Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Verificação e Validação de Software - 206580

Plano de Projeto

Autores

Fabíola Malta Fleury
Lucas de Araujo Martins
Marlon Mendes Ciriatico Guimaraes
Matheus de Sousa Bernardo
Matheus Vitor Costa Joranhezon
Matheus Richard Torres Gomes de Melo
Thalisson Barreto de Melo Silva

Professor
Ricardo Ajax Dias Kosloski

Brasília, DF
2017

1. Tema

Segundo o SWEBOK v3.0 (2014), qualidade de software pode ser definida como o grau em que um produto de software atende aos requisitos estabelecidos. Entretanto, a qualidade depende do grau em que esses requisitos estabelecidos representam com precisão as necessidades, desejos e expectativas das partes interessadas.

Nesse contexto, Verificação e Validação se encaixa como o processo de aferição de que um software corresponde às suas especificações e de que atende seus propósitos. Estas disciplinas garantem não apenas que os desenvolvedores de software elaborem o produto corretamente, mas que elaboram o produto correto.

1.1. Pesquisa-Ação

Como apresentado por Tripp (2015), a pesquisa-ação é um tipo de investigação-ação, possuindo o intuito de aprimorar uma prática agindo em campo e a investigando. Para isso, são feitos planejamentos, implementações e avaliações sobre mudanças. Essas avaliações podem ser consideradas para melhorar tanto o que está se investigando quanto a investigação em si.

Este trabalho propõe a pesquisa-ação dentro do contexto de um projeto desenvolvido no ambiente de graduação, com o intuito de aplicar conhecimentos da área de verificação e validação de *software* para melhorar a qualidade do processo de desenvolvimento e, por consequência, do código resultante, de forma que este atenda às especificações anteriormente levantadas.

2. Problema

Durante o trabalho conjunto nas disciplinas de Métodos de Desenvolvimento de Software e Gestão de Portfólios e Projetos de Software (GPP/MDS) acontece o desenvolvimento de um produto de software, na primeira parte da disciplina utiliza-se o RUP e na segunda parte é utilizado o Scrum e o XP.

Para esta pesquisa ser desenvolvida, acompanhou-se o desenvolvimento do Falko (https://github.com/fga-gpp-mds/Falko-2017.2-FrontEnd/wiki). Falko é um dos em realização durante o segundo semestre de 2017 nas disciplinas de GPP/MDS com intuito de construir uma ferramenta para gestão de projetos de software desenvolvidos seguindo a metodologia ágil. O grupo é formado por cinco integrantes cursando a disciplina GPP e sete cursando MDS.

O projeto FALKO está sendo desenvolvido utilizando metodologias ágeis baseadas em scrum, XP e Kanban. Os papéis atribuídos dentro do projeto são: scrum master, product owner e time de desenvolvimento. O time trabalha em sprints com duração de uma semana, possuindo os eventos do scrum de: planejamento da sprint, desenvolvimento da sprint, revisão da sprint, retrospectiva da sprint, reunião diária. O fluxo de trabalho da equipe é

documentado em um quadro kanban presente no repositório do *github*. Além disso, métricas para qualidade foram definidas a partir do modelo GQM (*Goal Question Metric)* (Solingen, 1999). Um diagrama representativo das métricas escolhidas pode ser observado na Figura 1.

