Universidade Federal de Goiás, Instituto de Informática
Bacharelado em Sistemas de Informação

Projeto de Teste de Software - ***Hangman Java***

INF0303 - Teste de Software - 2022.1

Aluno(s): Evaldo F. Lima,
Walysson C. Paiva.

Goiânia, 2022.

Hangman Corp.

Departamento de Produtos - Time de Tecnologia e Desenvolvimento,
Política(s) de Teste e Prática(s) de Teste Organizacional.

Política(s) de Teste

Objetivos de testagem: a testagem tem como objetivo assegurar as funcionalidades e a confiabilidade de sistemas desenvolvidos pela Tied Corp, garantindo o funcionamento das regras de negócio de seus jogos.

Processo de teste: os testes são conduzidos de acordo com a ISO/IEC/IEEE 29119-2, seguindo variações que atendam às regras de negócio. O processo de teste inicia-se no planejamento, seguindo-se de: design dos casos de testes, baseando-se principalmente na execução dos testes, registro de incidentes, relatórios e registros de chamados. Os projetos de desenvolvimento incluem um projeto de testes, iniciado ao mesmo tempo. Ao final do desenvolvimento têm-se testes mais rigorosos, validando o funcionamento do jogo como um todo, através de testes de caixa-preta e de caixa-branca.

Organização dos testes: os projetos são alocados com testadores, assim, vários testadores atuam em um projeto e relatam individualmente o resultado dos mesmos para seu gerente, a fim de validar consistências e inconsistências frente a diferentes técnicas de teste. Os testadores possuem equipamento e documentação necessária para realização destes.

Treinamento: a rigor, para todos os testadores é necessário possuir diploma de conclusão do ensino superior na área de tecnologia da informação, entre eles os cursos de Sistemas de Informação, Engenharia de Software, Ciências da Computação ou correlatos, e certificação no setor de Teste de Software.

Código de ética: todos os testadores devem aderir ao código de ética da companhia, mantendo em segredo informações referentes ao status do desenvolvimento dos jogos.

Padrões: a companhia segue os padrões estabelecidos pela ISO/IEC 20246 (quando se trata de testes estáticos), ISO/IEC/IEEE 29119-2 (i.e. define processo de projeto de teste e processo de implementação de testes), ISO/IEC/IEEE 29119-3 (i.e. define um conjunto de padrões internacionalmente aceitos para teste de software, que pode ser usado por qualquer organização ao realizar qualquer forma de teste de software) e ISO/IEC/IEEE 29119-4 (i.e. define técnicas de projeto de teste de software, conhecidas como técnicas de projeto de caso de teste ou métodos de teste) que podem ser usadas dentro do que é definido na ISO/IEC/IEEE 29119-2.

Políticas relevantes relacionadas: o ciclo de desenvolvimento de software é primariamente pautado nos padrões acima definidos com o adendo de seguirem as boas práticas de teste de

software disponíveis no repositório de documentos, acessível pelos colaboradores através da rede privada da companhia.

Aprimoramento do processo de teste e determinação de valor: obrigatoriamente, em todos os projetos, deve-se prover relatórios com dados coletados em diversos pontos do ciclo de desenvolvimento e testes a fim de aprimorar o processo, boas práticas e normas seguidas pelos testadores. Assim, ao ter embasamento necessário, são estudados e estabelecidos pontos de melhoria nos pontos identificados.

Armazenamento e reuso de artefatos de teste: obrigatoriamente, deve-se manter registro de todos os processos, boas práticas, normas e artefatos de teste à disposição, com a companhia provendo repositórios em sua rede privada com os devidos cuidados em relação à segurança da informação.

Publicado por: Evaldo Felipe (Chefe de Testes) e Walysson Paiva (Testador Sênior).

Aprovado por: Oscar Nogueira (Chief Technology Officer - CTO, Hangman Corp.), Lucas Esquí Walker (Gerente de Projeto - Hangman Corp.), Will Lumen (Chefe de Desenvolvimento) e Lampard White (Agilista - Hangman Corp.)

