



UNIVERSIDADE FEDERAL DO CEARÁ
Campus de Quixadá

Plano de Medição de Qualidade

CITRA

Responsável: Ewaldo de Carvalho Bezerra Junior e Isânio Vitor Félix Araujo

ÍNDICE

1. INTRODUÇÃO	3
1.1. Descrição dos produtos a serem avaliados	3
1.2. Objetivos da avaliação	3
2. MÉTODO	4
2.1. Ambiente de avaliação	4
2.2. Procedimentos da Avaliação	5
2.3. Participantes (caso necessite / depende da característica escolhida)	6
2.4. Medidas de Software	6
2.4.1. Operabilidade	6
2.4.2. Adaptabilidade	7
2.4.3. Testabilidade	8
2.4.4. Tolerância a falhas	9
2.5. Procedimentos de Interpretação	10
3. REFERÊNCIAS	11

HISTÓRICO DE REVISÕES

Data	Versão	Descrição	Responsável
12/10/2022	1.0	Criação do documento.	Ewaldo Junior
17/10/2022	2.0	Objetivos da avaliação	Ewaldo J. e Isânio V.
19/10/2022	3.0	Consolidação das métricas	Ewaldo J. e Isânio V.
26/10/2022	4.0	Finalização do documento	Ewaldo J. e Isânio V.

1.INTRODUÇÃO

Este documento é destinado ao time de desenvolvimento e partes interessadas do projeto, visando a garantia e o controle de qualidade do software a ser desenvolvido.

1.1. Descrição dos produtos a serem avaliados

Este plano de medição tem o intuito de fornecer as principais métricas avaliadas na aplicação CITRA. Para o projeto, serão avaliados as seguintes funcionalidades planejadas e desenvolvidas pelo sistema:

- Controle de Postagem de vagas de emprego
- Controle de Candidatura nas vagas de emprego
- Apresentação de informações acerca das vagas de emprego
- Filtragem de vagas de emprego

1.2. Objetivos da avaliação

O objetivo da avaliação é validar a usabilidade, segurança, manutenibilidade e portabilidade das principais funcionalidades do CITRA. Usuários representativos serão solicitados a realizar tarefas típicas da aplicação. Dentro de cada funcionalidade, serão avaliadas tais subcaracterísticas:

Qualidades que serão avaliadas:

- **Usabilidade:**
 - **Operacionalidade:** grau de facilidade com a qual um produto ou sistema é operado, ou controlado.
- **Portabilidade:**
 - **Adaptabilidade:** grau em que um produto ou sistema pode eficazmente e eficientemente ser adaptado para um hardware, software ou ambientes de uso diferentes ou em evolução.
- **Manutenibilidade:**
 - **Testabilidade:** grau de eficácia e eficiência com que critérios de teste podem ser estabelecidos e executados.
- **Confiabilidade:**
 - **Tolerância a falhas:** grau em que um sistema, produto ou componente opera como pretendido, apesar da presença de falhas de hardware ou software.

Analisar	Usabilidade
Para o propósito de	avaliar e melhorar o grau em que um produto ou sistema pode ser utilizado por usuários específicos para atingir metas especificadas com eficácia, eficiência e satisfação, em um determinado contexto.
Com respeito a	Operacionalidade
Do ponto de vista	dos usuários da aplicação
No contexto de	ambiente em execução da aplicação no celular.

Analisar	Portabilidade
Para o propósito de	avaliar o grau de eficácia e eficiência com a qual um sistema, produto ou componente pode ser transferido de um hardware, software ou ambientes de uso.
Com respeito a	Adaptabilidade
Do ponto de vista	dos usuários da aplicação
No contexto de	ambiente em execução da aplicação no celular.

Analisar	Manutenibilidade
Para o propósito de	entender e melhorar o grau de eficácia e eficiência com que um produto ou sistema pode ser modificado pela equipe de manutenção.
Com respeito a	Testabilidade
Do ponto de vista	dos desenvolvedores e testadores
No contexto de	ambiente em execução da aplicação de integração contínua (SonarQube).