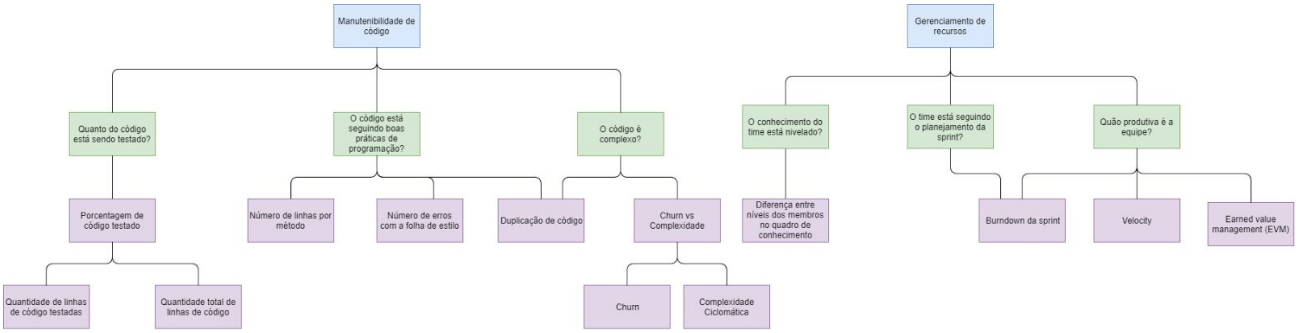

Figura 1 - Diagrama GQM Falkon

Ao decorrer do desenvolvimento do projeto especialmente na segunda parte da disciplina foi detectado a possibilidade de introdução de erros em incrementos do produto. Portanto, foi levantado por meio de um gráfico de espinha de peixe que esses erros podem ser causados por três grandes grupos e todos eles estão relacionados à verificação como mostra a imagem a seguir.

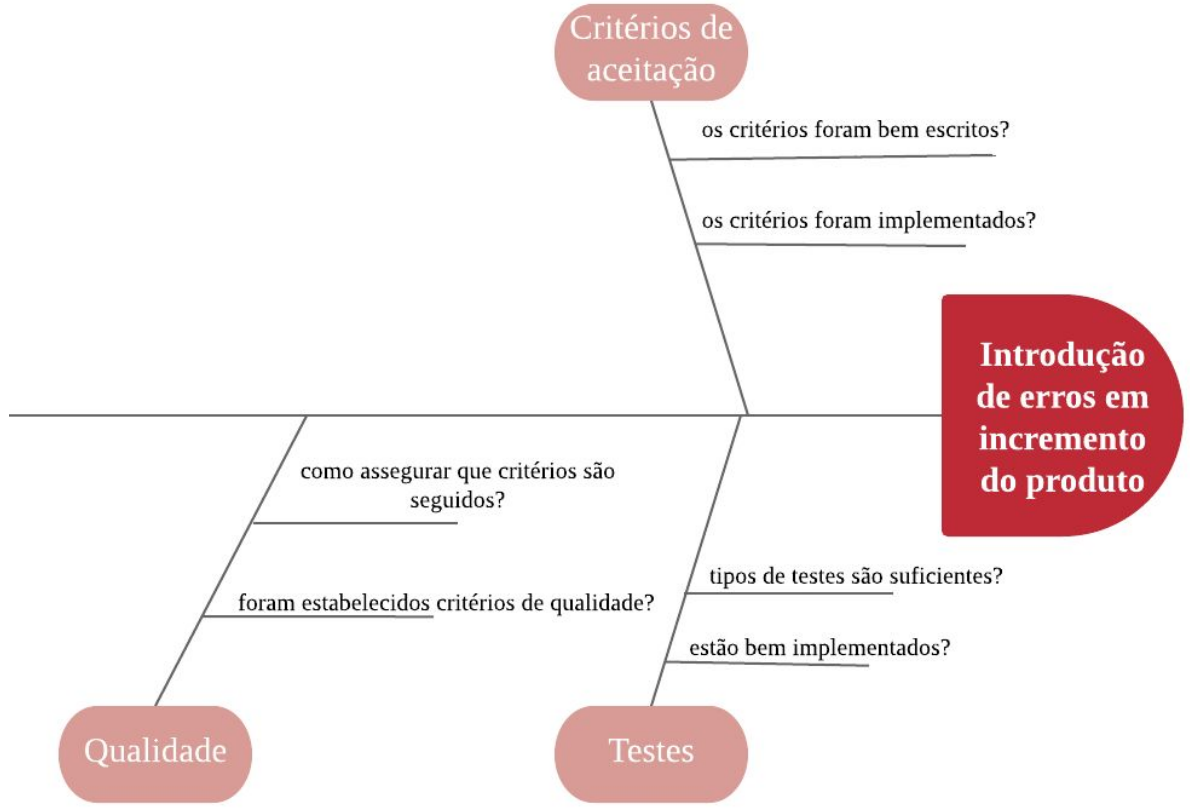

Figura 2 - Diagrama espinha de peixe

3. Objetivo

3.1 Objetivos Gerais

O objetivo dessa solução é estabelecer um processo que melhore a verificação dos incrementos do produto em uma equipe ágil para evitar a introdução de erros na aplicação.

