

Comunicação de requisitos em quadros kanban virtuais

Jullia Saad¹[0000-0003-3581-0233], Priscila Portela Costa²[0000-0003-4087-9585],
and Luciana A. M. Zaina¹[0000-0002-1736-544X]

¹ Universidade Federal de São Carlos (UFSCar), Campus Sorocaba,
Rdv. João L Santos, Km 110, Sorocaba, SP, Brasil {julliasaad,lzaina}@ufscar.br

² Crunchyroll priscila.portela.c@gmail.com

Resumo Quadros kanban virtuais permitem reunir informações sobre tarefas e requisitos do projeto, funcionando como uma das principais fontes de visualização sobre o desenvolvimento do projeto de software. Entretanto, muitas vezes alguns recursos não são bem compreendidos pelo time, o que pode acarretar problemas de comunicação entre o kanban virtual e o time. O objetivo deste artigo é apresentar uma proposta de funcionalidades que visam melhorar a comunicação sobre requisitos de software a partir do uso de um quadro kanban virtual. Para atingir o objetivo proposto, foram realizadas sessões de grupos de foco com 10 desenvolvedores de software para identificar as dificuldades que eles enfrentam diariamente utilizando quadros kanban virtuais. Em particular, observou-se dificuldades de acompanhamento das mudanças no quadro e o entendimento das tarefas de outros membros do time. A partir da análise dos dados coletados nos grupos de foco, foram propostas funcionalidades de visualização de atualizações e filtros de atividades. Essas funcionalidades são genéricas e podem ser aplicadas em diferentes ferramentas de quadro kanban virtual. Uma avaliação das funcionalidades propostas foi conduzida com 49 participantes. Os resultados da avaliação indicaram que houve uma melhora da comunicação de requisitos a partir do uso de um quadro kanban com as funcionalidades propostas. Em particular, a visualização explícita do progresso e das atividades do projeto contribuiu para que os participantes entendessem corretamente os requisitos apenas pelo uso do quadro kanban.

Keywords: Requirements engineering · Virtual kanban board · Requirements communication

1 Introdução

O quadro kanban se tornou uma ferramenta muito importante para times ágeis de software pela sua capacidade de reunir todas as informações necessárias a respeito do estado das tarefas de um projeto. Além disso, o quadro kanban também funciona como a principal forma de visualização do desenvolvimento de um projeto em sua totalidade [9].

O kanban nasceu como um quadro físico, sendo necessário um espaço adequado para que ele fosse montado e também a presença de todos os membros do time no mesmo espaço [4]. No entanto, em sincronia com o uso do quadro kanban por times de software, surgiram diversas ferramentas para que pudessem ser gerenciados virtualmente. Assim, houve uma adaptação do uso do quadro ao dinamismo de um time de desenvolvimento de software [9]. O quadro kanban virtual também facilita o acesso a informação, independente da localidade dos desenvolvedores.

Ao longo do tempo, as ferramentas kanban virtuais foram se tornando mais completas e complexas devido aos inúmeros recursos agregados, como, por exemplo sua integração com repositórios de códigos de projeto. Surgiram então as chamadas ferramentas de *Application Life-cycle Management* (ALM). O ALM é um processo contínuo de gerenciamento de todas as atividades que acontecem dentro de um projeto, envolvendo também processos relacionados a construção de um software e a manutenção do mesmo [7]. As ALMs reúnem todas as atividades, artefatos e membros das equipes de software.

Ao trabalhar em grupos, as pessoas necessitam estar conscientes sobre as atividades relacionadas ao grupo e outras relacionadas ao contexto de trabalho. No trabalho presencial, essa conscientização pode ocorrer naturalmente através da interação no dia-a-dia com os outros membros da equipe. No trabalho remoto, essa conscientização de grupo muitas vezes é parcial ou deficiente [7]. Tanto as ferramentas ALMs como as ferramentas que focam apenas no gerenciamento de quadros kanban virtuais são responsáveis por auxiliar na conscientização do time, principalmente no trabalho de times distribuídos [7].

Com o aumento da complexidade das ferramentas utilizadas pelo time e o quadro possuindo cada vez mais diferentes tipos de informações, problemas na comunicação sobre o projeto podem surgir [2,8]. Um dos problemas que podem aparecer está relacionado ao entendimento de requisitos de software que são apresentados no quadro kanban. O não entendimento sobre requisitos podem levar a uma visão não clara do projeto, causando como consequência implementações técnicas errôneas. Além disto, pode também causar a falta de entendimento adequada sobre os papéis existentes no time, impactando negativamente na colaboração entre seus membros [2].

O objetivo deste artigo é propor funcionalidades que visam melhorar a comunicação de requisitos de software a partir do uso de um quadro kanban virtual. Como estratégia para atingir este objetivo, primeiramente foi conduzido um estudo exploratório para identificar quais eram as principais dificuldades que desenvolvedores de software enfrentavam ao usar quadros kanban virtuais. A partir dos resultados desse estudo, foram propostas duas funcionalidades genéricas que podem ser aplicadas em diferentes ferramentas de gerenciamento de quadros kanban virtuais, que são: (i) área de atividades e (ii) área de filtros. Essas duas novas funcionalidades foram implementadas em forma de protótipos e avaliadas por 49 participantes. A avaliação revelou que as melhorias propostas permitiram ter uma visualização explícita do progresso e das atividades do projeto e con-

tribuíram para que os participantes entendessem corretamente os requisitos do projeto.