Analisar	Confiabilidade
Para o propósito de	entender e controlar o grau em que um sistema, produto ou componente mantém, ao longo do tempo, um comportamento consistente com o esperado, sob as condições especificadas.
Com respeito a	Tolerância a falhas
Do ponto de vista	dos desenvolvedores da aplicação
No contexto de	ambiente em execução da aplicação de integração contínua (SonarQube).

2.MÉTODO

2.1. Ambiente de avaliação

Tarefas selecionadas para os participantes executarem:

- Criar uma postagem de vaga de emprego;
- Alterar dados da vaga de emprego;
- Deletar vaga de emprego;
- Se candidatar na vaga de emprego;
- Excluir candidatura;
- Visualizar as candidaturas feitas pelo usuário;
- Visualizar vagas de emprego disponíveis;
- Filtrar vagas de emprego;
- Fazer login;
- Fazer cadastro.

Ambientes que serão executados para análise das métricas:

- [Usabilidade] - será executado em ambiente real por meio de um dispositivo móvel, que poderá ser no celular dos desenvolvedores da aplicação, no qual, os modelos são: Samsung A21s com Android 12 e Samsung A11 com Android 10.
- [Manutenibilidade e Confiabilidade] - para analisar a testabilidade da aplicação, destaca-se o uso de SonarQube para realização da inspeção contínua da qualidade do código.
- [Portabilidade] - emuladores e/ou aparelhos para analisar o desempenho da aplicação em cenários diversos.

2.2. Procedimentos da Avaliação

Usabilidade: Primeiramente, será coletado dos usuários informações acerca de sua familiaridade com celulares e aplicações móveis. Logo em seguida, os participantes serão informados de que a usabilidade do CITRA seria avaliada para descobrir se o produto satisfaz suas necessidades. Os usuários serão informados de que não será um teste de suas habilidades e sim da usabilidade da aplicação. Depois, o avaliador explicará para os usuários como será a avaliação, apresentando o dispositivo a ser utilizado, o ambiente e informar que sua interação seria registrada. Por fim, o avaliador pedirá para os usuários responderem perguntas sobre o uso da aplicação, a fim de coletar dados de operacionalidade no CITRA.

Portabilidade: no CITRA será avaliada por meio da adaptabilidade da aplicação em outros aparelhos e/ou emuladores. A análise será avaliada conforme as portabilidades dos aparelhos de possíveis usuários. Para avaliar esta métrica, utilizaremos os emuladores do Android Studio para rodar nossa aplicação e mapearmos o nível de abrangência que o CITRA poderá executar. Ademais, estaremos utilizando também aparelhos físicos para ver a adaptabilidade da aplicação em celulares que estão rodando com outras aplicações em segundo plano, a fim de mensurar o quão adaptável o CITRA é nesses cenários.

Manutenibilidade: no CITRA será baseada na subcaracterística de testabilidade, e por meio desta premissa, será utilizado o SonarQube que é uma ferramenta para garantir a qualidade do código-fonte em desenvolvimento. Partindo da ferramenta utilizada, os desenvolvedores após implementarem os testes na aplicação, será utilizada como objetivo de garantir a testabilidade das funcionalidades a fim de analisar a cobertura de testes.

Confiabilidade: esta métrica será avaliada por meio da tolerância a falhas, então será utilizado novamente o SonarQube para avaliar como o sistema opera mesmo havendo falhas na aplicação. Esta métrica se comunica fortemente com a testabilidade, afinal, quando avaliarmos e encontrarmos baixa cobertura de testes, consequentemente poderá existir ocorrência de falhas, e assim, analisar e avaliar o comportamento dos componentes e o sistema na totalidade é importante para mensurar a execução da aplicação para evitar erros iminentes.

2.3. Participantes (caso necessite / depende da característica escolhida)

Os participantes entrariam na métrica de usabilidade. Partindo do pressuposto que não temos participantes conhecidos que trabalham no ramo de construção civil, iremos convidar para avaliar de idades e escolaridade diferentes. Pretendemos avaliar 3 pessoas:

- uma mulher de até 50 anos, com ensino médio completo e com pouca experiência em smartphones;
- um/uma jovem de até 25 anos, com ensino médio completo e com experiência em smartphones;
- um homem de até 50 anos, com ensino fundamental completo e pouquíssima experiência em smartphones.

