste de Iteração é reunir todas as informações necessárias para planejar e controlar o esforço de teste referente a uma iteração específica. Ele descreve a abordagem dada ao teste do software e é o plano de nível superior gerado e usado pelos gerentes para coordenar o esforço de teste.

Este documento *Plano de Teste* referente ao Gerador Burndown suporta os seguintes objetivos:

• Identifica os itens que devem ser inspecionados pelos testes.

• Identifica a motivação e as idéias subjacentes às áreas de teste a serem abrangidas.

• Descreve a abordagem de teste que será usada.

• Identifica os recursos necessários e fornece uma estimativa dos esforços de teste.

• Lista os elementos liberados do projeto de teste.

**1.2**     **Escopo**

O Burndown passará pelos testes unidade. Os testes unidade vão lidar com a qualidade funcional, abordando tanto os testes de caixa preta quanto os testes de caixa branca. Para a execução dos testes serão utilizadas máquinas mais semelhantes, em termos de hardware, a fim de garantir a previsibilidade de performance e compatibilidade.

Os testes mais críticos serão os testes das classes de negócio BurndownBO e SprintBO, uma vez que elas possuem um grau de dependência muito forte das classes de objeto.

**1.3**     **Público-alvo**

Todas as pessoas relacionadas ao ciclo de vida do software devem ter conhecimento sobre este documento, principalmente as pessoas relacionadas a fase de requisitos, desenvolvimento, gerenciamento da qualidade e teste.

**1.4**     **Terminologia e Acrônimos do Documento**

N/A

**1.5**     **Referências**

*[Esta subseção fornece uma lista dos documentos mencionados em qualquer outra parte do Plano de Teste. Identifique cada documento por título, número da versão (ou do relatório, se aplicável), data, organização de publicação ou autor original. Evite listar documentos que exercem influência no contexto mas que não foram mencionados diretamente. Especifique as fontes a partir das quais as "versões oficiais" das referências podem ser obtidas como, por exemplo, nomes UNC de intranet ou códigos de referência de documento. Essas informações podem ser fornecidas por um anexo ou outro documento.]*

**2.**     **Missão de Avaliação e Motivação dos Testes**

**2.1**     **Fundamentos**

Garantir que com os testes das classes de negócio possa-se gerar.

**2.2**     **Missão de Avaliação**

Localizar problemas importantes, avaliar os riscos de qualidades perceptíveis e informar sobre os riscos perceptíveis do projeto.

**2.3**     **Motivadores dos Testes**

Detectar riscos de qualidade, riscos técnicos, de projeto, casos de uso, requisitos funcionais e não funcionais.

**3.**     **Itens de Teste-Alvo**

A listagem abaixo identifica os itens de software, de hardware e elementos de suporte do produto que foram identificados como objetivos dos testes. Esta lista representa os itens que serão testados.

* SprintBO.java

Métodos:

* + calculaTotalHoras
  + calculaTotalPontos
* BurnDownBO.java

Métodos:

* + calculaEixosXYHoras
  + eixosXYPontos
  + geraBurndownHoras
  + geraBurndownPontos

Artefatos complementares que infuenciam nos testes principais:

* Sprint.java
* Estoria.java
* Historico.java
* ItemHistorico.java
* DataUtil.java

**4.**     **Resumo dos Testes Planejados**

**4.1**     **Resumo das Inclusões dos Testes**

|  |  |
| --- | --- |
| Objetivo do tipo de teste: | Cobertura de comando e de desvios |
| Técnica | Teste funcional de caixa branca |
| Critério de êxito | O resultado obtido ser equivalente ao esperado |
| Condiderações especiais | N/A |

**5.**     **Abordagem dos Testes**

Os testes serão executados de forma automatizada com o auxilo da ferramenta JUnit utilizando a IDE Java versão INDIGO.

**5.1**     **Catálogos Iniciais de Idéias de Teste e Outras Fontes de Referência**

*[Forneça uma listagem dos recursos existentes que serão consultados para estimular a identificação e a seleção de testes específicos a serem conduzidos. É fornecido um Catálogo de Idéias de Teste de exemplo na seção de exemplos do RUP.]*