2 Fundamentos e trabalhos relacionados

Esta seção apresenta os fundamentos e trabalhos relacionados sobre quadros kanban e comunicação de requisitos.

2.1 Quadros kanban

O Kanban (com K maiúsculo) é um método evolucionário que utiliza o quadro kanban (k minúsculo) com outras ferramentas de visualização para introduzir as ideias do *Lean* no desenvolvimento de software [1]. O quadro kanban se tornou uma ferramenta importante para times ágeis de software pela sua capacidade de reunir todas as informações relevantes sobre o estado das tarefas do projeto. Assim, o quadro funciona como a principal forma de visualização do desenvolvimento do projeto [9].

Nas últimas décadas, surgiram diversas ferramentas para que os quadros kanban pudessem ser gerenciados virtualmente e adaptados à dinâmica de funcionamento dos times de software [9]. Trello³, Asana⁴ e Jira⁵ são alguns exemplos de ferramentas de quadro kanban amplamente utilizadas. Neste artigo, o termo quadro kanban refere-se a versão virtual que é utilizada pelos times de software.

No quadro kanban, as tarefas são representadas por cartões (i.e., *post-it*) e os status são indicados de acordo com cada região do quadro que o cartão está inserido, rotuladas comumente de “A fazer”, “Fazendo” e “Feito” [4]. Tanto o nome das colunas como a quantidade é variável de projeto para projeto. O quadro kanban ajuda a sinalizar visualmente as tarefas e a delimitar o WIP (*Work in Progress*) (i.e., tarefas ativamente sendo trabalhadas) [4].

O principal elemento visual de um quadro kanban virtual são os cartões, responsáveis por representar as tarefas do projeto. Nos cartões, podem existir descrições, etiquetas, *checklists* e outros, dependendo da ferramenta que é utilizada para gerenciar o quadro. As etiquetas são elementos visuais flexíveis e podem ter diferentes significados (e.g., nomes de diferentes times, projetos, etc.), dependendo do que foi estabelecido pelo time.

2.2 Comunicação de requisitos

A comunicação de requisitos é um processo de troca de informações sobre requisitos do software em construção entre todos os *stakeholders*. Requisitos que foram inicialmente elicitados devem ser comunicados entre todos os envolvidos, assim como suas eventuais mudanças ao longo do projeto [2].

³ <https://trello.com/>

⁴ <https://asana.com/>

⁵ <https://www.atlassian.com/br/software/jira>

Em relação à comunicação de requisitos, existem os conceitos de conhecimentos implícito e explícito. O conhecimento implícito diz respeito ao que não foi especificado de forma concreta, ou seja, suposições, opiniões e valores. Já o conhecimento explícito diz respeito à interpretação de especificações explícitas, como requisitos, documentos e manuais [3]. A comunicação de requisitos visa alcançar um entendimento compartilhado explícito do projeto, ou seja, um entendimento compartilhado dos seus requisitos. Times ágeis tem como objetivo utilizar a menor quantidade possível de documentação [13]. Muitas vezes o quadro kanban utilizado pelo time é a única documentação disponível. Desta maneira, o entendimento implícito pode se tornar muito mais comum [3] e causar ruídos na comunicação de requisitos, uma vez que estes não estão visíveis para o time.

2.3 Trabalhos relacionados

Liskin & Fagerholm [8] destacam que uma boa comunicação interna de times de software e tornar os requisitos explícitos são fatores que afetam o uso de artefatos de requisitos (e.g., *stories* e *tasks*). Os autores relatam um estudo em que os membros de um time de software consideravam que apenas uma boa comunicação interna era suficiente para o entendimento das *stories* e *tasks* de um projeto. No entanto, no estudo foram detectados entendimentos incorretos do time em relação aos requisitos. Deste modo, os autores ressaltam a importância de comunicar os requisitos no nível mais concreto possível [8]. Os autores concluem que quando os artefatos de requisitos são exibidos abertamente para todos, eles podem representar de maneira eficaz e visível os requisitos do projeto.

Bjarnason et al. [2] apontam que a falta de entendimento dos papéis que outros membros do time executam no projeto e visão não clara do objetivo macro do projeto afetam a comunicação de requisitos em times de software. Segundo os autores, a falta de entendimento sobre o propósito e responsabilidade de um membro traz dificuldades na compreensão geral sobre projeto e afeta diretamente o entendimento dos requisitos. Os autores ainda ressaltam que uma visão não clara do objetivo geral do projeto pode levar a conflitos com problemas menores, como implementações técnicas, ao invés de uma comunicação construtiva sobre como alcançar o objetivo comum.