2.4. Medidas de Software

2.4.1. Operabilidade

Nome	Taxa de coerente operacional nas funcionalidades da aplicação
Objetivo	O objetivo desta medida é para avaliar o entendimento dos usuários acerca das funcionalidades da aplicação.
Descrição	Quão fáceis de entender e intuitivas são os componentes e elementos da interface do usuário do CITRA?
Fórmula	<p>A= Quantidade de funções da aplicação onde o usuário compreende com sucesso as funcionalidades em um curto período de avaliação.</p> <p>B= Quantidade de funções da aplicação que serão selecionadas para realizar a avaliação de usabilidade.</p> <p>$X = (A/B) \times 100$</p> <p>X é o percentual de funcionalidades concluídas com sucesso de uso.</p>
Procedimento de coleta	<p>Como coletar: a operabilidade dos usuários será coletada a partir de seus registros na avaliação de usabilidade pelo avaliador.</p> <p>Como armazenar: por um documento de avaliação de usabilidade.</p>
Análise	Como analisar: analisar a quantidade de

	<p>funcionalidades concluídas e bem sucedidas pelo usuário durante a avaliação.</p> <p>Ações corretivas: caso ocorra uma longa demora na realização de alguma funcionalidade, o avaliador deve orientar o usuário para prosseguir para uma nova atividade.</p> <p>Onde armazenar: descrever no documento de avaliação de usabilidade com os registros dos percalços analisados.</p> <p>Como executar a revisão da efetividade do indicador: revisar o design do projeto, notificar ao UX Designer e logo após, o desenvolvedor das interfaces.</p>
Parâmetros	<p>Se $\leq 50\%$ indica um baixo grau de funcionalidades que foram bem operadas pelo usuário.</p> <p>Se $> 50\%$ e $< 80\%$ indica um grau mediano de funcionalidades abrangidas, mas que exige alerta.</p> <p>Se $> 80\%$ indica um alto grau de abrangência das funcionalidades bem entendidas pelos usuários.</p>

2.4.2. Adaptabilidade

Nome	Taxa de adaptação da aplicação em versões do Android
Objetivo	O objetivo desta medida é para analisar o quanto a aplicação CITRA é adaptável aos aparelhos e suas respectivas versões do Android para garantir a abrangência da aplicação.
Descrição	Quão adaptável o CITRA é em relação aos aparelhos e emuladores do Android?
Fórmula	<p>A= Quantidade de aparelhos/emuladores que a aplicação CITRA funcionou sem problemas.</p> <p>B= Quantidade de aparelhos/emuladores que serão utilizados para testar a adaptação da aplicação.</p> <p>$X = (A/B) \times 100$</p> <p>X é o percentual de assertividade na adaptação</p>

	da aplicação nos aparelhos/emuladores.
Procedimento de coleta	<p>Como coletar: a adaptabilidade será coletada atrás da análise da aplicação em vários aparelhos/emuladores.</p> <p>Como armazenar: em um documento de análise de adaptabilidade.</p>
Análise	<p>Como analisar: analisar a quantidade de aparelhos/emuladores que funcionaram sem problemas na aplicação.</p> <p>Ações corretivas: caso ocorra de aparelhos/emuladores não conseguirem rodar a aplicação, o avaliador deve registrar o aparelho/emulador, a versão do Android e em qual parte da aplicação o problema surge.</p> <p>Onde armazenar: a ocorrência deve ser feita no documento de análise de adaptabilidade com o registro deste impasse.</p> <p>Como executar a revisão da efetividade do indicador: reportar aos desenvolvedores este problema e analisar possíveis soluções para resolver este problema.</p>
Parâmetros	<p>Se $\leq 50\%$ indica um baixo grau de aparelhos/emuladores que executaram a aplicação;</p> <p>Se $> 50\%$ e $< 80\%$ indica um grau mediano de aparelhos/emuladores abrangidos, mas que exige alerta.</p> <p>Se $> 80\%$ indica um alto grau de abrangência dos aparelhos/emuladores que rodam a aplicação.</p>