Prática(s) de Teste Organizacional

Critérios de entrada: em todos os níveis de teste, previamente, ao iniciar cada tipo de teste recomenda-se a construção de um Plano de Teste do Projeto, assinado pelo Gerente de Projeto, Gerente de Teste e Chefe de Teste. Em relação ao Plano de Teste do Projeto, têm-se alguns dos itens necessários para os critérios de entrada nos testes de sistema, unidade e integração, estático, aceitação, verificação de produção, desastres, penetração e de desempenho: os requisitos de cada teste, definições do ambiente de teste, verificação de abrangimento nos estágios anteriores de teste, os casos de teste devem ser projetados, rastreados os requisitos é assinado pelas partes interessadas.

Critérios de saída: em testes estáticos, deve-se produzir e atentar aos requisitos do projeto, projeto e inspeções de código. Adicionalmente, para todos os níveis de teste, deve-se atentar que é necessário a execução e aprovação dos testes de software com a necessidade de abranger aos testes que ainda passaram pela aprovação das partes interessadas mas não foram abrangidos.

Critérios de conclusão do teste: para se ter como concluídos, o projeto deve ter 100% cobertura em relação a testes, além disso, deve obrigatoriamente solucionar e realizar o reteste de erros de média e alta gravidade previamente encontrados. Ademais, os erros pendentes devem ser informados às partes interessadas e junto a isso um prazo mínimo para correção de pendências.

Escopo de teste: as práticas de teste organizacional devem ser aplicadas a todos os testes realizados na Hangman Corp. e estender-se para suas subsidiárias.

Gerenciamento de configuração de ativos de teste: dado que os ativos baseados em documentos são armazenados na rede interna da equipe, têm-se o com controle de versão manual aplicado, a fim de garantir que cada versão de documento exclusiva seja capturada e armazenada em backup. Assim, ferramentas de gerenciamento de requisitos, testes e automação possuem o gerenciamento de configuração integrado para todos os ativos.

Gerenciamento de risco: as avaliações de risco são conduzidas durante a iteração zero e revisitadas durante as reuniões de planejamento de iteração subsequentes, assim, para todas as equipes de desenvolvimento ágeis, deve-se realizar a avaliação de risco do produto e do projeto com base nos processo de gerenciamento de risco relacionado a testes, disponíveis na rede interna da equipe.

Automação de teste, ferramentas, gerenciamento de defeitos e incidentes: a priori, as equipes ágeis devem preparar um plano de teste de versão de uma página, os planos de teste de iteração, os casos de teste para recursos de alto risco e um relatório com o resumo de teste de uma página no final da versão. Dessa forma, os testes são automatizados sempre que possível por meio de uma estrutura de automação central da Hangman Corp. e esses scripts armazenados no sistema de controle de origem central, assim, as sessões de teste exploratório são capturadas por meio de ferramenta de gravação aprovada, com as evidências da execução armazenadas na ferramenta central de gerenciamento de teste. Já os testes manuais (i.e. no caso de serem necessários) também serão armazenados na ferramenta central de gerenciamento de testes. Posto isso, todos os testes manuais e automatizados devem ser rastreados para histórias de usuários e os defeitos armazenados no sistema de gerenciamento de tarefas e requisitos, assim, sendo descritos como defeitos vinculados a histórias de usuários ou comentários em histórias de usuários usando as práticas padrão de gerenciamento de defeitos, histórias e modelos previamente configuradas na ferramenta.

Documentação e relatórios de teste: a rigor, deve-se atentar ao produzir a documentação e os relatórios de testes pautados na ISO/IEC/IEEE 29119-3 em todos os casos de aplicação de testes dinâmicos e o ISO/IEC 20246 em todos os casos de aplicação de testes estáticos.

Grau de independência e Seleção e priorização de teste: a seleção e priorização de teste será pautada na avaliação de risco de iteração realizada durante cada sessão de planejamento de iteração, E, a fim de garantir independência na prática de teste, os testadores devem seguir a hierarquia estabelecida pela companhia e reportar ao seu gerente de projeto e ao chefe de teste.

Técnicas de projeto de teste: as técnicas devem ser implementadas para cobrir os requisitos e decrescer riscos, em adendo, adaptações impostas na norma definida pela companhia deve ser aprovada e assinada pelo Chefe de Testes e CTO. Adicionalmente, as especificações das técnicas e estrutura de execução definidas pela ISO/IEC/IEEE 29119-4.