**5.2**     **Tipos e Técnicas de Teste**

*5.2.1*     *Teste de Funcionamento*

*[O teste de funcionamento do objetivo do teste deve concentrar-se em todos os requisitos de teste que possam ser diretamente associados a casos de uso ou funções e regras de negócios. Esse teste tem por fim verificar a adequada aceitação, processamento e recuperação dos dados, e a implementação apropriada das regras de negócios. Esse tipo de teste baseia-se em técnicas de caixa preta; ou seja, verificar o aplicativo e seus processos internos interagindo com o aplicativo através da Interface Gráfica do Usuário (GUI) e analisar a saída ou os resultados. A tabela a seguir identifica um resumo do teste recomendado para cada aplicativo.]*

|  |  |
| --- | --- |
| Objetivo da Técnica: | *[Experimentar a funcionalidade do objetivo do teste, incluindo a navegação, a entrada, o processamento e a recuperação de dados a fim de observar e registrar o comportamento-alvo.]* |
| Técnica: | *[Experimentar os recursos e fluxos ou funções de cada um dos cenários de caso de uso, utilizando dados válidos e inválidos para verificar se:*  *os resultados esperados ocorrerão quando forem usados dados válidos*  *as mensagens de erro ou de aviso apropriadas serão exibidas quando forem usados dados inválidos*  *cada regra de negócio será aplicada de forma adequada]* |
| Estratégias: | *[Descreva uma ou mais estratégias que podem ser usadas pela técnica para observar, de forma precisa, os resultados do teste. A estratégia combina elementos do método através do qual a observação pode ser feita e das características dos resultados específicos que indicam um provável êxito ou falha do teste. O ideal é que as estratégias sejam autoverificadas, permitindo que os testes automatizados façam uma avaliação inicial do êxito ou falha do teste. No entanto, tenha atenção para reduzir os riscos inerentes à determinação automática dos resultados.]* |
| Ferramentas Necessárias: | *[A técnica exige as seguintes ferramentas:*  *Ferramenta de Automação de Scripts de Teste*  *restaurador e reprodutor de imagem da configuração básica*  *ferramentas de backup e de recuperação*  *ferramentas de monitoramento de instalação (registro, disco rígido, CPU, memória etc)*  *ferramentas de geração de dados]* |
| Critérios de Êxito: | *[A técnica suporta o teste de:*  *todos os principais cenários de caso de uso*  *todos os principais recursos* |
| Considerações Especiais: | *[Identifique ou descreva os itens ou problemas (internos ou externos) que exercem influência sobre a implementação e a execução do teste de funcionamento.]* |

**5.3**     **Casos de Teste**

*[Para cada uma das técnicas descritas acima liste os casos de teste que serão executados. Um caso de testes funcional deve conter as entradas, as saídas esperadas (valores ou mensagens de erro) e o nome do caso de uso relacionado ao caso de teste. Casos de testes estruturais devem indicar as entradas, as saídas esperadas e o método ou classe a que se destina testar. ]*

**6.**     **Critérios de Entrada e de Saída**

**6.1**     **Plano de Teste SprintBO ( Calcula total hora)**

*6.1.1*     *Critérios de Entrada de Plano de Teste SprintBO*

*Para teste caixa preta invalida : sprint vazia*

*Para teste caixa preta válida: Sprint preenchida*

*para testa caixa branca:*

Estoria est1 = new Estoria();

est1.setTempoEstimado(40);

est1.setCodEstoria(1);

est1.setQtdePontos(5);

Definir histórico da sprint

ItemHistorico item = new ItemHistorico();

item.setCodEstoria(1);

item.setTempoGasto(34);

item.setData(DataUtil.converteStringParaDate(DT\_INICIO\_SPRINT));

*6.1.1.2*     *Critérios de Saída de Plano de Teste*

*Para teste caixa preta invalida : sprint invalida*

*Para teste caixa preta válida: resultado igual a 140*

*para testa caixa branca: resultado igual a 6*

**6.1.2**     **Plano de Teste SprintBO(Calcula total ponto)**

*6.1.2.1*     *Critérios de Entrada de Plano de Teste Sprint*

*Para teste caixa preta invalida : sprint vazia*

*Para teste caixa preta válida: Sprint preenchida*

*Para teste caixa branca: COM COBERTURA TOTAL DE DESVIOS E DE COMANDOS.*

Estoria est1 = new Estoria();

est1.setTempoEstimado(40);

est1.setCodEstoria(1);

est1.setQtdePontos(5);

Definir histórico da sprint

ItemHistorico item = new ItemHistorico();

item.setCodEstoria(1);

item.setTempoGasto(34);

item.setData(DataUtil.converteStringParaDate(DT\_INICIO\_SPRINT));

*6.1.2.2*     *Critérios de Saída de Plano de Teste*

*Para teste caixa preta invalida : mensagem de erro : Sprint invalida*

*Para teste caixa preta válida: resultado igual a 17*

*para testa caixa branca: resultado igual a 5*

**6.1.3**     **Plano de Teste BurndownBO(calculaEixoXYHoras)**

*6.1.4.1*     *Critérios de Entrada de Plano de Teste (calculaEixo XYHoras)*

*Teste de caixa preta*

*Teste de caixa branca*

*Classe de equivalência válida:*

* *COM COBERTURA TOTAL DE COMANDO*

A Sprint deverá ter no mínimo 2 itens de histórico e 2 estórias de usuário

e setar a variavel quantidadeDias = 3

Entradas:

sprint.getQtdeDias(3);