2.4.3. Testabilidade

Nome	Taxa de cobertura de testes na aplicação
Objetivo	O objetivo desta medida é analisar a testabilidade da aplicação, para garantir a cobertura de testes unitários nas funcionalidades do CITRA.
Descrição	Quão coberto de testes são as funcionalidades da aplicação?
Fórmula	<p>A = Quantidade de funcionalidades testadas da aplicação</p> <p>B = Quantidade de funcionalidades da</p>

	<p>aplicação</p> $X = (A/B) \times 100$ <p>X é o percentual de funcionalidades testadas e devidamente validadas.</p>
Procedimento de coleta	<p>Como coletar: avaliar as implementações no código que precisam ser testadas.</p> <p>Como armazenar: armazenar os testes realizados em uma planilha informando a modularidade que está sendo testada para analisar a cobertura dos testes feitos.</p>
Análise	<p>Como analisar: mapear as funcionalidades que foram e não foram testadas.</p> <p>Ações corretivas: caso exista testes que não levam a nenhuma funcionalidade, deve-se removê-los.</p> <p>Onde armazenar: em outra aba na planilha de testes realizados conforme as funcionalidades.</p> <p>Como executar a revisão da efetividade do indicador: alinhar com os desenvolvedores e time de requisitos a concordância destas funcionalidades.</p>
Parâmetros	<p>Se $\leq 50\%$ indica um baixo grau de funcionalidades que foram testadas.</p> <p>Se $> 50\%$ e $< 80\%$ indica um grau mediano de funcionalidades abrangidas pelos testes, mas que exige alerta.</p> <p>Se $> 80\%$ indica um alto grau de abrangência das funcionalidades pelos testes.</p>

2.4.4. Tolerância a falhas

Nome	Taxa da não permissão de operações incorretas na aplicação
Objetivo	O objetivo desta medida é para avaliar a capacidade do CITRA de manter um nível de desempenho desejável em caso de falhas operacionais.
Descrição	Quão tolerante a falhas a aplicação demonstra ser?
Fórmula	$A = \text{Número de operações incorretas realizadas nas funcionalidades da aplicação.}$

	<p>B = Número de operações incorretas tratadas nas funcionalidades da aplicação.</p> <p>$X = (B/A) \times 100$</p> <p>X é o percentual da quantidade de operações incorretas tratadas e validadas nas funcionalidades.</p>
Procedimento de coleta	<p>Como coletar: analisar as funcionalidades tolerantes a falhas.</p> <p>Como armazenar: por meio de um documento para registrar a validação das operações.</p>
Análise	<p>Como analisar: analisar a quantidade de funcionalidades toleráveis a falhas em relação à quantidade de falhas operadas.</p> <p>Ações corretivas: caso ocorra a não tolerância dessas falhas, ou seja, as operações incorretas não serem tratadas, deve-se documentar este problema.</p> <p>Onde armazenar: em um relatório de falhas do projeto, bem como, este relatório poderá estar atrelado com a planilha de testes da medida anterior.</p> <p>Como executar a revisão da efetividade do indicador: repassando aos desenvolvedores as problemáticas encontradas.</p>
Parâmetros	<p>Quanto mais próximo de 100%, melhor a tolerância de falhas da aplicação.</p> <p>Se $> 70\%$ indica o cenário ideal de tolerância a falhas, porque não há operações incorretas.</p> <p>$\leq 70\%$ e $> 40\%$ indica uma média tolerância a falhas e pode ser melhorada.</p> <p>$\leq 40\%$ indica que tem uma alta taxa de operações incorretas em relação às funcionalidades que deveriam tolerar essas operações.</p>

2.5. Procedimentos de Interpretação

Foi utilizado nas medidas acima o sistema de medidas GQM (Goals Questions Metrics) para consolidação das medidas conforme as métricas. Em cada medida citada acima, foi especificado como serão coletadas e analisadas as métricas.

3.REFERÊNCIAS

ISO/IEC 25000. Software Engineering - Software Product Quality Requirements and Evaluation (SQuaRE) – Guide to SQuaRE. v. 2005, 2005.

ISO/IEC 9126. Software Engineering – Product Quality – Part 1. 2001

ISO/IEC 9126-4: 2000. Software engineering– Software product quality - Part 4:Quality in Use Metrics.

Qualidade de Software. André Koscianski e Michel dos Santos Soares. Editora Novatec. 2a Edição. ISBN 978-85-7522-112-9.