Técnicas de teste, seus tipos e níveis: com a utilização das técnicas de teste da ISO/IEC/IEEE 29119-4, adaptadas para garantir que os testes permaneçam enxutos, durante a iteração zero, e em sessões de planejamento subsequentes, serão escolhidos os tipos de teste que aferem se as histórias dos usuários atendem aos requisitos. Com base nisso, os níveis de

teste aplicados são: testes de unidade, integração, história e sistema, integração de sistema, verificação de produção e aceitação. Por fim, os tipos de teste normalmente aplicados são testes funcionais, de desempenho, acessibilidade, penetração e desastre/recuperação.

Diretrizes para desvio das Práticas de Teste Organizacionais: segue-se o fato de que cada equipe deve usar os mesmos modelos para gerenciamento de histórias e requisitos, de defeitos e teste no conjunto da ferramenta, assim, estes podem ser adaptados conforme exigido pela equipe.

Ambiente de teste e métricas coletadas: em relação às métricas, deve-se relatar durantes os testes as métricas que dizem respeito à: número de erros em produção normalizado pelo tamanho do projeto(densidade de defeitos em produção), porcentagem de defeitos que surgiram por não serem abordados em uma etapa de teste anterior(vazamento de defeitos) e o número esperado e real de testes projetados e executados por dia. Já em relação ao ambiente de teste, com os dados sensíveis previamente não identificáveis, todos os testes de unidade e integração no ambiente de desenvolvimento e os testes de história, sistema e integração devem ser realizados no ambiente de teste do sistema. Além disso, testes de penetração e verificação da produção são realizados em ambiente de produção, porém, teste de desempenho, acessibilidade, penetração, desastre e recuperação são realizados em um ambiente de pré-produção.

Reteste e teste de regressão: no caso do reteste, após corrigir-se os defeitos encontrados em casos de teste anteriores, os casos de teste devem ser executados novamente para verificar se foram de fato abrangidos ou ainda estão defeituosos. Em relação ao teste de regressão, os casos de teste devem ser obrigatoriamente executados após os níveis de Teste de Sistema e Teste de aceitação, com o nível de Teste de Sistema ficando a cargo da decisão de necessidade do Chefe de Teste, do Agilista e do Gerente de Projeto.

Descrição geral do software

O software trata-se de um jogo, denominado Hangman Game(i.e. em poruguês do Brasil é equivalente à Jogo da Forca), desenvolvido predominantemente na linguagem de programação Java e baseado em entradas no prompt de comando. O jogo é jogado entre dois jogadores, e neste caso o jogador 1 coloca a palavra através do argumento da linha de comando e configura o jogo para o jogador 2. Assim, o jogador 2 adivinha a resposta em um número determinado de tentativas e se o jogador 2 adivinhou a palavra certa dentro das tentativas limitadas, ele ganha, senão é dado como perdedor.

Features em Gherkin

```
@tag
Feature: Validacao das regras do jogo da forca
  Como usuario eu quero que o jogo funcione corretamente.

  @tag1
  Scenario Outline: Validar cenarios de derrota
    Given O jogo iniciou
    And Eu digitei a palavra "<palavra>"
    When Eu digitei a letra 1 "<letra1>"
    And Eu digitei a letra 2 "<letra2>"
    And Eu digitei a letra 3 "<letra3>"
    And Eu digitei a letra 4 "<letra4>"
    And Eu digitei a letra 5 "<letra5>"
    And Eu digitei a letra 6 "<letra6>"
    Then Eu perdi o jogo

    Examples:
      | palavra | letra1 | letra2 | letra3 | letra4 | letra5 | letra6 |
      | banana  | e      | c      | d      | j      | t      | f      |
      | alface  | t      | d      | g      | i      | j      | b      |
      | repolho | a      | f      | c      | d      | u      | b      |
      | laranja | e      | b      | c      | d      | i      | o      |
      | mamao   | u      | b      | i      | d      | e      | j      |
      | pera    | q      | c      | i      | b      | w      | o      |

  @tag2
  Scenario Outline: Validar cenarios de vitoria
    Given O jogo iniciou
    And A palavra para adivinhar e "<palavra>"
    When Eu digitei a letra correta 1 "<letra1>"
    And Eu digitei a letra correta 2 "<letra2>"
    And Eu digitei a letra correta 3 "<letra3>"
    And Eu digitei a letra correta 4 "<letra4>"
    And Eu digitei a letra correta 5 "<letra5>"
    Then Eu ganhei o jogo

    Examples:
      | palavra | letra1 | letra2 | letra3 | letra4 | letra5 |
      | banana  | b      | n      | a      | w      | u      |
      | alface  | a      | l      | f      | c      | e      |
      | laranja | l      | r      | a      | n      | j      |
```

