

Como desenvolvedor quero utilizar *user story* para representar os requisitos que levam à definição do MVP e criação de Mockups

Taynah Almeida¹, Ana Carolina Oran¹, Gleison Santos², Tayana Conte¹

¹Instituto de Computação – Universidade Federal do Amazonas (UFAM)
Manaus – AM – Brasil

²Universidade Federal do Estado do Rio de Janeiro (UNIRIO)
Rio de Janeiro – RJ – Brasil

{tmas, ana.oran, tayana}@icomp.ufam.edu.br,
gleison.santos@uniriotec.br

Abstract. *User stories reflect user’s necessities and the MVP (Minimum Viable Product) aims to validate a software product with its users. Aiming to evaluate if user stories help to define MVP, this paper presents the results of an empirical study. The study shows positive acceptance of user story as a way to specify requirements, to define MVP, and build mockups, due to its ease of use and simplicity. The results show that the defects found after mockups’ inspection are related to human factors. The results show that the defects found after mockups’ inspection were not generated from defects inserted in the user story’s specification, but due the fact that the participants did not follow the specification for the building of the mockups.*

Resumo. *User stories refletem necessidades de um usuário e o MVP (Minimum Viable Product) tem por objetivo validar um produto junto aos seus usuários. Com o propósito de avaliar se user stories auxiliam na definição do MVP, este artigo apresenta os resultados de um estudo experimental que mostram a aceitação positiva da user story para especificar requisitos, definir o MVP e construir mockups, em virtude da sua facilidade de uso e simplicidade. Os resultados mostram que os defeitos encontrados após a inspeção dos mockups não foram gerados a partir de defeitos inseridos na especificação da user story, mas devido ao fato de os participantes não seguirem a especificação para a construção dos mockups.*

1. Introdução

No processo de desenvolvimento de software a fase de especificação de requisitos é uma das mais importantes, qualquer erro cometido nesta fase pode comprometer todo o projeto. Por conta disso, escolher bem a técnica que será utilizada para descrever e documentar as funcionalidades de um sistema é essencial para que o projeto seja bem sucedido [Agarwal e Rathod 2006].

Há diferentes formas para especificar requisitos de um sistema. Uma forma que tem despertado interesse nas comunidades acadêmicas e industriais é a *user story* utilizada no método *Behavior-Driven Development* (BDD ou Desenvolvimento Dirigido

a Comportamento). *User stories*, no formato utilizado em BDD, ajudam a resolver os problemas de comunicação e validação entre *stakeholders* [Carrera Barroso *et al.* 2012], facilitando o entendimento dos requisitos do sistema pelo time de desenvolvimento. De acordo com Silva (2016), BDD ajuda na prevenção de problemas devido à utilização de *user stories* descritas em formato de linguagem natural.

Além de descrever as necessidades dos *stakeholders*, utilizando as *user stories* pode-se construir e validar a ideia principal de um produto junto ao cliente, através da técnica de desenvolvimento *Minimum Viable Product* (MVP). MVP possibilita a criação de um produto mínimo para que a empresa de desenvolvimento antecipe o feedback do cliente sobre a solução proposta [Ries 2014]. Com a utilização de MVP pode-se testar o produto e medir sua aceitação, buscando o ajuste da solução ao problema [Cooper e Vlaskovits, 2010].

Para a definição de um MVP é necessário escolher os requisitos mínimos que atendam às necessidades do cliente e para isso deve-se ter conhecimento dos requisitos do produto. Com o objetivo de verificar se *user stories* auxiliam na definição do MVP e na construção de mockups (representação dos aspectos da interface do usuário [Luna *et al.* 2010]), este artigo descreve a condução de um estudo experimental exploratório e a análise dos resultados, onde foi verificada a utilização da *user story* para especificar requisitos, para definir o MVP e construir mockups, avaliando a aceitação e dificuldades na utilização. Para direcionar a pesquisa, foi definida a seguinte questão de pesquisa: “A *user story* auxilia na definição do MVP e construção de mockups?”. Os resultados demonstram que as *user stories* possuem algumas deficiências, porém estas não comprometem a definição do MVP. Além disso, a análise dos mockups mostrou que os defeitos foram, em maioria, devido aos erros de fator humano.

O restante do artigo está organizado da seguinte forma: a Seção 2 apresenta a fundamentação teórica com conceitos sobre *user stories* no formato utilizado em BDD, sobre MVP e trabalhos relacionados; a Seção 3 descreve o planejamento e execução do estudo experimental; a Seção 4 apresenta a análise dos resultados encontrados no estudo e por fim, a Seção 5 apresenta as conclusões e ameaças à validade do estudo.

2. Fundamentação Teórica

2.1 *User story* usada em *Behavior-Driven Development*

Behavior-Driven Development (BDD) é uma abordagem ágil de desenvolvimento sugerida por Dan North (2006). Esse método surgiu em resposta aos problemas de ambiguidade e erro de comunicação que ocorriam quando técnicos e clientes se comunicavam no *Test-Driven Development* (TDD) [North 2006]. Segundo Solis e Wang (2011), o BDD fornece uma linguagem ubíqua específica que ajuda os *stakeholders* a especificar seus testes. Para North (2006), requisitos são comportamentos associados à *user stories*, que capturam as funcionalidades, com cenários para especificar tais requisitos. O autor criou um critério de aceitação para o comportamento da *user story*, papel que é desempenhado pelos cenários.