Raith et al. [11] mostram que uma das desvantagens no uso de ALMs é a dificuldade de encontrar mudanças específicas no quadro de tarefas do time quando o conteúdo foi modificado recentemente (e.g., a descrição de uma *user story*). Os autores ainda sugerem como alternativa para contornar a dificuldade de encontrar mudanças específicas no quadro kanban, a implementação de uma visão mais detalhada do progresso do projeto nessas ferramentas.

Katsma et al. [6] afirmam que o uso de quadros virtuais impactam negativamente na comunicação e discussão entre os membros de um time. Karras et al. [5] investigam como customizações no quadro de tarefas impactam na sua compreensão. Os autores propuseram três customizações no quadro (i.e., estrutura, cor e tamanho dos *cards*) e obtiveram evidências de que apenas a modificação na estrutura foi substancialmente benéfica na compreensão do quadro.

Os estudos anteriores apresentem alguns dos problemas que são enfrentados em relação ao uso de quadros kanban por times de software. Contudo, até o momento não foram encontrados estudos que avaliassem o impacto da visualização explícita do progresso e das atividades do projeto para os desenvolvedores, conforme sugerido por Raith et al. [11]. Ademais, também não foram encontrados estudos avaliando como a visualização explícita do progresso do projeto pode gerar melhorias da comunicação de requisitos para os desenvolvedores de software.

3 Estudo exploratório

Foram conduzidos grupos de foco com desenvolvedores de software. Grupos de foco são normalmente indicados para investigar problemas que são compartilhados pelos participantes e são realizados como uma entrevista em grupo, orientada por um facilitador [10]. O objetivo deste estudo foi entender o contexto, necessidades e problemas que os desenvolvedores enfrentam no seu dia-a-dia de trabalho com ferramentas virtuais para gerenciamento de quadros kanban.

3.1 Planejamento e execução

Foram realizadas duas sessões de grupos de foco no mês de setembro de 2019. Na primeira sessão participaram 6 indivíduos e na segunda 4, totalizando 10 desenvolvedores de software que atuam no mercado. A discussão foi pautada por um roteiro elaborado previamente, com 5 tópicos principais: (i) Rotina de trabalho em relação ao uso da ferramenta kanban virtual, (ii) Procedimentos a seguir quando *bugs* são detectados, (iii) Discussão sobre a relação entre a ferramenta e a visão geral do projeto, (iv) Integrações utilizadas na ferramenta, e (v) Sugestões de funcionalidades. As sessões tiveram o áudio gravado.

Primeiramente, o facilitador orientou os participantes sobre o roteiro e objetivo da atividade. Logo após, cada participante respondeu o Termo de Consentimento Livre e Esclarecido (TCLE)⁶ e um questionário de perfil visando obter informações sobre o ambiente da empresa e do time que trabalhavam⁷. Em seguida, o facilitador apresentou um cenário⁸ no qual o participante deveria se imaginar no papel de desenvolvedor de software que, juntamente com seu time, utiliza um quadro kanban para o gerenciamento das atividades de um projeto. A partir deste cenário, os participantes foram orientados a desenhar em uma folha sulfite um quadro kanban que representasse esse cenário. O objetivo de desenhar o quadro era ajudar os participantes a lembrarem dos quadros que utilizam no dia-a-dia de trabalho e também possíveis problemas enfrentados. Após o desenho dos quadros pelos participantes, o facilitador iniciou a discussão das questões propostas.

⁶ <https://bit.ly/tcle-focus-groups>

⁷ <https://bit.ly/questionario-focus-groups>

⁸ <https://bit.ly/roteiro-focus-group>

3.2 Análise e resultados

Questionário aplicado aos desenvolvedores de software: os resultados apontaram que 6 participantes possuíam até 3 anos atuando no mercado como desenvolvedores e 4 participantes possuíam 6 ou mais anos de experiência como desenvolvedores. Os participantes trabalhavam em empresas com tempos de vida variados: 3 participantes trabalhavam em empresas recém criadas, com até 2 anos no mercado, 4 em empresas com mais de 3 anos, e 3 participantes trabalhavam em empresas mais consolidadas, com mais de 10 anos de atuação no mercado. Das empresas citadas, 7 possuem produtos ou serviços já lançados no mercado. As empresas com menos tempo de atuação possuíam, segundo os participantes, menos funcionários em comparação às empresas mais consolidadas.

A quantidade de membros por projeto e a forma de trabalho também variavam. Entre os 10 participantes, 5 reportaram trabalhar em times com até 5 pessoas. Apenas 1 dos participantes relatou que seu trabalho era parcialmente presencial. Para os demais, o trabalho é estritamente presencial (5 pessoas) ou estritamente remoto (4 pessoas). Em relação à composição do time, excetuando-se o papel do desenvolvedor, os principais papéis existentes nos times eram: (i) *Designers* (UX/UI), (ii) *Product Owner*, (iii) *Tester* e (iv) *Scrum Master*.

As abordagens mais citadas pelos participantes como utilizadas pelos times de software em que atuavam foram o *Scrum* e o *Kanban*. Dentre os participantes, 6 relataram que eram cobrados em relação à velocidade de desenvolvimento. Trabalhar em um ambiente dinâmico, no qual os requisitos mudam constantemente foi mencionado por 5 participantes.