Plano(s) de Teste

Hangman Corp.
Observação: a versão mais recente do plano de teste faz-se necessário estar disponível para a equipe ágil através da rede interna.

Plano de Teste para: Jogo Hangman Java

Iteração: 1

Contexto, itens de teste e escopo: Hangman Java, iteração 1. Dado que as histórias são capturadas na ferramenta de gerenciamento de tarefas e requisitos da equipe, inclui-se todos os níveis e tipos de teste para as histórias em uma sprint definida.

Equipe e comunicação: faz-se necessário que a equipe ágil reporta-se ao líder de entrega, enquanto todos os desenvolvedores também se reportam independentemente ao gerente de projeto, ao chefe de desenvolvimento, aos testadores e ao chefe de testes. Assim, em cada iteração é realizada pela equipe ágil que se compõem em um líder de entrega, desenvolvedores, analista de negócios, representantes de usuários ou partes interessadas, testadores e um proprietário do produto.

Partes interessadas: O Gerente de Projeto e o Agilista são as partes interessadas no que diz respeito à aprovação. Assim, somente chega-se ao chefe de desenvolvimento e chefe de teste quando não se tem um consenso na equipe ágil.

Riscos: cada risco está descrito e relacionado nos cartões de história com renomada relevância para o contexto. Posto isso, temos que uma variável de risco atualmente se dá pela falta de acesso da equipe ágil aos dados disponibilizados pelo suporte da aplicação.

Estratégia de Teste:

- Certificar-se que nenhum defeito com gravidade 1 (grave) ou 2 (médio) permaneça pendente quando uma história for marcada como “Concluída”;
- Defeitos que restarem devem ser preenchidos como novas histórias e colocadas no “Backlog” de desenvolvimento.
- Histórias de maior risco exigem teste com maior completude, e em caso de histórias de baixo risco, recomenda-se a utilização de técnicas de cobertura de combinação minimizada;
- Teste de regressão em quaisquer recursos que tenham sido alterados desde as iterações anteriores e todo o sistema antes da reunião de demonstração, fazendo o uso do conjunto de testes automatizados;

- Criar testes automatizados de unidade e integração com base em histórias antes do início da codificação, cuja a técnica é nomeada de TDD (i.e. Test Driven Development);
- Compartilhar, em reuniões diárias, o progresso dos testes e principais bloqueadores que impedem a completude da avaliação;
- Para que estejam disponíveis para reteste e teste de regressão, armazene planilhas e testes de sessão na ferramenta central de gerenciamento de teste e todos os scripts utilizados nos testes automatizados no repositório de código central;
- Certifique-se de que os clientes estejam envolvidos e conduzam idealmente os testes de aceitação do usuário, com o máximo de testes automatizados possíveis;
- Retorne ao backlog quaisquer histórias de usuários que não possam ser alcançadas até o fim da iteração, sendo incluso quaisquer dívidas técnicas acumuladas que não possam ser resolvidas na iteração atual. Dessa maneira, no planejamento de liberação e iteração, estime o esforço de teste e desenvolvimento com base na velocidade média;
- Por fim, todos os outros detalhes da estratégia estão de acordo com as Práticas de Teste Organizacional e Carta de Equipe Ágil.

Casos de Teste

Cenário 1: validar cenários de vitória no jogo Hangman

Em um cenário de vitória no jogo Hangman, um jogador é declarado vencedor quando acerta todas as letras de uma palavra sem que as tentativas(i.e. quantidade de tentativas é sempre menor ou igual que a quantidade de letras da palavra) se esgotem.

Assim, o sistema recebe como entrada uma letra por vez, verifica se a palavra contém a letra e valida o acerto.