Lista de Estórias da Sprint

Estoria est1 = new Estoria();

est1.setTempoEstimado(40);

est1.setCodEstoria(1);

est1.setQtdePontos(5);

Estoria est2 = new Estoria();

est2.setTempoEstimado(32);

est2.setCodEstoria(2);

est2.setQtdePontos(3);

Lista de itens de histórico da sprint

ItemHistorico item = new ItemHistorico();

item.setCodEstoria(1);

item.setTempoGasto(34);

item.setData(DataUtil.converteStringParaDate(DT\_INICIO\_SPRINT));

ItemHistorico item2 = new ItemHistorico();

item2.setCodEstoria(2);

item2.setTempoGasto(18);

item2.setData(DataUtil.converteStringParaDate("09/09/2014"));

COM COBERTURA TOTAL DE DESVIO

Entradas válidas:

* Sprint vazia - Sem estórias e sem histórico
* A Sprint deverá ter no mínimo 2 itens de histórico e 2 estórias de usuário

e setar a variavel quantidadeDias = 3

Entradas:

sprint.getQtdeDias(3);

Lista de Estórias da Sprint

Estoria est1 = new Estoria();

est1.setTempoEstimado(40);

est1.setCodEstoria(1);

est1.setQtdePontos(5);

Estoria est2 = new Estoria();

est2.setTempoEstimado(32);

est2.setCodEstoria(2);

est2.setQtdePontos(3);

Lista de itens de histórico da sprint

ItemHistorico item = new ItemHistorico();

item.setCodEstoria(1);

item.setTempoGasto(34);

item.setData(DataUtil.converteStringParaDate(DT\_INICIO\_SPRINT));

ItemHistorico item2 = new ItemHistorico();

item2.setCodEstoria(2);

item2.setTempoGasto(18);

item2.setData(DataUtil.converteStringParaDate("09/09/2014"));

*6.1.4.2*     *Critérios de Saída de Plano de Teste*

*Teste de caixa branca*

*Classe de equivalência válida*

*COM COBERTURA TOTAL DE COMANDO*

*Saida esperada: Map<date,Double> com os seguintes valores*

*08/09/2014, 34*

09/09/2014, 18

*COM COBERTURA DE DESVIO:*

* *Saída esperada : Map<date,Double> vazia;*
* *Saida esperada: Map<date,Double> com os seguintes valores*

*08/09/2014, 34*

09/09/2014, 18

**6.1.4**     **Plano de Teste BurndownBO(eixoXY Pontos)**

*6.1.3.1*     *Critérios de Entrada de Plano de Teste (EixoXY Pontos)*

*Teste de caixa Branca*

*Classe de equivalência válida: A Sprint com mais de uma estoria e mais de um item de estoria.*

*Entradas válidas:*

Lista de Estórias da Sprint

Estoria est1 = new Estoria();

est1.setTempoEstimado(40);

est1.setCodEstoria(1);

est1.setQtdePontos(5);

Estoria est2 = new Estoria();

est2.setTempoEstimado(32);

est2.setCodEstoria(2);

est2.setQtdePontos(3);

Lista de itens de histórico da sprint

ItemHistorico item = new ItemHistorico();

item.setCodEstoria(1);

item.setTempoGasto(34);

item.setData(DataUtil.converteStringParaDate(DT\_INICIO\_SPRINT));

ItemHistorico item2 = new ItemHistorico();

item2.setCodEstoria(2);

item2.setTempoGasto(18);

item2.setData(DataUtil.converteStringParaDate("09/09/2014"));

*Teste de caixa Preta*

*Classe de equivalência invalida: A Sprint vazia*

*6.1.3.2*    *Critérios de Saída de Plano de Teste (EixoXY Pontos)*

*Teste de caixa branca*

*Classe de equivalência válida: Data da estoria e o numero de pontos da estoria,*

*map<date,Double> = 08/09/2014, 5*

*09/09/2014, 3*

*Teste de caixa Preta*

*Classe de equivalência invalida: Mapa vazio*

**6.1.5**    **Plano de Teste BurndownBO(geraBurndownHoras)**

*6.1.5.1*     *Critérios de Entrada de Plano de Teste (geraBurndownHoras)*

Teste de caixa branca

COM COBERTURA TOTAL DE DESVIOS E DE COMANDO

Classe de equivalência válida: sprint com mais de uma estória de usuário e com mais de um itemHistórico

Entradas Válidas:

* A Sprint deverá ter no mínimo 2 itens de histórico e 2 estórias de usuário

Entradas:

Lista de Estórias da Sprint

Estoria est1 = new Estoria();

est1.setTempoEstimado(40);

est1.setCodEstoria(1);

est1.setQtdePontos(5);

Estoria est2 = new Estoria();

est2.setTempoEstimado(32);

est2.setCodEstoria(2);

est2.setQtdePontos(3);