Sessões de grupo de foco: Depois de realizadas as duas sessões de grupo de foco, os áudios foram transcritos para o formato texto manualmente. Em seguida, as duas primeiras autoras realizaram a leitura completa de todas as transcrições e tabularam os relatos de acordo com os tópicos discutidos nos grupos de foco. A cada tópico foram associados problemas e sugestões relatados pelos participantes. Os resultados das análises foram reavaliados e refinados pela terceira autora do artigo.

Entre os 10 participantes, 8 relatam que seus times utilizam o *Jira* como a principal ferramenta para gerenciamento das tarefas. Examinando os artefatos gerados pelos participantes, observou-se que haviam alguns quadros mais complexos que outros, possuindo mais estágios (i.e., colunas no quadro) e divisões de times. Verificou-se também o uso de divisões horizontais para diferenciar os tipos de atividades (*back-end* e *front-end*), as *user stories* e, eventualmente, atividades relacionadas a manutenção de *bugs*.

Os resultados obtidos demonstraram que dentre os quadros apresentados pelos participantes, alguns possuíam divisões horizontais por *stories* e, dentro de cada *story*, divisões verticais em colunas contendo diversas *tasks*. Segundo alguns dos participantes, a criação das *user stories* fica a cargo de um *Product Owner*, e

geralmente as tarefas são quebradas e pontuadas pelos próprios desenvolvedores. Os quadros ilustrados pelos participantes estão disponíveis para visualização⁹.

Vários participantes relataram ter dificuldades em entender o objetivo do projeto e também da *sprint* a partir do quadro que usam. Ainda foram relatadas dificuldades no entendimento do quadro de outras áreas e a relação das tarefas de outras áreas com a própria tarefa (problema 1). Essas dificuldades podem estar relacionadas ao uso de diferentes quadros pelas diversas equipes de um mesmo projeto (e.g., um quadro para *front-end* e outro para *back-end*), como relata um dos participantes: “difícilmente é possível ter um entendimento do quadro de outra pessoa ou área” (problema 2). As muitas segmentações horizontais presentes em alguns quadros descritos pelos participantes também trazem dificuldades ao entendimento dos objetivos do projeto: “muita segmentação [horizontal] no quadro não permite que o desenvolvedor tenha uma visão geral do projeto”. Os participantes ainda relatam que preferem quadros simples: “gosto de ferramentas que simples e que não precisem de um esforço para utilizar e aprender”.

Os participantes reportaram ter que empregar um esforço para identificação, compreensão dos requisitos do projeto e acompanhamento de todas as mudanças apenas pela visualização do quadro. Muitas vezes é necessário tirar dúvidas com quem descreveu a tarefa, normalmente o *Product Owner*. A relação de tarefa com outros elementos, como, por exemplo de qual *user story* é, de qual projeto, se possui alguma outra tarefa bloqueando, também foi uma dificuldade apontada pelos entrevistados.

4 Proposta de melhorias para quadros kanban

A partir dos resultados obtidos com a análise das sessões de grupos de foco, foram elaboradas propostas para melhorias do quadro kanban.

Primeiro, identificou-se a necessidade de se acompanhar de maneira mais direta as mudanças no quadro. Esta necessidade foi elencada com base na dificuldade que os participantes reportaram em acompanhar as mudanças no quadro e a menção a frequentes mudanças nos requisitos (problema 1). Também identificou-se dificuldade dos desenvolvedores no entendimento de tarefas de outras áreas e também a relação das tarefas de outras áreas com a própria tarefa (problema 2).

Para minimizar os problemas relacionados ao acompanhamento das mudanças no projeto (problema 1) e entendimento de outras áreas (problema 2) são propostas as seguintes melhorias: (i) uma área contendo um resumo de todas as atividades, incluindo a data de execução e o responsável pela alteração, como mostra a Figura 1 e (ii) uma área para possibilitar a aplicação de filtros no quadro, que podem ser feitos de acordo com etiquetas associadas às tarefas, por integrante ou por status de realização da tarefa, como mostra a Figura 2.

As melhorias propostas neste trabalho não estão relacionadas a uma ferramenta específica e podem ser aplicadas em diferentes ferramentas para a criação

⁹ <https://bit.ly/quadros-focus-groups>

de quadros kanban virtuais. Contudo, para uma melhor visualização da proposta, estas são demonstradas no formato de protótipo de interfaces e detalhadas a seguir.

O item (i) visa fornecer um panorama de todas as atividades que ocorreram no quadro do projeto, baseada na dificuldade dos participantes em acompanhar as mudanças no quadro. O item (i) se baseia na sugestão de Raith et al. [11] a respeito da implementação de uma visão mais detalhada do progresso do projeto nessas ferramentas. Esta melhoria também pode contribuir positivamente na comunicação de requisitos, tornando-os mais explícitos para os desenvolvedores.