Por exemplo,

Palavra	Letra 1	Letra 2	Letra 3	Letra 4	Letra 5	Letra 6	Resultado	Exceção
banana	**b**	**n**	**a**	**-**	**-**	**-**	venceu	não
alface	**a**	**l**	**f**	**c**	**e**	**-**	venceu	não
laranja	**l**	**r**	**a**	**n**	**j**	**-**	venceu	não

Cenário 2: validar cenários de derrota no jogo Hangman

Em um cenário de vitória no jogo Hangman, um jogador é declarado perdedor quando não acerta todas as letras de uma palavra sem que as tentativas(i.e. quantidade de tentativas é sempre menor ou igual que a quantidade de letras da palavra) se esgotem.

Assim, o sistema recebe como entrada uma letra por vez, verifica se a palavra contém a letra e valida o acerto.

Por exemplo,

Palavra	Letra 1	Letra 2	Letra 3	Letra 4	Letra 5	Letra 6	Resultado	Exceção
banana	**e**	**c**	**d**	**e**	**t**	**j**	perdeu	não
alface	**t**	**d**	**g**	**i**	**j**	**b**	perdeu	não
repolho	**a**	**f**	**c**	**d**	**u**	**b**	perdeu	não
laranja	**e**	**b**	**c**	**d**	**i**	**o**	perdeu	não
mamao	**u**	**b**	**i**	**d**	**e**	**j**	perdeu	não
pera	**q**	**c**	**i**	**b**	**w**	**o**	perdeu	não
melancia	**r**	**u**	**d**	**b**	**q**	***1***	perdeu	***sim***
melancia	**r**	**u**	**d**	**b**	**q**		perdeu	***sim***
melancia	**r**	**u**	**d**	**b**	**q**	***q***	perdeu	***sim***

Relatórios de Teste - Relatório de Status do Teste

Hangman Corp.
Observação: dado as obrigações contratuais com o cliente, a versão mais recente do relatório de status de teste faz-se necessário estar disponível para a equipe ágil através da rede interna, em um endereço eletrônico definido na aba de “Informações do Projeto”.

Relatório de Status de Teste de Iteração para: Jogo Hangman Java - StatusPoint

Período de relatório: Iteração 1, Sprint 1

Progresso em relação ao plano de teste:

- Os testes do jogo foram feitos para todas as possíveis histórias descritas na análise valor limite;
- Os testes alcançaram a cobertura de 100% em classes, 92.9% em métodos e 95.2% em linhas, utilizando a ferramenta JUnit para testes unitários;
- Os testes alcançaram a cobertura de 94% em linhas e 69% em mutações, utilizando a ferramenta PIT Test para testes de mutação;

Fatores que bloqueiam o progresso nos testes: assim como descrito acima a porcentagem de cobertura para o sistema não alcançou em alguns pontos o total de 100% de cobertura. Posto isso, ao analisarmos, temos embasamento técnico para inferir sobre alguns pontos a respeito de algumas possíveis correções no sistema. Sendo assim, listam-se algumas histórias:

- Para a classe Game, as funções que envolvem o Guess tem 3 formas de validação de entrada de dados, sendo: uma para quando o input é vazio, uma para quando não é letra e uma para quando é repetido. Assim, em uma abordagem resolutiva em que uma dessas sobreviva nos teste de mutação, seria necessário validação de entradas complexas que averiguasse um e evitasse todos os outros;
- Na classe Hangman, em seu método main, não se faz possível testar o que envolve a função de exit no Java 8, dado o fato de que ela interrompe todo o fluxo, inclusive o fluxo de testes;

Medidas de Teste: serão desenvolvidas novas possíveis entradas dos usuários, que deverão ser também acordadas com a análise de valor limite.

Riscos novos e alterados: no presente momento, não foram encontrados novos riscos de alta gravidade, com os anteriores sendo encerrados por meio de correções.

Teste planejado: continuar seguindo os modelos definidos previamente no plano de teste a fim de garantir maior cobertura de testes no sistema ao decorrer da próxima sprint.

Observação: os relatórios de testes produzidos pelas ferramentas de automação de testes devem ser mantidos disponíveis a todos os interessados, neste caso, os testadores, o chefe de testes, o chefe de desenvolvimento, o agilista e aqueles que tiverem o aval da organização.