3.2 Objetivos Específicos

Essa solução tem como objetivo criar um processo de verificação que permita auxiliar equipes ágeis a evitar introduzir erros em incrementos do produto. Esse objetivo será alcançado por meio de alguns objetivos específicos.

- Propor um processo de revisão de código para verificar os testes daquele incremento
- Estabelecer um processo para detectar erros causados por falta de critérios com a qualidade do produto
- Estabelecer um processo para verificar os critérios de aceitação daquele incremento do produto

Será verificado se esses processos alcançam os resultados esperados comparando-se as métricas de qualidade do produto e também quantitativamente comparando-se a quantidade de erros introduzidos utilizando os processos e sem utilizá-los

4. Questões de pesquisa

- **Questão 1**: Como garantir que os testes feitos são significativos para o desenvolvimento?
- **Questão 2:** Como evitar que os problemas relacionados aos critérios de aceitação ocorram?
- **Questão 3:** Como assegurar que os critérios de qualidade escolhidos estão sendo seguidos e estão sendo significativos?

5. Referencial teórico

Para esta pesquisa, é importante destacar conceitos ligados a área de testes de *software* e também sobre a metodologia ágil scrum. Estes termos são utilizados como base dentro do estudo de caso estudado para implementação da solução proposta.

De acordo com Meyers (2011), testes são feitos para encontrar erros. Existem diversas técnicas de testes que podem ser dividas em realizadas por máquinas e realizados por humanos. Exemplos de técnicas realizadas por pessoas são: auditorias, revisões em pares, *walkthrough* e inspeções. Testes feitos em máquinas, dentre muitos tipos podem ser: testes de regressão, testes unitários, testes de sistema e testes funcionais.

O Guia Scrum (Sutherland, 2013) define um processo de desenvolvimento que tem como valores a transparência, inspeção e adaptação e como base papéis, artefatos e eventos. O time *scrum* é a equipe de projeto definida em três diferentes papéis: o *scrum master,* responsável por garantir a implementação da metodologia, o time de desenvolvimento, responsável pela entrega de artefatos e o *product owner* que possui a visão do cliente e gerencia o *product backlog*. Os eventos são: a *sprint*, tempo utilizado para criar um incremento de produto,o planejamento da *sprint,* reunião para definir o objetivo da *sprint,*reunião diária, para verificar o progresso da sprint e possíveis impedimentos,revisão da *sprint*, para verificar o incremento produzido e retrospectiva da *sprint* para verificar o processo utilizado. Os artefatos do *scrum* são: *product backlog*, requisitos do projeto, *sprint backlog*, requisitos da *sprint* e o incremento que é aquilo entregue ao final de cada *sprint*.

O estudo de caso conduzido no artigo "*Product Backlog Rating: A Case Study On Measuring Test Quality In Scrum*" (Kayes, 2016) apresenta a utilização de uma adaptação da metodologia *Scrum* utilizada na empresa *SoftwarePeople* pelos suas equipes de desenvolvimento e uma métrica relacionada a qualidade de teste.

No artigo, os artefatos gerados pelos times são: gráfico *burndown*, *sprint backlog* e *dashboard*. Os papéis: *scrum master, product owner*, time de desenvolvimento e engenheiro de garantia da qualidade. Observa-se que o papel engenheiro de garantia da qualidade não é proposto dentro do Guia Scrum (Sutherland, 2013) porém tem o objetivo de garantir a qualidade das entregas, participando ativamente do desenvolvimento, delimitando histórias e entendimentos dos requisitos, ajudando a construir os testes, apresentando para um cliente uma versão do *software* durante a *sprint*, chamado de *sneak peek*, para caso sejam necessárias mudanças nos requisitos, criando o plano de teste para realização de teste de regressão na entrega do incremento ao final da *sprint*.