As *user stories* são artefatos informais de linguagem natural que surgiram no Desenvolvimento XP (Extreme Programming) [Beck 1999]. Elas são úteis na identificação de funcionalidades de um sistema de software e fazem parte de uma

abordagem ágil cujo objetivo é discutir os requisitos, ajudando a especificá-los. Para Zeaaraoui *et al.* (2013), as *user stories* consistem em descrições breves, na perspectiva do usuário final, das funcionalidades que eles desejam ou necessitam. Na Figura 1 pode-se verificar a estrutura de uma *user story*.

```
Como <papel>: ator que irá utilizar a funcionalidade.  
Eu quero <desejo/característica/objetivo/necessidade>: a funcionalidade do sistema.  
Para que <motivo/justificativa/benefício>: valor de negócio da funcionalidade.
```

Figura 1: Estrutura da *user story*

As *user stories* apoiam a comunicação entre desenvolvedores e clientes, o que permite que ambos tirem dúvidas sobre o projeto e validem as informações de requisitos [Carrera Barroso *et al.* 2012]. Apesar de ter estrutura simples, elas possuem informações que permitem estimar a dificuldade ou o consumo de tempo que o desenvolvimento pode custar, pois podem ser decompostas em tarefas que se tornam unidades quantificáveis de esforço de desenvolvimento [Breitman e do Prado Leite 2002].

As *user stories* são uma maneira simples de expressar comportamentos ou características da necessidade do usuário. No entanto, elas possuem certa limitação, e uma delas é ser aberta a muitas interpretações. Por melhor detalhada que uma *user story* possa ser, ela não apresenta a interação do usuário com o sistema. A *user story* mostra como determinada tarefa pode ser feita, e apesar de poder ser decomposta em tarefas menores, não mostra a complexidade e nem o detalhamento da execução dessas tarefas [Khanh 2017]. Por esse motivo, a *user story*, no formato apresentado na Figura 1, não é suficiente para especificar requisitos e necessita de cenários que descrevem o comportamento do sistema para ajudar nessa parte.

North (2006) desenvolveu um modelo para capturar o critério de aceitação da *user story* que fosse estruturado em forma de cenários com comportamentos, como apresentado na Figura 2. Cenários são descritos por Sutcliffe (1998) como “fatos descrevendo um sistema existente e seu ambiente, incluindo o comportamento dos agentes e informações de contexto suficientes para permitir a descoberta e validação de requisitos do sistema”. Breitman e do Prado Leite (2000) afirmam que cenários devem ser amplamente utilizados durante o processo de desenvolvimento de software, pois utilizam elementos conhecidos pelos usuários e isso o torna um meio confiável de extrair e validar informações de requisitos. O BDD propõe como formato de especificação de requisitos a integração entre *user stories* e cenários, mais detalhes são apresentados no relatório técnico [Almeida *et al.* 2017].

```
Dado <contexto>: pré-condição que deve ser atendida.  
Quando <evento>: interação do usuário com o sistema.  
Então <resposta>: como o sistema se comporta após o evento, sua resposta.
```

Figura 2: Estrutura do cenário utilizado em BDD

A integração das *user stories* com os cenários permite a especificação e o detalhamento do comportamento do sistema em uma linguagem simples que facilita a comunicação e compreensão dos requisitos do sistema e pode ajudar a identificar quais

funcionalidades são prioritárias para o usuário e devem ser incluídas no produto mínimo viável do sistema (MVP), cujo conceito veremos a seguir.

2.2 Minimum Viable Product

O conceito de MVP (Produto Mínimo Viável, do inglês *Minimum Viable Product*) foi criado aproximadamente em 2001, tornando-se popular alguns anos mais tarde, no contexto de *startups* [Ries 2014]. Trata-se da identificação e implementação de um produto que entrega valor real ao cliente [Paternoster 2014]. MVPs são definidos como produtos com funcionalidades suficientes apenas para reunir aprendizado validado sobre o produto e devem ter enfoque nos estágios iniciais de desenvolvimento [Duc e Abrahamsson 2016]. É a construção da versão mais simples de um produto que pode ser disponibilizada para negócio e que necessita de quantidade mínima de esforço e desenvolvimento [Ries 2014].

Segundo Ries (2014), MVP é a versão do produto que garante uma volta completa no ciclo construir-medir-aprender, com o mínimo de esforço e menor tempo de desenvolvimento. Embora muitas funcionalidades do produto que não estão no MVP logo sejam demandadas pelos usuários, é preciso evitar o impulso de incluí-los no desenvolvimento inicial, pois o tempo será mais bem gasto desenvolvendo os experimentos que medirão o impacto do MVP [Moogk 2012]. Caroli (2015) explica que o objetivo do MVP é validar a ideia principal de um produto antes de investir tempo, dinheiro e esforço em algo que não irá atender as expectativas do cliente. Portanto, o MVP é uma versão mais simples de um produto, com as mínimas características necessárias para ele ser inserido no mercado e que deve ser complementado em ciclos posteriores de desenvolvimento.

Ries (2014) demonstra o passo a passo do processo da seguinte forma: primeiro deve-se imaginar o problema que precisa de solução e então desenvolver um MVP para começar o processo de aprendizagem o mais rápido possível. Uma vez que o MVP está estabelecido, é possível trabalhar uma versão inicial da aplicação.

Neste trabalho são utilizados dois dos sete passos citados por Caroli (2015) para a definição de um MVP, devido ao foco estar relacionado à elicitación de requisitos e não na parte de negócios que visa avaliar a viabilidade econômica. Os passos utilizados foram: (i) visão do produto, que descreve o software a ser desenvolvido em termos de utilidade e usuário; (ii) elicitación de requisitos e jornadas, que divide as funcionalidades do MVP e demais iterações.

2.3 Trabalhos Relacionados

Keogh (2010) destaca a importância do método BDD para todo o ciclo de vida do desenvolvimento de software, especialmente para a interação entre atividades de negócios e desenvolvimento de software. O autor argumenta que o método permite fazer entregas de valor para o cliente ao definir comportamentos. Ele afirma que a linguagem natural utilizada no BDD, para ilustrar exemplos e comportamentos, ajuda a decompor a aprendizagem para elicitación de requisitos e gerar entendimento compartilhado do domínio.