O item (ii) foca na obtenção de uma visão mais precisa de diferentes partes do projeto com o filtro por etiquetas as quais identificam visualmente a divisão de papéis no time. O item (ii) se baseia na dificuldade em relação ao entendimento das tarefas de outros membros e como elas se relacionam com a própria tarefa. Esta é uma dificuldade que foi apontada por Bjarnason et al. [2]. Segundo os autores, a falta de compreensão sobre a responsabilidade de um membro traz dificuldades para que o time obtenha um entendimento geral do projeto e, conseqüentemente, isto afeta diretamente o entendimento dos requisitos.

Nas ferramentas kanban citadas pelos participantes como usadas em seus projetos (e.g. Trello, Jira), não foi encontrado um resumo das atividades que aparecesse sempre para o usuário, o que seria o correspondente ao item de melhoria (i). O item de melhoria (ii), que corresponde à área de filtros por etiquetas, foi idealizada a partir dos relatos dos desenvolvedores, uma vez que a possibilidade de filtrar por 1 ou mais itens faz com que a melhoria proposta seja flexível o suficiente para atender necessidades como busca por integrantes, tarefas em atraso, dentre outras. Em particular, no Trello, é possível abrir a janela de atividades, no entanto, é necessário clicar no item de menu para encontrar essas atividades. Conforme relatado na Seção 3.2, a dificuldade em encontrar certas funcionalidades faz com que os desenvolvedores afirmem que elas não existem na ferramenta. Esta afirmação foi reportada por participantes que usavam a ferramenta Jira. Observa-se que a ferramenta Jira, por possuir complexidade em suas customizações, podem trazer este grau de dificuldade. Desta maneira, deixar a funcionalidade de filtros mais visível nos quadros kanban pode trazer melhorias na busca de informações.

5 Avaliação com usuários

5.1 Planejamento e execução

Um protótipo navegável¹⁰ foi desenvolvido na ferramenta Marvel¹¹, a partir das propostas de funcionalidades apontadas na Seção 4. O quadro foi preenchido com tarefas relacionadas à criação de um *site* de cursos, com 3 divisões de papéis que representam a divisão do time (i.e., *backend*, *front-end*, *design*, *tester*) e 4 integrantes fictícios (i.e., Teresa, Marcela, Marco e José).

¹⁰ <https://marvelapp.com/9g624a6/screen/62946118>

¹¹ <https://marvelapp.com/>

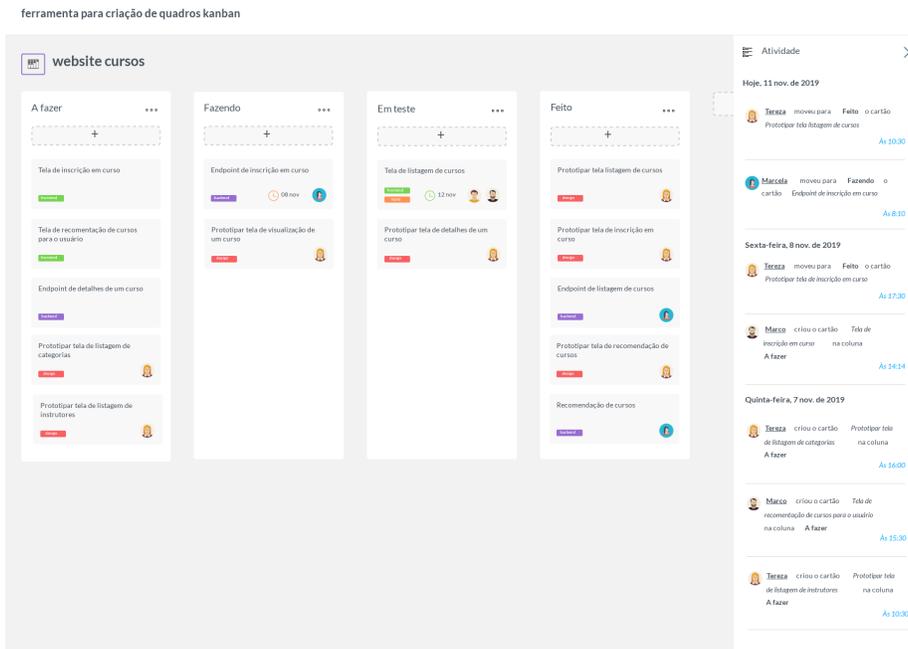


Figura 1. Funcionalidade proposta - Área de atividades (lado direito).

Planejou-se uma avaliação presencial onde os participantes responderiam primeiro um questionário demográfico e depois executariam tarefas no protótipo a partir de cenários de uso. Os dados de interação dos participantes com o protótipo foram coletados pela ferramenta Maze¹² juntamente com questionários. Foram planejadas questões as quais eram apresentadas após as interações. Essas questões visavam avaliar o entendimento que o participante tinha sobre as interações de um quadro kanban e a facilidade de interação e execução das tarefas no quadro com as novas funcionalidades propostas.

O questionário demográfico visou medir a familiaridade dos participantes com trabalhos em grupo e experiência de trabalho, bem como sua familiaridade com quadro kanban e as ferramentas utilizadas. Usando uma descrição hipotética, foi também adicionada uma pergunta sobre o entendimento global do projeto de um novo membro recém-chegado ao time, usando como fonte de informação apenas o quadro kanban proposto. O conteúdo do questionário está disponível para acesso¹³.