Testes de aceitação de usuário e *deploy* são realizados por equipes separadas que não fazem parte do time *scrum*.

6. Proposta de solução

Com o intuito de prevenir a introdução de erros no incremento do produto entregue ao cliente do projeto Falko, propõe-se seguir um processo de verificação detalhadamente definido. Assim, são sugeridas mudanças no processo de desenvolvimento e no gerenciamento de qualidade do time, baseando-se no estudo de caso apresentado por Kayes (2016), adaptando o que foi observado para o contexto do projeto sendo desenvolvido e também no que é proposto por Myers (2011, p.21) no que diz respeito à técnicas de testes não baseada em computadores. Serão conduzidas então, ao final da pesquisa, entrevistas com os integrantes da equipe de desenvolvimento quanto às diferenças notadas com as implementações propostas e serão estudadas também as métricas coletadas.

A solução propõe três mudanças:

- Introdução do papel engenheiro de garantia de qualidade;
- Novo objetivo de medição;
- Mudança na revisão antes da entrega de uma história;

O processo de desenvolvimento utilizado atualmente pela equipe Falkon é composto por três papéis: *Scrum master*, *product owner* e time de desenvolvimento. O acréscimo do papel engenheiro de garantia da qualidade tem como objetivo assegurar a qualidade dos artefatos entregues, trabalhando em paralelo à equipe de desenvolvimento. O engenheiro de qualidade é responsável por escrever uma versão concisa de casos de testes, que é então entregue para que os desenvolvedores possam aplicá-las, e acompanha o desenvolvimento das histórias de usuário e do incremento de cada sprint. Assim, a inclusão deste papel proporciona suporte para revisões de código e escrita de testes (KAYES, 2016). O que é importante dentro do contexto que está sendo estudado, tendo em vista que metade da equipe de desenvolvimento tem pouco ou nenhum contato com testes e pode então contar com o apoio deste participante.

Também é modificada como é feita a entrega de cada história. O scrum master é responsável por aceitar a história no incremento da sprint, porém não há descrição dessa atividade, então criou-se um processo formalizado de revisão que será realizado com apoio de *checklist*.

O novo processo pode ser observado na Figura 2 com detalhamento do subprocesso de desenvolvimento da sprint na Figura 3.