Lazăr *et al.* (2010) afirmam que o BDD permite que desenvolvedores e clientes falem a mesma língua, encorajando a colaboração entre todos os participantes do projeto. Os autores pontuam dois princípios do BDD: 1) desenvolvedores e clientes

devem se referir ao mesmo sistema da mesma forma e 2) todo sistema deve ter um valor identificado e verificável para o negócio.

Münch (2013) destaca, que durante o desenvolvimento de um novo produto de software, faz-se necessário obter feedback do cliente final o mais rápido possível. Para isto, realizou um estudo que utilizou a colaboração indústria-academia para desenvolver um software com ciclo interativo e incremental, com o método de MVP que trata de um produto de software que pode estar incompleto em funcionalidade ou qualidade, mas exibe características que permitem determinar o valor para o cliente.

Nos trabalhos apresentados não encontramos aplicação de *user stories* como base para construção de MVP. Por esta razão, o objetivo deste estudo foi verificar se *user stories* ajudam desenvolvedores a definir o MVP e construir mockups.

3. Estudo experimental

Com o propósito de verificar a aceitação e a dificuldade de uso das *user stories*, utilizadas no BDD, na definição do MVP e construção de mockups foi realizado um estudo experimental exploratório. As Seções 3.1 e 3.2 apresentam, respectivamente, o planejamento e a execução desse estudo. A versão completa do pacote experimental encontra-se disponível no relatório técnico [Almeida *et al.* 2017].

3.1 Planejamento do estudo

Para orientar o projeto do estudo experimental, seu objetivo foi definido usando o paradigma GQM [Basili 1988]: “Analisar a relação entre *user story* usada em BDD e MVP com o propósito de verificar a facilidade de uso, com respeito à especificação de requisitos e construção de mockups do ponto de vista de pesquisadores em engenharia de software no contexto de estudantes de graduação e pós-graduação”.

O estudo foi realizado em um ambiente acadêmico e conduzido com 31 estudantes de graduação dos cursos de Ciência da Computação, Sistemas de Informação e Engenharia da Computação da Universidade Federal do Amazonas (UFAM), que estavam cursando a disciplina introdutória de “Engenharia de Software”. Os alunos foram caracterizados como novatos, uma vez que não conheciam especificação de requisitos e tinham apenas conhecimento acadêmico sobre *user stories*. Não foi realizado teste piloto, entretanto o protocolo de experimentação e formulários foram elaborados por dois dos autores e revisados pelos outros dois.

Todos os participantes assinaram o formulário de consentimento, o Termo de Consentimento Livre e Esclarecido (TCLE), no qual concordaram em fornecer os resultados para análise. Os artefatos utilizados no estudo foram: a) descrição textual de um cenário de aplicação; b) modelos de especificação de *user stories*; c) artefatos para criação da especificação; d) artefatos para definição do MVP; e) artefato para criação de mockups; e) artefato para inspeção dos mockups; f) questionários de avaliação, conforme apresentados na Tabela 1 e g) artefato para criação da visão do produto.

Todos os participantes receberam treinamento de duas horas em um mesmo ambiente sobre *user stories* utilizadas em BDD e sobre MVP. Durante o treinamento também foram realizados exercícios práticos sobre *user story* e MVP. Não foi

necessário realizar treinamento sobre construção de mockups, pois os participantes já possuíam conhecimento prévio.

Tabela 1: Extrato dos questionários aplicados [Almeida et al. 2017]

Questionário	Perguntas
A) Avaliação de especificação	1) Você teve dificuldades ao especificar no formato user stories? Quais? 2) Quais são as vantagens da user story? Justifique.
B) Avaliação de MVP	1) Você teve dificuldade em extrair informações da especificação user story para definição do MVP? Teve algum aspecto que não estava especificado que fez falta na definição do MVP? Quais? 2) Você acredita que a user story é útil para identificar os requisitos do MVP? Justifique.
C) Avaliação de mockups	1) Você teve dificuldade em fazer o mockup do MVP? Teve algum aspecto do MVP ou da user story que dificultou a construção do mockup? Quais? 2) Se você tivesse que fazer outro mockup de MVP, acredita que a especificação user story ajudaria? Por quê?
D) TAM:	1) Minha interação com a user story foi clara e compreensível. 2) Interagir com a user story não exige muito do meu esforço mental.
E) Avaliação geral	1) A especificação user story detalhou bem as informações para a definição do MVP? Por quê? 2) Houve retrabalho ou outro problema ao utilizar a especificação user story? Quais?

Para avaliar os mockups criados baseados nas *user stories*, foi criado um formulário de inspeção com tipos de defeitos baseados em Travassos *et al.* (1999), apresentado na Tabela 2.

Tabela 2: Classificação de defeitos entre especificação e mockups.

Categoria	Descrição	Exemplo
Omissão	Informações que foram descritas na especificação e não foram inseridas no mockup	Na especificação tem a descrição: “O cliente seleciona a opção Salvar” No mockup não tem o botão Salvar
Fato incorreto	Informações que foram descritas na especificação e inseridas no mockup de outra forma	Na especificação tem a descrição: “O cliente seleciona a opção Salvar” No mockup o nome do botão é Inserir
Informação estranha	Informações que foram inseridas no mockup e que não estavam descritas na especificação	No mockup tem a opção Salvar e Cancelar. Na especificação só tem a descrição da opção Salvar.
Inconsistência	Informações que foram descritas como um cálculo ou informação derivada de outra que no mockup não obedeceram a dependência, um conflito de informações.	No mockup tem que mostrar os campos idade e preço da entrada do cinema, dependentes da data de nascimento. Ou seja, o campo idade e preço devem ser calculados a partir da data de nascimento e não inserido pelo usuário.
Ambiguidade	As informações que foram inseridas no mockup são ambíguas, isto é, é possível que os stakeholders interpretem de maneira diferente.	No mockup tem botões com o mesmo nome, campos com descrição iguais ou sem navegação entre as telas. Nome de botão ou campo sem o objetivo claro: o nome do botão é OK e deveria ser Salvar.