As atividades propostas foram divididas em duas tarefas, cada uma referente a uma funcionalidade de melhoria do quadro (i.e., área de resumo das atividades e aplicação de filtros). Na primeira etapa (Tarefa 1), ao olhar o quadro kanban

¹² <https://maze.design/>

¹³ <https://bit.ly/questionario-user-evaluation>

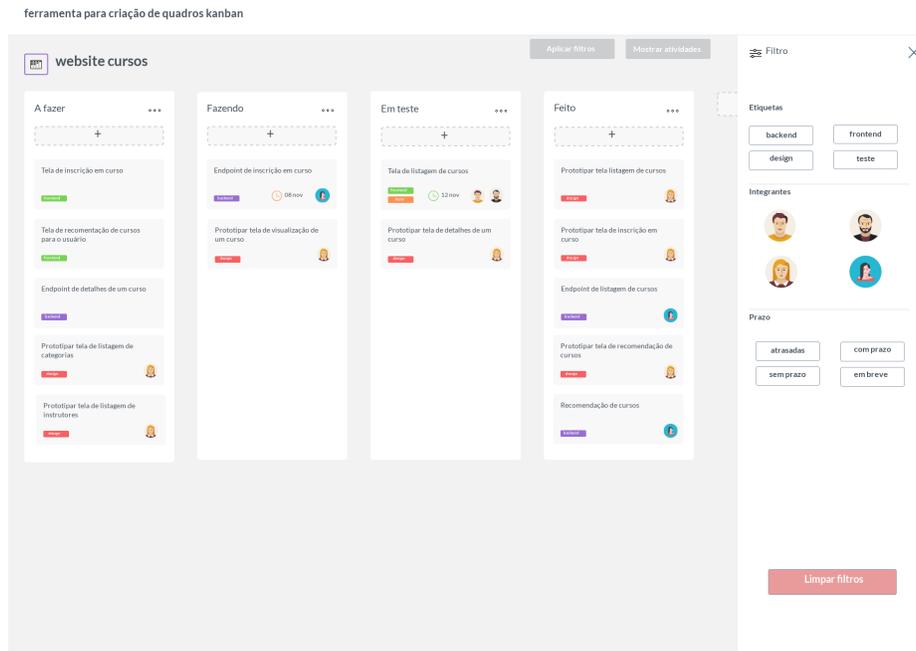


Figura 2. Funcionalidade proposta - Área de filtros (lado direito).

com a área de atividades, os participantes foram questionados sobre (i) quem é a pessoa mais sobrecarregada do time, também foi solicitada uma interação visando a (ii) alocação de tarefa a um participante e (iii) papel de um integrante no time. Por fim, perguntou-se sobre a facilidade de achar as informações requisitadas. Na segunda etapa (Tarefa 2), foi solicitada a (i) interação com a aba de filtros, para filtragem de tarefas de acordo com o time, (ii) andamento das tarefas entre os times de *design* e *front-end* e (iii) a facilidade em interagir com a funcionalidade proposta e o seu impacto no entendimento das atividades. As perguntas realizadas nas atividades estão disponíveis para acesso¹⁴.

Participaram da sessão de avaliação 50 alunos do curso de Graduação em Ciência da Computação da Universidade Federal de São Carlos. Primeiramente, foi apresentado aos participantes o objetivo e cada uma das etapas da avaliação. Dos 50 participantes, 1 teve as respostas invalidadas, pois o fluxo proposto pela avaliação não foi realizado. As respostas de 49 participantes foram utilizadas para análise.

5.2 Análise e resultados

Questionário demográfico: Todos os participantes apontaram estar familiarizados com a ferramenta Trello, possuem entre 21 e 24 anos e a maioria (aproxima-

¹⁴ <https://bit.ly/questiones-user-evaluation>

damente 86%) não possui experiência profissional, o que mostra que os quadros kanban utilizados foram associados a projetos de disciplinas da graduação ou a projetos pessoais. Outras ferramentas de quadro kanban foram mencionadas por 10% dos desenvolvedores, como *Jira* e *GitHub Projects*.

Apesar das menções ao Trello como ferramenta utilizada para construção dos quadros virtuais, mais da metade dos participantes (aproximadamente 68%) se autoavaliaram com pouca familiaridade com quadros kanban.

Cenários de uso: A análise dos resultados foi realizada paralelamente pelo primeiro e segundo autor. Na sequência, os resultados prévios foram reavaliados pela terceira autora em conjunto com as demais, gerando assim, um refinamento nos resultados. As 49 respostas dos participantes foram analisadas. As questões que possuíam certo ou errado foram analisadas quantitativamente, e questões abertas qualitativamente. As informações obtidas no questionário demográfico também foram utilizadas para avaliar se a experiência prévia com quadros kanban virtuais teve alguma influência (positiva ou negativa) na realização das atividades e no entendimento das informações apresentadas.