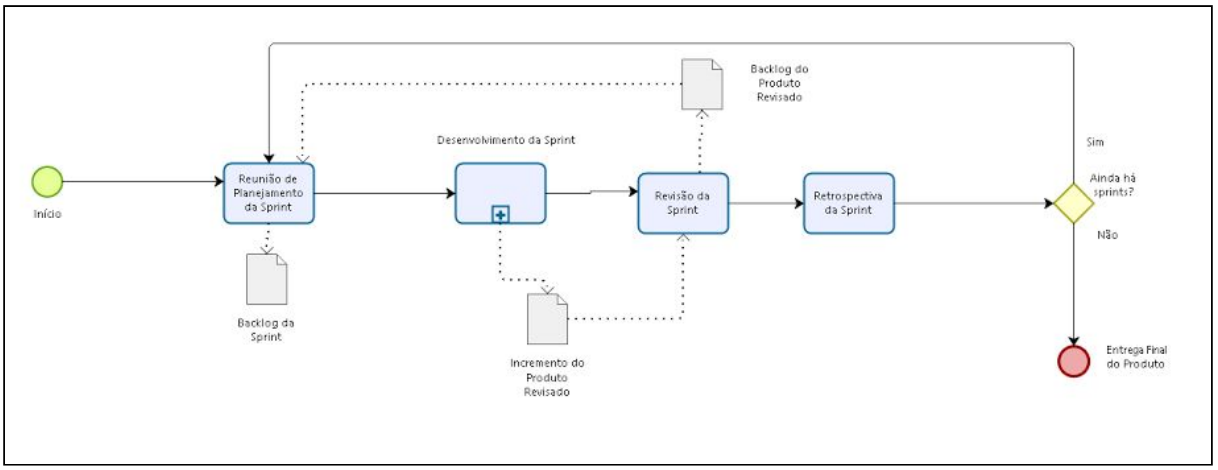

Figura 3 - Processo de desenvolvimento proposto

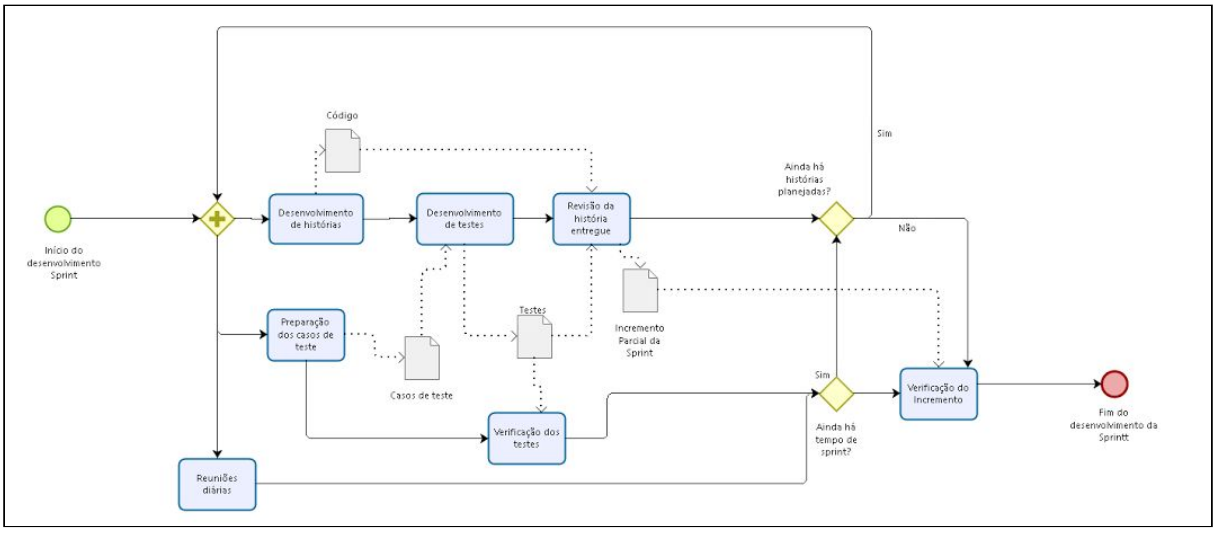

Figura 4 - Subprocesso de desenvolvimento da sprint

Das atividades propostas no processo, é importante destacar que a revisão da história entregue utiliza a funcionalidade de *Pull Request (PR)* da ferramenta *github* como auxílio. O *PR* permite que tanto ferramentas automatizadas quanto pessoas comentem no pedido e deem sugestões ou o recusem. Ao terminar de implementar a história e seus respectivos testes, o desenvolvedor submete um PR para poder acrescentar seu código no que será o incremento da sprint. As ferramentas automatizadas de teste de integração são ativadas e avisam se aquele pedido pode ou não ser integrado. A tabela 1 mostra a revisão realizada após as ferramentas automatizadas cumprirem seu papel.

Tabela 1 - Atividade: Revisão da história entregue

Descrição	Depois da submissão do código da história por meio do *PR* do github, caso seja aceito pelas ferramentas automatizadas,o scrum master revisa esse código utilizando a *checklist* da figura 4. Caso não esteja de acordo,

	ele comenta o tópico criado pelo PR solicitando ao desenvolvedor responsável que faça as alterações necessárias, podendo então, após correções, aceitá-lo para tornar-se parte do incremento da sprint.
Artefatos de Entrada	Código e testes da história
Artefatos Gerados	Incremento parcial da sprint
Responsável	Scrum Master

- [ ] A build foi feita com sucesso
- [ ] Cada critério de aceitação possui ao menos um teste de aceitação
- [ ] Foram implementados todos os casos de testes
- [ ] A cobertura de testes manteve-se ou aumentou
- [ ] Não existem erros do linter (folha de estilo)
- [ ] A complexidade ciclomática dos métodos não passa de 6
- [ ] Não existem métodos duplicados
- [ ] Os métodos possuem 30 linhas ou menos

Figura 5 - *Checklist* da Revisão da história

Além disso também foram revisadas as métricas coletadas pela equipe por meio do modelo GQM e proposto mais um objetivo ligado à qualidade seguindo o processo proposto por Van Solingen (1999). O documento desenvolvido para este objetivo encontra-se no apêndice A. As métricas relacionadas à este objetivo estão ligadas a qualidade dos testes implementados, sendo avaliadas pelo engenheiro de garantia de qualidade.