Foi utilizado o cenário de um sistema de apoio ao gerenciamento do andamento do Trabalho de Conclusão de Curso (TCC) por parte do coordenador do curso, do orientador e do aluno. As funcionalidades do sistema são: cadastrar os orientadores e alunos, cadastrar cronograma de atividades, disponibilizar programação do Workshop de TCCs, formalizar a orientação, cadastrar o tema do TCC, trocar mensagens com o orientador via chat, realizar as entregas de tarefas e atividades pelo sistema, visualizar o calendário de atividades e de workshop de TCC, cadastrar temas de interesse para orientação, cadastrar o plano de trabalho com as metas definidas, cadastrar as tarefas do aluno, visualizar as tarefas do aluno, cadastrar as notas das atividades do orientando e cadastrar a defesa. Como restrição, o sistema não permite cadastrar mais de uma defesa de TCC no mesmo horário e local. O cenário foi representado em formato de linguagem natural e sua qualidade foi verificada por todos os autores.

3.2 Execução do estudo experimental

O estudo experimental foi realizado em quatro dias: no primeiro dia, foi realizado o treinamento sobre *user stories* e sobre MVP para todos os participantes. Além do treinamento, foram realizados exercícios sobre ambos os tópicos.

No segundo dia, os participantes foram divididos aleatoriamente em seis grupos, com entre quatro e seis integrantes cada. Os participantes receberam o cenário (Gerenciamento de TCC) para fazerem as especificações com *user stories* no formato utilizado em BDD e também modelos para serem seguidos e auxiliar a especificação dos requisitos com as *user stories*. Foi aplicado um questionário de avaliação de especificação com o objetivo de capturar a experiência dos participantes na utilização de *user story* para especificação (questionário A da Tabela 1).

No terceiro dia, as atividades foram voltadas para definição de um MVP baseado na especificação e na construção dos mockups do MVP. Cada grupo utilizou sua própria especificação, feita no segundo dia, artefatos para elaboração de visão do produto, do MVP e dos mockups para realizar essas atividades. Ao final da atividade, foram aplicados questionários de avaliação sobre a definição do MVP com o objetivo de capturar a experiência dos participantes na utilização da *user story* para a definição do MVP (questionário B da Tabela 1) e questionário sobre os mockups para avaliar a utilização do MVP e *user stories* na construção dos mockups (questionário C da Tabela 1). Além disso, foi aplicado o questionário TAM para avaliação da facilidade de uso, utilidade e uso futuro da *user story* (questionário D da Tabela 1).

No quarto dia foi realizada a inspeção dos mockups, onde cada grupo recebeu a especificação e os mockups criados por outro grupo, para identificar os defeitos entre o que foi especificado, o que foi incluído no MVP e o que foi construído no mockup. Dessa forma, o grupo 1 verificou os defeitos dos mockups criados pelo grupo 5, o grupo 2 verificou nos mockups feitos por 6, grupo 3 os feitos por 4, grupo 4 os feitos por 3, grupo 5 os feitos por 1 e grupo 6 os feitos pelo grupo 2. Após a inspeção dos mockups, foi aplicado um questionário geral (questionário E da Tabela 1) visando capturar a experiência da atividade como um todo.

4. Análise dos Resultados

Nesta seção, apresenta-se a análise dos resultados sobre a aceitação e dificuldades de uso das *user stories* para definição do MVP e construção de mockups. Para isso foi analisada a percepção sobre facilidade de uso, utilidade e uso futuro da especificação com *user story* e os resultados da inspeção realizada para detectar defeitos cometidos na criação dos mockups a partir das especificações.

4.1 Análise da percepção sobre facilidade de uso, utilidade e uso futuro

Com o objetivo de avaliar a aceitabilidade das *user stories*, foram analisados os questionários sobre a aceitação da tecnologia definidos com base nos indicadores do *Technology Acceptance Model* (TAM) [Davis 1989] – questionário D da Tabela 1. Os dados coletados foram apresentados em forma de gráficos para fim de análise. Os indicadores definidos foram: (i) **utilidade percebida**, que indica o grau que uma pessoa acredita que a tecnologia pode melhorar seu desempenho no trabalho, (ii) **facilidade de uso percebida**, que indica o grau que uma pessoa acredita que usar a tecnologia específica seria livre de esforço e (iii) **intenção de uso futuro**, que indica o grau que uma pessoa acredita que voltará a utilizar a tecnologia novamente. A razão para focar nestes indicadores é que, de acordo com Davis (1989), estes aspectos são fortemente correlacionados com a aceitação da tecnologia pelo usuário.

Os participantes forneceram suas respostas em uma escala de seis pontos, baseados nos questionários aplicados por Lanubile *et al* (2003). As possíveis respostas nos questionários são: concordo totalmente, concordo amplamente, concordo parcialmente, discordo parcialmente, discordo amplamente e discordo totalmente. Não foi utilizado um nível intermediário, uma vez que este nível neutro não fornece informações sobre o lado ao qual os sujeitos estão inclinados (positivos ou negativos) [Laitenberger e Dreyer 1998]. Neste questionário, os participantes responderam o seu grau de aceitação referente à utilidade, facilidade de uso e intenção de uso futuro.