Lista de itens de histórico da sprint

ItemHistorico item = new ItemHistorico();

item.setCodEstoria(1);

item.setTempoGasto(34);

item.setData(DataUtil.converteStringParaDate(DT\_INICIO\_SPRINT));

ItemHistorico item2 = new ItemHistorico();

item2.setCodEstoria(2);

item2.setTempoGasto(18);

item2.setData(DataUtil.converteStringParaDate("09/09/2014"));

*6.1.4.2*     *Critérios de Saída de Plano de Teste*

*teste de caixa branca*

*Gráfico LineChart burndow por horas para dois dias de estórias*

**6.1.6**    **Plano de Teste BurndownBO(geraBurndowPontos)**

*6.1.6.1*     *Critérios de Entrada de Plano de Teste (geraBurndownPontos)*

Teste de caixa branca

COM COBERTURA TOTAL DE DESVIOS E DE COMANDO

Classe de equivalência válida: sprint com mais de uma estória de usuário e com mais de um itemHistórico

Entradas Válidas:

* A Sprint deverá ter no mínimo 2 itens de histórico e 2 estórias de usuário

Entradas:

Lista de Estórias da Sprint

Estoria est1 = new Estoria();

est1.setTempoEstimado(40);

est1.setCodEstoria(1);

est1.setQtdePontos(5);

Estoria est2 = new Estoria();

est2.setTempoEstimado(32);

est2.setCodEstoria(2);

est2.setQtdePontos(3);

Lista de itens de histórico da sprint

ItemHistorico item = new ItemHistorico();

item.setCodEstoria(1);

item.setTempoGasto(34);

item.setData(DataUtil.converteStringParaDate(DT\_INICIO\_SPRINT));

ItemHistorico item2 = new ItemHistorico();

item2.setCodEstoria(2);

item2.setTempoGasto(18);

item2.setData(DataUtil.converteStringParaDate("09/09/2014"));

*6.1.6.2*     *Critérios de Saída de Plano de Teste*

*teste de caixa branca*

*Gráfico LineChart burndow por horas para dois dias de estórias*

**7.**     **Produtos Liberados**

Com os casos de teste listados nas seções anteriores serão liberados os seguintes artefatos:

* SprintBO
* Sprint
* Estoria
* Historico
* ItemHistorico
* DataUtil
* BurnDownBO

**7.1**     **Sumários de Avaliação de Testes**

*[Forneça um breve resumo da forma e do conteúdo dos sumários de avaliação de testes e indique com que freqüência eles serão gerados.]*

**7.2**     **Geração de Relatórios sobre Cobertura de Teste**

*[Forneça um breve resumo da forma e do conteúdo dos relatórios usados para medir a extensão do teste e indique com que freqüência eles serão gerados Forneça uma indicação referente ao método e às ferramentas usados para registrar, medir e reportar a extensão do teste.]*

**7.3**    **Registros de Incidentes e Solicitações de Mudança**

*[Forneça um breve resumo do método e das ferramentas usados para registrar, rastrear e gerenciar incidentes dos testes, as solicitações de mudança associadas e seus status.]*

**7.4**     **Conjunto de Testes de Regressão e Scripts de Teste de Suporte**

*[Forneça um breve resumo dos recursos dos testes que serão liberados para permitir testes de regressão contínuos dos builds subseqüentes do produto, a fim de ajudar a detectar as regressões na qualidade do produto.]*

**7.5**     **Produtos de Trabalho Adicionais**

*[Nesta seção, identifique os produtos de trabalho que são opcionais ou os que não deverão ser usados para medir ou avaliar a execução bem-sucedida do Plano de Teste.]*

*7.5.1*     *Resultados Detalhados dos Testes*

*[Trata-se de um conjunto de planilhas do Microsoft Excel relacionando os resultados determinados para cada caso de teste ou refere-se ao repositório dos registros de testes e dos resultados determinados mantidos por um produto de teste especializado.]*

*7.5.2*     *Scripts de Teste Funcionais Automatizados Adicionais*

*Estes scripts consistem em um conjunto dos arquivos de código-fonte dos scripts de teste automatizados ou no repositório do código-fonte e dos executáveis compilados referentes aos scripts de teste mantidos pelo produto de automação de testes.]*

*7.5.3*     *Guia de Teste*

*[O Guia de Teste abrange um amplo conjunto de categorias incluindo Catálogos de Idéias de Testes, Orientações de Práticas Adequadas, Padrões de Teste, Modelos de Erros e de Falhas, Padrões de Design de Automação etc.]*

*7.5.4*     *Matrizes de Rastreabilidade*

*[Utilizando uma ferramenta como o Rational RequisistePro ou o Microsoft Excel, forneça uma ou mais matrizes de relacionamentos de rastreabilidade entre os itens rastreados.]*