No processo proposto, existem atividades que serão realizadas pelo novo papel, engenheiro de garantia da qualidade como estão destacados na Figura 5. Estas atividades são voltadas para coleta das métricas propostas pelo novo objetivo e também para apoio na realização dos testes. Os casos de testes serão descritos de forma resumida em formato de *checklist*, sendo entregue para que os desenvolvedores realizem os testes unitários. Depois de terminados os testes e submetidos para avaliação pelo *scrum master* em conjunto ao código, o engenheiro também estará responsável por verificar estes testes.

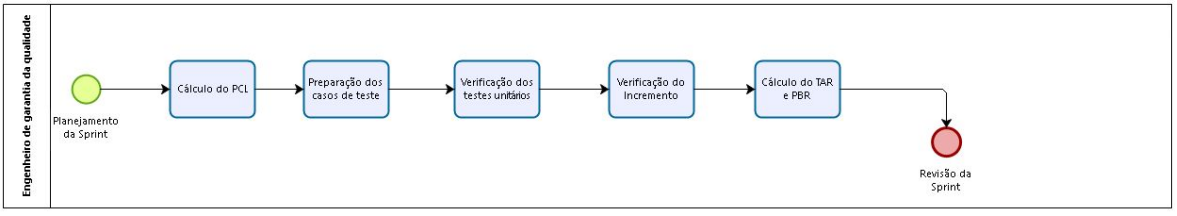

Figura 6 - Subprocesso do Engenheiro de garantida qualidade

A equipe do projeto Falkon trabalhou durante três sprints utilizando sua organização anterior, é proposto que trabalhem três semanas com as alterações aqui sugeridas, sendo as métricas monitoradas e entrevistas feitas com os membros ao final da implementação para que se possa avaliar se a proposta de solução foi de fato efetiva para o seu contexto.

Referências bibliográficas

Bourque, Pierre; Fairley, Dick. **SWEBOK 3.0 Guide to the Software Engineering Body of Knowledge.** [S.l.]: IEEE Computer, 2014

TRIPP, David. Pesquisa-ação: uma introdução metodológica. **Educação e pesquisa**, v. 31, n. 3, 2005.

SUTHERLAND, Jeff; SCHWABER, Ken. The scrum guide. **The Definitive Guide to Scrum: The Rules of the Game. Scrum. org**, 2013.

KAYES, Imrul; SARKER, Mithun; CHAKARESKI, Jacob. Product backlog rating: a case study on measuring test quality in scrum. **Innovations in Systems and Software Engineering**, v. 12, n. 4, p. 303-317, 2016.

MYERS, Glenford J.; SANDLER, Corey; BADGETT, Tom. **The art of software testing**. John Wiley & Sons, 2011.

VAN SOLINGEN, Rini; BERGHOUT, Egon. **The Goal/Question/Metric Method: a practical guide for quality improvement of software development**. McGraw-Hill, 1999.

APÊNDICE

APÊNDICE A - GQM do Objetivo de Medição 03

Objetivo de Medição

Tabela 1 - Objetivo de Medição 03

Analisar	código
para o propósito de	melhorar
com respeito a	detecção de erros
do ponto de vista	do time de projeto
no contexto de	Falko (GPP/MDS)

Questões

A partir da construção do *Abstraction Sheet* foram estabelecidas as questões e métricas da Figura 2.