Todos os 49 participantes considerados no estudo conseguiram cumprir as tarefas, e não foi registrada nenhuma desistência. A familiaridade com quadros kanban não impactou na dificuldade em realizar as tarefas propostas, como pode ser visto na Tabela 1. Os participantes deveriam apontar sua familiaridade com quadros kanban em uma escala de 1 (pouco familiar) a 4 (muito familiar) e a facilidade na realização de tarefas em uma escala de 1 (pouca facilidade) a 5 (muita facilidade).

Tabela 1. Relação entre a facilidade de realizar as tarefas propostas na avaliação e a familiaridade com kanban respondida em questionário demográfico.

Familiaridade com kanban (Autoavaliação)	Proporção de alunos	Facilidade em realizar a Tarefa 1 (média)	Facilidade em realizar a Tarefa 2 (média)
1	2%	5,0	5,0
2	62%	4,6	4,7
3	28%	4,7	4,9
4	8%	4,8	5,0

Em relação ao entendimento do papel de um integrante no time, questão apresentada na Tarefa 1, observou-se que não houve relação entre as respostas e a experiência dos participantes, como mostra a Tabela 2. Os grupos que responderam de forma correta e incorreta possuem médias semelhantes de familiaridade com ferramentas kanban conforme relatado no questionário demográfico. 54% dos participantes relataram utilizar as informações na área de atividades para apontar o papel do integrante do time, conforme solicitado na Tarefa 1.

Para analisar as respostas da pergunta aberta 6 (i.e., *Explique brevemente sobre o escopo/objetivo/andamento do projeto e outras informações que você conseguiu obter observando o quadro kanban apresentado*), as respostas de cada

Tabela 2. Análise das respostas à questão: De acordo com seu entendimento do quadro, qual é o papel do Marco no time?

	Utilizou informações da área de atualização		Familiaridade com kanban (média)
	Sim	Não foi possível interpretar	
Marco como PO	38%	0%	2,5
Marco como outro papel	16%	46%	2,4

um dos participantes foram tabeladas e divididas em três colunas: (i) objetivos do projeto, (ii) requisitos do projeto e (iii) outras informações. Se necessário, o pesquisador poderia colocar suas observações em uma coluna chamada “comentários do pesquisador”. A quantidade de requisitos apontada pelos respondentes na questão aberta foram classificadas em três diferentes categorias (i) inferiu maior parte dos requisitos (4 ou mais requisitos citados corretamente), (ii) inferiu alguns requisitos (3 ou 2 requisitos citados corretamente) e (iii) inferiu 1 ou nenhum requisito. A partir dessa categorização, foi possível obter a quantidade de respostas que pertenciam a cada uma das categorias. Dos 49 participantes, 16 conseguiram inferir a maior parte dos requisitos, 17 conseguiram inferir alguns requisitos e 16 inferiram apenas 1 ou nenhum dos requisitos do projeto apresentado.

5.3 Ameaças à validade

As ameaças levantadas à validade foram mitigadas conforme descrito a seguir.

A ferramenta Maze¹⁵, utilizada para construir os cenários de avaliação, poderia não ser familiar aos usuários, gerando dúvidas sobre o uso da ferramenta e impondo dificuldades não associadas às tarefas propostas. Com o intuito de minimizar esse problema, foi realizada uma seção de aquecimento (i.e, *warm up*) onde os participantes puderam realizar atividades para se familiarizarem com a ferramenta.

A quantidade de informações requeridas na última questão pode ter impactado negativamente o seu resultado, uma vez que os participantes podem ter focado em responder mais sobre um aspecto do que responder sobre todas as informações pedidas. Para minimizar esse problema, não houve limite de tempo para a resolução das questões e os participantes foram incentivados a tirarem quaisquer dúvidas sobre as questões propostas.

Sobre a avaliação ser realizada com estudantes, o estudo de Salman et al. [12] fornece evidências de que há poucas diferenças de desempenho entre estudantes e profissionais em relação a uma tarefa com a qual eles não possuem conhecimento prévio.

¹⁵ <https://maze.design/>

6 Discussões e conclusões

Este estudo partiu do objetivo de melhorar a comunicação de requisitos em quadros kanban virtuais através de propostas baseadas na literatura e nos problemas apontados por 10 desenvolvedores em um estudo exploratório. O estudo exploratório revelou necessidade de ter um acompanhamento direto as mudanças no quadro e o entendimento das tarefas de outros membros e suas relações com a própria tarefa. A proposta de melhoria elaborou duas funcionalidades a um quadro kanban virtual: uma área de atividades e uma área de filtros. Com essas funcionalidades, seria possível acompanhar as atualizações do quadro e personalizar visualizações de acordo com a necessidade e o interesse dos membros de um time.

Foi conduzida uma avaliação com 49 participantes e concluiu-se que, as duas novas funcionalidades, não adicionaram dificuldades de interação mesmo em participantes com pouca ou nenhuma familiaridade com quadros kanban virtuais. As informações apresentadas foram úteis para se detectar a função de um integrante do time, que não seria possível se não houvesse a área de atividades presente na proposta de funcionalidades. Em relação à comunicação dos requisitos, observou-se que aproximadamente 33% dos participantes não obteve sucesso em inferir quais os requisitos necessários para o sucesso do projeto. Dentre os restantes, aproximadamente 67% dos participantes conseguiu apontar com sucesso os requisitos, e também relataram que a área de atividades ajudou no entendimento do projeto. Em relação ao entendimento do projeto, a maior parte dos participantes (i.e, aproximadamente 82%) conseguiu inferir com sucesso o objetivo do projeto.