A Figura 4 apresenta as percepções dos participantes em relação à facilidade de uso das *user stories* no formato utilizado em BDD. O eixo X dos gráficos da Figura 4 refere-se às possíveis respostas do questionário pós-modelagem e o eixo Y refere-se à quantidade de participantes. É possível observar que na questão “Interagir com a *user story* não exige muito do meu esforço mental” não houve homogeneidade nas respostas e os participantes ficaram divididos entre concordar parcialmente e discordar parcialmente. Entre os que discordaram, o participante P6 afirmou que teve dificuldade “(...) em descrever as funções”, P12 afirmou que teve “*dificuldade de diferenciar uma funcionalidade (user story) de um cenário*” e P30 disse que não sabia “(...) Definir o nível de detalhes das especificações, se deve estar mais próxima do cliente ou dos desenvolvedores”. Na questão “Minha interação com a *user story* foi clara e compreensível” apenas dois participantes discordaram de alguma forma, o participante P14 disse que “(...) fiquei um pouco confuso em como diferenciar as funcionalidades (*user stories*) dos cenários”. Na questão “Considero a *user story* fácil para especificar requisitos” três participantes discordaram, o participante P22 afirmou que teve dificuldade “(...) para saber a forma de separar e lidar com outra *user story* relacionada”. Na questão “Eu acredito que é fácil conseguir que a *user story* faça o que quero” cinco participantes discordaram, o participante P9 disse “(...) demorei para

identificar funcionalidades que deveriam existir para o programa funcionar corretamente". A grande maioria dos participantes concordou total, ampla ou parcialmente, ou seja, no geral, *user stories* foi considerada fácil de utilizar.

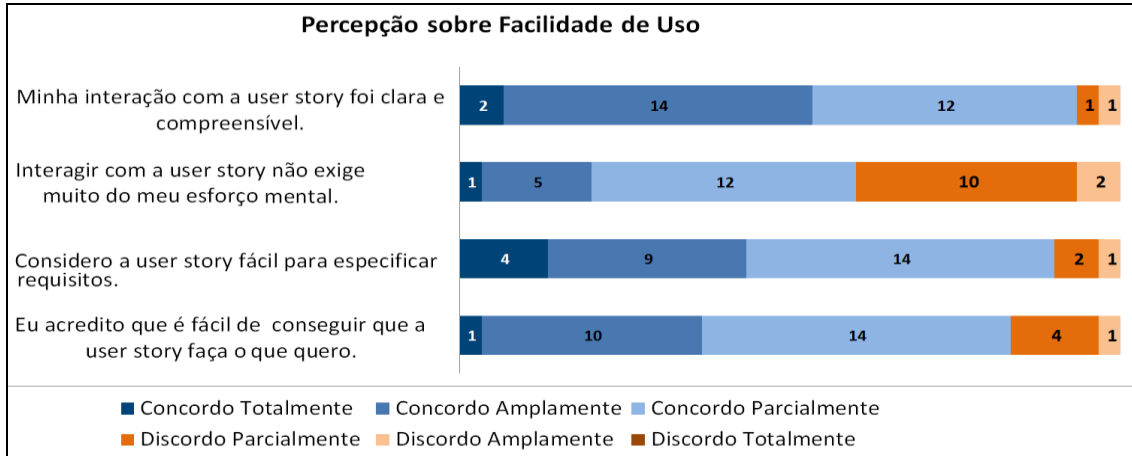


Figura 4: Percepção dos participantes sobre a Facilidade de Uso da user story

A Figura 5 apresenta as percepções dos participantes em relação à utilidade das *user stories*. Novamente, os gráficos apontam para a concordância dos participantes, que consideraram as *user stories* úteis para definição do MVP do sistema. Apenas um participante discordou totalmente em três das quatro questões (P23). Este participante afirmou que para a definição do MVP a “(...) *user story* não é clara o suficiente”. Para a questão “Usando a *user story*, melhorou o meu desempenho na definição de MVPs” dois participantes discordaram parcialmente, o participante P30 afirmou que “*Uma das desvantagens da user story é a simplicidade, pois não é possível listar todas as funcionalidades de uma vez.*”. Para a questão “Usando a *user story*, aumentou minha produtividade na definição de MVPs” três participantes discordaram parcialmente, o participante P21 disse que “(...) *faltaram algumas coisas*”. E para a questão “Usando a *user story*, me permitiu detectar os requisitos do MVP mais rapidamente” o participante P21 discordou parcialmente e disse que “(...) *foi percebido que faltou uma tela de login*” e P16 discordou amplamente, dizendo que “*user story estava bem especificada, mas o aspecto como login e etc. não foram usados na user story, causou dúvidas*”.