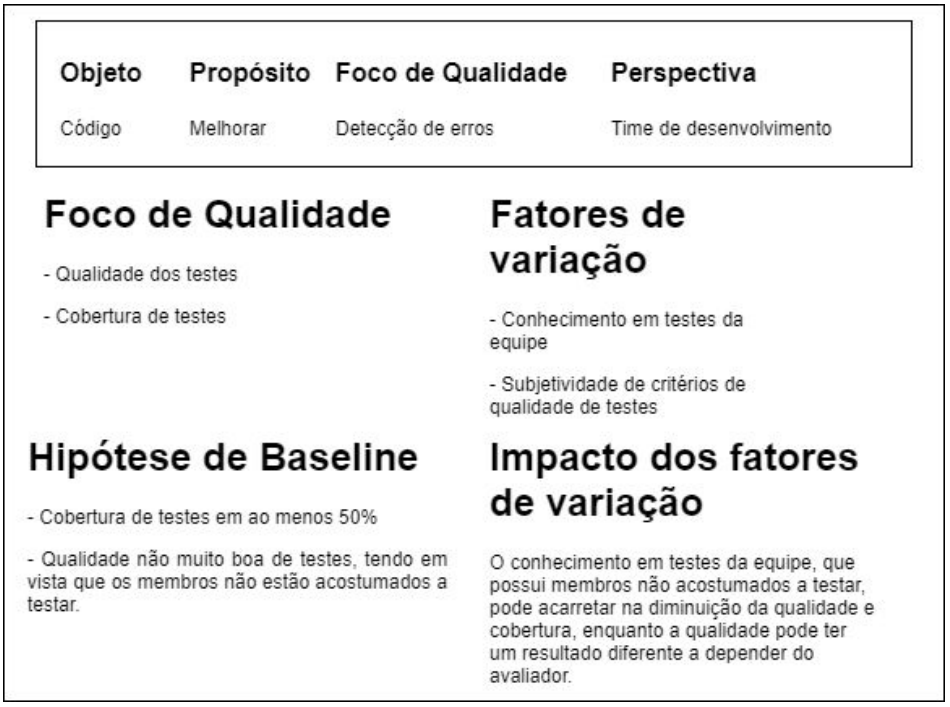

Figura 1 - Abstraction Sheet do Objetivo de Medição 03

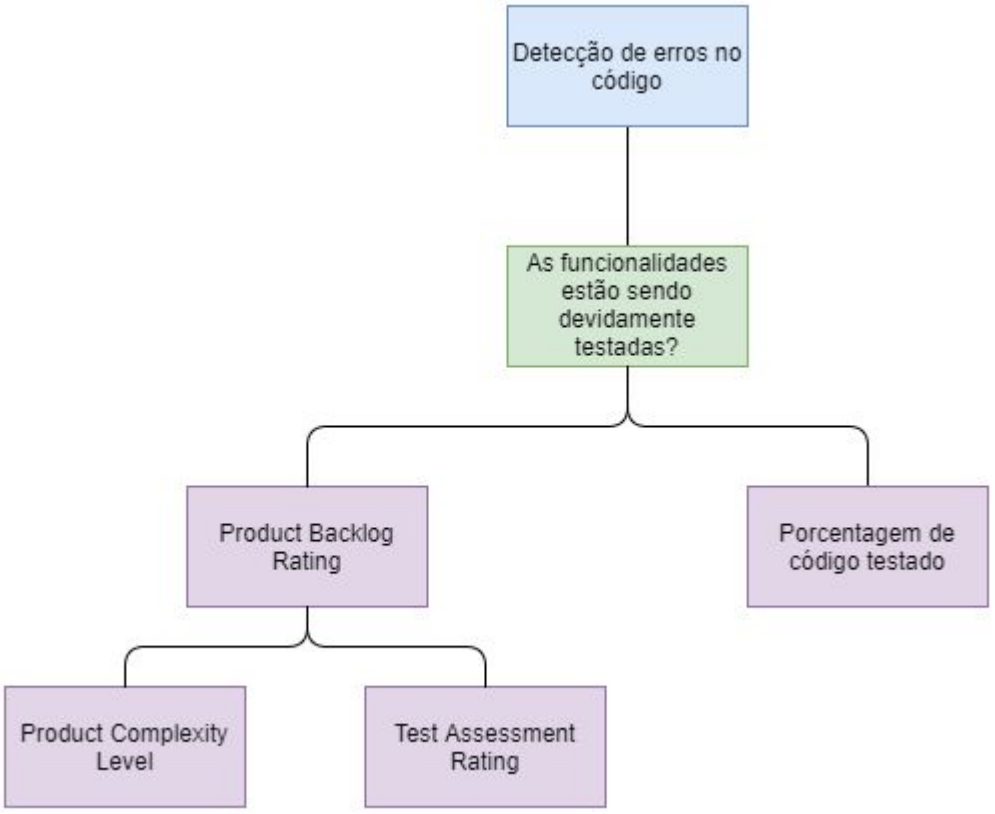

Figura 2 - Diagrama GQM do Objetivo de Medição 03

Métricas

As métricas das tabelas 2 e 3 são utilizadas para cálculo da métrica na tabela 4. As tabelas são detalhamentos das métricas, bem como suas coletas e respectivas análises.

Tabela 2 - Métrica PCL

Métrica	***Product Complexity Level (PCL)***
Objetivo de Medição	Detecção de erros no código
Descrição	É uma nota dada à dificuldade de se realizar testes em um item do product backlog, levando em consideração os seguintes fatores: Necessidade de dados de teste, estimação do teste, dependência interna, dependência externa e complexidade de execução do teste. Estes fatores são avaliados pelo engenheiro de garantia de qualidade.

Fórmula	$PCL_k = \frac{\sum_{i=1}^{i \in P} FP_i * W_i}{\sum_{i=1}^{i \in P} W_i}$, sendo FP os fatores de complexidade (descritos acima) e W o peso de cada um deles.
Escala	Racional
Coleta	Engenheiro de garantia de qualidade Periodicidade do Evento: Ao início de cada sprint.
Procedimento	O cálculo é feito manualmente pelo responsável da coleta.
Análise	Métrica auxiliar
Providência	Ele deve ser usado, juntamente com o *Test Assessment Rating* para o cálculo da métrica *Product Backlog Rating.*

Tabela 3 - Métrica TAR

Métrica	***Test Assessment Rate (TAR)***
Objetivo de Medição	Detecção de erros no código