Os resultados da avaliação com usuários indicam que as propostas de melhorias para quadros kanban contribuíram para o entendimento dos requisitos de um projeto, trazendo evidências dos benefícios gerados pelo aumento da comunicação explícita de requisitos. A visualização explícita do progresso e das atividades do projeto para os desenvolvedores contribuiu para que eles entendam corretamente os requisitos do projeto. Este estudo demonstra evidências de que a visão mais detalhada do projeto, sugerida por Raith et al. [11] traz benefícios para o rastreamento de mudanças específicas no quadro. Este estudo também contribuiu propondo uma solução parcial para o problema relatado por Bjarnason et al. [2], a respeito da falta de entendimento sobre a responsabilidade de um membro do time e suas consequências para o entendimento dos requisitos.

Em trabalhos futuros, vê-se a necessidade da condução de uma avaliação com desenvolvedores de software que atuem no mercado de trabalho. Também é necessário avaliar quais papéis de um time se beneficiam mais das funcionalidades propostas.

Agradecimentos. O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001; e do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPQ) - Brasil, Processo 313312/2019-2.

Referências

1. Anderson, D.: Kanban: Successful Evolutionary Change for Your Technology Business. Blue Hole Press (2010), <https://books.google.com.br/books?id=RJ0VUkfUWZkC>
2. Bjarnason, E., Wnuk, K., Regnell, B.: Requirements are slipping through the gaps — a case study on causes effects of communication gaps in large-scale software development. In: 2011 IEEE 19th International Requirements Engineering Conference. pp. 37–46 (Aug 2011). <https://doi.org/10.1109/RE.2011.6051639>
3. Glinz, M., Fricker, S.A.: On shared understanding in software engineering: an essay. *Computer Science - Research and Development* **30**, 363–376 (Aug 2015). <https://doi.org/10.1007/s00450-014-0256-x>, <https://doi.org/10.1007/s00450-014-0256-x>
4. Hiranabe, K.: Kanban Applied to Software Development: from Agile to Lean. <https://www.infoq.com/articles/hiranabe-lean-agile-Kanban/> (2008), <https://www.infoq.com/articles/hiranabe-lean-agile-Kanban/>, Acessado em 27 Set., 2019
5. Karras, O., Klünder, J., Schneider, K.: Is task board customization beneficial? - an eye tracking study (2017), <https://arxiv.org/abs/1708.00275>
6. Katsma, C., Amrit, C., van Hillegersberg, J., Sikkel, K.: Can agile software tools bring the benefits of a task board to globally distributed teams? In: Oshri, I., Kotlarsky, J., Willcocks, L.P. (eds.) *Advances in Global Sourcing. Models, Governance, and Relationships*. pp. 163–179. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
7. Lanubile, F., Calefato, F., Ebert, C.: Group awareness in global software engineering. *IEEE Software* **30**(2), 18–23 (2013). <https://doi.org/10.1109/MS.2013.30>
8. Liskin, O., Schneider, K., Fagerholm, F., Münch, J.: Understanding the role of requirements artifacts in kanban. In: *Proceedings of the 7th International Workshop on Cooperative and Human Aspects of Software Engineering*. p. 56–63. CHASE 2014, Association for Computing Machinery, New York, NY, USA (2014). <https://doi.org/10.1145/2593702.2593707>, <https://doi.org/10.1145/2593702.2593707>
9. Nakazawa, S., Tanaka, T.: Prototype of Kanban Tool and Preliminary Evaluation of Visualizing Method for Task Assignment. In: *Proceedings - 2015 International Conference on Computer Application Technologies, CCATS 2015*. pp. 48–49. IEEE (2016). <https://doi.org/10.1109/CCATS.2015.21>
10. Preece, J., Rogers, Y., Sharp, H.: *Interaction Design: beyond human-computer interaction*. Wiley, Indianapolis, IN, 5 edn. (2019)
11. Raith, F., Richter, I., Lindermeier, R.: How project-management-tools are used in agile practice: Benefits, drawbacks and potentials. In: *Proceedings of the 21st International Database Engineering & Applications Symposium*. p. 30–39. IDEAS 2017, Association for Computing Machinery, New York, NY, USA (2017). <https://doi.org/10.1145/3105831.3105865>, <https://doi.org/10.1145/3105831.3105865>
12. Salman, I., Misirli, A.T., Juristo, N.: Are students representatives of professionals in software engineering experiments? In: *Proceedings of the 37th International Conference on Software Engineering-Volume 1*. pp. 666–676. IEEE Press (2015)
13. Sharp, H., Giuffrida, R., Melnik, G.: Information flow within a dispersed agile team: A distributed cognition perspective. In: Wohlin, C. (ed.) *Agile Processes in Software Engineering and Extreme Programming*. pp. 62–76. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)