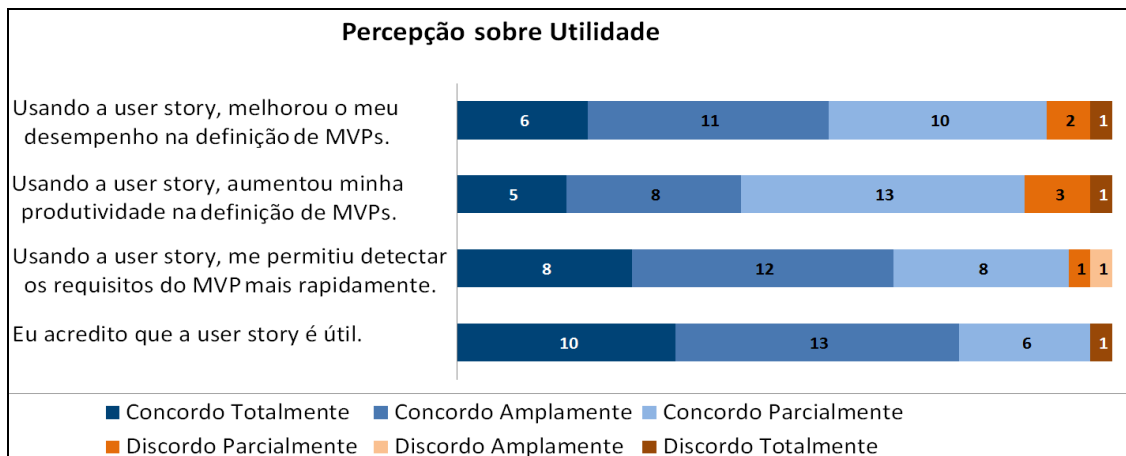


Figura 5: Percepção dos participantes sobre a Utilidade da user story

A Figura 6 apresenta as percepções dos participantes em relação ao uso futuro das *user stories*. Como podemos verificar, a maioria dos participantes concordou em utilizar novamente as *user stories* e apenas um participante (P22) discordou totalmente das duas questões. Este participante afirmou que o “(...) tempo de especificação demora para ser feito”. Na questão “Eu prevejo que vou usar *user story* no futuro para auxiliar a definição de MVPs” apenas um participante discordou parcialmente (P16) apesar de ter tido uma opinião contrária na questão “Se você tivesse que fazer outro MVP novamente, gostaria de receber uma especificação *user story*? Por quê?”, do questionário B da Tabela 1, onde o participante respondeu “*Sim, pois a user story auxilia na construção da funcionalidade*”. Na questão “Eu preferiria usar a *user story* para auxiliar a definição de MVPs ao invés de fazê-los de forma tradicional” três participantes discordaram parcialmente, o participante P9 disse que “(...) Devido às falhas na especificação, o MVP saiu com menos coisas do que nossa equipe gostaria”.

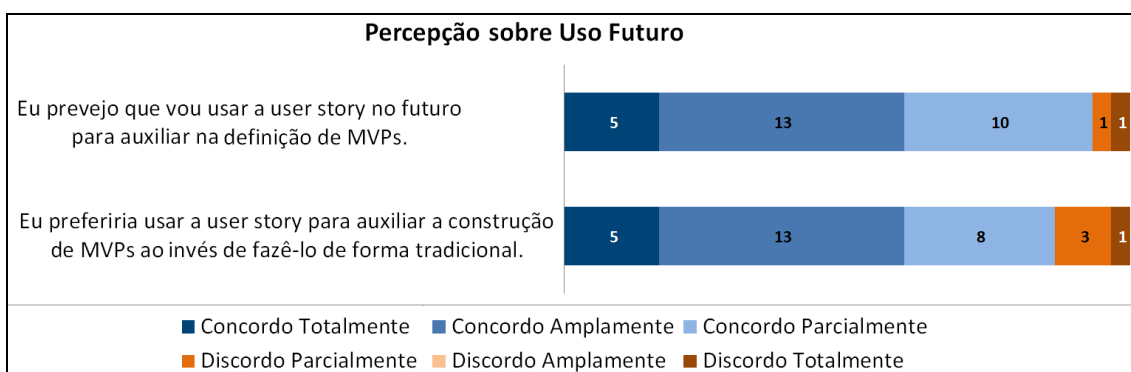


Figura 6: Percepção dos participantes sobre o Uso Futuro da *user story*

4.2 Análise das inspeções

A avaliação dos mockups do MVP em relação à especificação resultou em 48 defeitos. Os tipos de defeitos identificados na inspeção dos mockups criados a partir da especificação de *user story* foram: Fato incorreto (18), Informação estranha (12), Omissão (12), Inconsistência (2) e Ambiguidade (4).

A grande maioria dos defeitos cometidos não estava relacionada aos defeitos inseridos na especificação das *user stories* e sim ao fato de a especificação não ter sido totalmente seguida na construção dos mockups, como por exemplo: não inserção de informações que estavam descritas, alteração de mensagem no mockup, inserção de botões não especificados, não indicação da navegabilidade entre as telas e apresentação de formato dos campos de maneira incorreta.

4.3 Análise Qualitativa

Além da análise realizada por meio do modelo TAM, foi feita uma análise específica dos dados qualitativos contidos nos questionários aplicados ao longo da execução do estudo, correspondente a comentários e observações dos participantes. O objetivo foi avaliar as *user stories* e identificar possíveis falhas.

A análise dos resultados sobre a especificação identificou as dificuldades encontradas pelos participantes (P) na utilização das *user stories*:

- Dificuldade em diferenciar funcionalidades (*user stories* – Figura 1) e cenários (comportamento – Figura 2):

- “(...) *fiquei um pouco confuso em como diferenciar as funcionalidades dos cenários*” – P14
- “*Tive dificuldade confundindo funcionalidades com cenários*” – P10

Quanto à comunicação de requisitos utilizando *user stories*, boa parte dos participantes afirma que existem outras especificações que comunicam os requisitos aos integrantes da equipe e que as *user stories* não cobrem detalhes, conforme mostram as citações abaixo:

- “(...) *acho que outras especificações serão necessárias antes de começar a implementar uma funcionalidade*” – P13
- “(...) *necessária especificação mais detalhista para os desenvolvedores*” – P30

Quanto à análise dos resultados com relação à utilização de *user stories* para definição do MVP e criação dos mockups, foram identificados três problemas: algumas funcionalidades do MVP não estavam especificadas, houve dificuldade em definir o que entra no MVP (P13) e algumas funções não ficaram claras (P6).

- “(...) *a tela de login que foi totalmente esquecida durante a especificação*” – P1
- “(...) *a user story estava bem especificada, mas o aspecto como login e etc não foram usados na user story, causou dúvidas*” – P16

O participante P4 destacou como ponto positivo as *user stories* ajudarem a identificar funcionalidades que faltaram. Além disso, os demais participantes ressaltaram que as *user stories* são fáceis, simples e que conseguem descrever as funcionalidades detalhadamente, o que facilita a visualização do sistema. Outros cinco participantes destacam que elas oferecem uma visão clara do grau de necessidade do usuário. Apenas o participante P13, discorda e acredita que *user stories* não são úteis para identificar os requisitos para definição do MVP.