Descrição	É uma nota dada ao sucesso dos testes em relação aos seguintes fatores: Contagem de bugs, mudança de requisitos e confiança nos testes.
Fórmula	$TAR_k = \sum_{i=1}^{i \in T} \frac{FT_i}{T}$, sendo FP os fatores de complexidade (descritos acima) e W o peso de cada um deles.
Escala	Racional
Coleta	Engenheiro de garantia de qualidade

	Periodicidade do Evento: Ao final de cada sprint.
Procedimento	O cálculo é feito manualmente pelo responsável da coleta.
Análise	Métrica auxiliar
Providência	Ele deve ser usado, juntamente com o *Product Complexity* para o cálculo da métrica *Product Backlog Rating.*

Tabela 4 - Métrica PBR

Métrica	***Product Backlog Rating (PBR)***
Objetivo de Medição	Detecção de erros no código
Descrição	É uma nota dada ao item do Product Backlog após sua entrega.
Fórmula	$PBR = \frac{\sum_{i=1}^{i \in N} PCL_i * TAR_i}{\sum_{i=1}^{i \in N} PCL_i}$, sendo PCL advinda da métrica *Product Complexity Level,* e TAR da métrica *Test Assessement Rate.*
Escala	Racional
Coleta	Engenheiro de garantia de qualidade Periodicidade do Evento: Ao final de cada sprint.
Procedimento	O cálculo é feito manualmente pelo responsável da coleta.
Análise	O PBR possui como resultado um número de 0 a 5 que pode ser interpretado: 5 - Excelente 4 - Bom 3 - Moderado

	2 - Ruim 1 - Péssimo
Providência	Itens do product backlog com notas ruins e péssimos possuirão histórias técnicas criadas para melhoria na qualidade de seus testes.