- “(...) *mais fácil para escolher os mais importantes*” – P4
- “(...) *demonstra com elegância as necessidades e cenários específicos para cada funcionalidade, especificando corretamente o espelho do comportamento*” – P15

Apesar de acreditarem que o uso de *user stories* ajudou na definição dos MVPs, os participantes afirmaram que elas não são suficientes. O participante P4 considera que a especificação fica incompleta e que algumas funcionalidades ficaram pendentes. Os participantes P13, P14 e P15 acham que é necessária deliberação adicional para o entendimento da funcionalidade. Outros acreditam que as *user stories* servem apenas para ter uma ideia geral do produto.

- “(...) *foi necessário deliberação adicional entre a equipe*” – P13
- “(...) *ainda faltou algumas funções que não estavam na user story*” – P6

A maior dificuldade encontrada em relação à construção do mockup do MVP foi por conta da especificação *user story* estar incompleta, conforme pontuam os participantes P30 e P19:

- “(...) *algumas funcionalidades não foram incluídas pois não foram incluídas na user story*” – P30
- “*A dificuldade foi devido a não fazer a user story com todos os cenários que só foram percebidos durante a execução dos mockups*” – P19

Entretanto, quando questionados se utilizariam novamente a técnica para fazer o mockup de um MVP, a maioria disse que sim, pois as *user stories* fornecem visão ampla do sistema e isso ajuda a identificar requisitos.

- “(...) *a user story bem feita fornece uma Visão ampla do sistema*” – P7
- “(...) *com a user story já é possível ter uma certa pré-visualização de como pode ser feito o mockup e o comportamento.*” – P8
- “(...) *a especificação idealiza a essência de funcionamento e das características da parte crítica (fundamental) como um todo.*” – P15

Nos resultados do questionário final, foi verificada a percepção dos participantes em relação a toda a atividade envolvendo especificação de requisitos até elaboração do mockup de MVP. Os aspectos positivos ressaltados foram em relação à visão detalhada do sistema a ser desenvolvido.

- “(...) *é fácil ter visão geral do sistema e analisar o essencial*” – P7
- “(...) *define de maneira clara os requisitos das funcionalidades que podem ser inseridas no MVP*” – P8
- “(...) *as funcionalidades e cenários que descrevem bem o comportamento do sistema e facilita a criação do projeto do MVP*” – P14

Os pontos negativos da utilização da técnica envolvem principalmente o retrabalho que pode ocorrer, tendo que escrever cenários muito parecidos para diferentes tipos de atores. Além disso, a falta da especificação de funcionalidades e de organização das *user stories* foi apontada por muitos participantes.

- “*percebemos que não havíamos especificado coisas essenciais como navegação e login*” – P7
- “(...) *não especificamos uma característica implícita no problema: a tela de login*” – P15
- “(...) *a organização da especificação não facilitou a verificação entre o MVP e as especificações*” – P2

Ao verificar se os participantes utilizariam novamente *user stories* para fazer especificação, os motivos listados por aqueles que concordaram foram: simplicidade, facilidade e praticidade de visualizar o projeto, nível de detalhamento, interação desenvolvedor-usuário, agilidade. Três participantes ficaram na dúvida e acreditam que depende do tamanho e tipo do projeto.

Dentre as vantagens das *user stories* mais citadas pelos participantes, estão a facilidade de comunicação entre equipe de desenvolvedores e o cliente, especificações serem detalhadas e a facilidade tanto de fazer as especificações quanto de enxergar o tamanho do sistema de maneira geral. As desvantagens identificadas pelos participantes

são o fato de que muitos cenários fazem a *user story* ficar extensa e repetitiva, e o fato de elas não evitarem que alguns cenários sejam esquecidos.

5. Conclusão

Este trabalho apresentou um estudo experimental exploratório que analisou os resultados relacionados à aceitação e dificuldades existentes na utilização de *user stories* para definição de MVPs e construção de mockups. A questão de pesquisa investigada foi avaliar se a *user story* auxilia na definição do MVP e construção de mockups.

De acordo com a análise realizada sobre os resultados alcançados, identificou-se que *user story*, como forma de especificação de requisitos, teve aceitação positiva pelos participantes no auxílio na definição do MVP e construção de mockups. Entretanto, foi possível observar que a técnica possui algumas deficiências, mas que não foram suficientes para comprometer a definição do MVP e construção de mockups. Portanto, não é possível afirmar que a utilização de *user stories* é capaz de prejudicar a definição do MVP de um produto de software. A partir da análise das inspeções realizadas nos mockups, foi verificado que os defeitos encontrados nos mockups não foram gerados a partir de defeitos inseridos na especificação da *user story*, mas devido aos participantes não seguirem estritamente a *user story* como base para construção de mockups.

Neste estudo, existiram algumas ameaças que podem afetar a validade dos resultados e que foram mitigadas quando possível. As principais ameaças encontradas foram: (1) efeitos de treinamento: os participantes receberam treinamento para especificação de *user stories* que incluíam atividades teóricas e exercícios práticos em sala de aula; (2) uso de cenário: foi minimizado utilizando cenário escrito em linguagem natural, onde os requisitos deste cenário estavam explícitos de forma similar aos exercícios realizados no treinamento; (3) Representatividade dos participantes: os participantes do estudo foram estudantes, e ainda que Höst *et al.* (2000) afirmem que estes podem representar adequadamente uma população de profissionais da indústria, os participantes foram caracterizados como novatos e podem ter tido dificuldade para compreender algumas questões dos formulários de avaliação; (4) Tamanho da amostra: devido ao número limitado de participantes os resultados deste estudo não podem ser considerados conclusivos. A participação de estudantes e a quantidade da amostra reduzida foram consideradas limitações aos resultados deste estudo.

Como trabalho futuro, pretende-se fazer estudos de caso na indústria para reafirmar ou descartar os problemas encontrados e, possivelmente, encontrar novos. Além disso, replicar o estudo utilizando outras especificações simultaneamente, para observar o resultado e comparar com os obtidos utilizando *user stories*.

Agradecimentos

Os autores agradecem o apoio financeiro do CNPq processo 423149/2016-4, CAPES processo 175956/2013, da FAPEAM, processo 062.00578/2014, da FAPERJ (projetos E-26/010.000883/2016, E-211.174/2016) e UNIRIO (PQ-UNIRIO 01/2016 e 01/2017).

Referências

- Agarwal, N. and Rathod, U. (2006). Defining ‘success’ for software projects: An exploratory revelation. *International journal of project management*, v. 24, n. 4, pages 358–370.
- Almeida, T., Oran, A.C., Santos, G. e Conte, T. (2017). Relatório técnico: Como desenvolvedor quero utilizar user story para representar os requisitos que levam à definição do MVP e criação de Mockups. TR-USES-2017-0014. Disponível em: <http://uses.icomp.ufam.edu.br/relatorios-tecnicos/>.
- Basili, V. R. and Rombach, H. D. (1988). The TAME project: Towards improvement oriented software environments. *IEEE Transactions on software engineering*, v. 14, n. 6, pages 758–773.
- Beck, K. (1999). Embracing change with extreme programming. *Computer*, v. 32, n. 10, pages 70–77.
- Breitman, K. and do Prado Leite, J. C. S. (2002). Managing *user stories*. In *International Workshop on Time-Constrained Requirements Engineering*, page 168.
- Breitman, K. and do Prado Leite, J. S. (2000). Scenario-based software process. In *Proceedings Seventh IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS 2000)*, pages 375–380.
- Caroli, P. (2015). Direto ao ponto: criando produtos de forma enxuta. Casa do Código.
- Carrera Barroso, A., Solitario, J. J. and Iglesias Fernandez, C. A. (2012). Behaviour driven development for multi-agent systems. In: *International Workshop on Infrastructures and Tools for Multiagent Systems*, page 107-120.
- Cooper, B. and Vlaskovits, P. (2010). *The Entrepreneur’s Guide to Customer Development*. Paperback.
- Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly*, pages 319–340.
- Duc, A. N. and Abrahamsson, P. (2016). Minimum viable product or multiple facet product? the role of mvp in software startups. In *International Conference on Agile Software Development*, pages 118–130. Springer.
- Höst, M., Regnell, B. and Wohlin, C. (2000). Using students as subjects—a comparative study of students and professionals in lead-time impact assessment. *Empirical Software Engineering*, pages 201–214.
- Keogh, E. (2010). BDD: A Lean Toolkit. In: *Proc. of Lean Software & Systems Conf.*.
- Khanh, N. T., Daengdej, J. and Arifin, H. H. (2017). Human stories: A new written technique in agile software requirements. In *Proceedings of the 6th International Conference on Software and Computer Applications (ICSCA 2017)*, pages 15–22.

- Laitenberger, O. and Dreyer, H. M. (1998). Evaluating the usefulness and the ease of use of a web-based section data collection tool. In Proceedings of the 5th International Symposium on Software Metrics, pages 122 - 132.
- Lanubile, F., Mallardo, T. and Calefato, F. (2003). Tool support for Geographically Dispersed Inspection Teams. In Software Process Improvement and Practice, pages 217 – 231.
- Lazăr, I., Motogna, S. and Pârv, B. (2010). Behaviour-driven development of foundational UML components. Electronic Notes in Theoretical Computer Science, pages 91 – 105.
- Luna, E. R., Panach, J. I. and Grigera, J. *et al.* (2010). Incorporating usability requirements in a test/model-driven engineering approach. Journal of Web Engineering, v. 9, n. 2, pages 132 – 156.
- Moogk, D. R. (2012). Minimum viable product and the importance of experimentation in technology startups. Technology Innovation Management, v. 2, n. 3, page 23.
- Münch, J., Fagerholm, F., Johnson, P., Pirttilahti, J., Torkkel, J. and Jäärinen, J. (2013). Creating minimum viable products in industry-academia collaborations. In Lean Enterprise Software and Systems, pages 137-151.
- North, D. (2006). “Introducing bdd”, <http://dannorth.net/introducing-bdd/>, Março.
- Paternoster, N., Giardino, C., Unterkalmsteiner, M., Gorschek, T. and Abrahamsson, P. (2014). Software development in startup companies: A systematic mapping study. In Information and Software Technology, v. 56, n. 10, pages 1200–1218.
- Ries, E. (2014). A startup enxuta, Leya, 1ª edição.
- Silva, T. R. (2016). Definition of a behavior-driven model for requirements specification and testing of interactive systems. In Requirements Engineering Conference (RE), 2016 IEEE 24th International, pages 444–449. IEEE.
- Solis, C. and Wang, X. (2011). A study of the characteristics of behaviour driven development. In Software Engineering and Advanced Applications, pages 383–387.
- Sutcliffe, A. (1998). Scenario-based requirements analysis. In: Requirements engineering, v. 3, n. 1, pages 48-65.
- Travassos, G., Shull, F., Fredericks, M. and Basili, V. R. (1999). Detecting defects in object-oriented designs: using reading techniques to increase software quality. In *ACM Sigplan Notices*, v. 34, n. 10, pages 47–56. ACM.
- Zearaoui, A., Bougroun, Z., Belkasmi, M. G. and Bouchentouf, T. (2013). *User stories* template for object-oriented applications. In Innovative Computing Technology (INTECH), 2013 Third Intl. Conf. on, pages 407–410. IEEE.