

Uma Pesquisa Qualitativa sobre Características de Produtividade em Equipes de Desenvolvimento de Software

Adriana Silveira de Souza^{1,2}, Juliano Lopes de Oliveira¹, Daniel Ares Moraes²

¹Instituto de Informática – Universidade Federal de Goiás (UFG)
Alameda Palmeiras, Quadra D, Câmpus Samambaia – 74690-900

²Escola de Ciências Exatas e de Computação
Pontifícia Universidade Católica de Goiás (PUCGO)
Av. Universitária 1.440, Setor Universitário – 74605-010
Goiânia - Goiás - Brasil

{adriana,juliano}@inf.ufg.br, daniel.ares@hotmail.com

Abstract. *Developing software is a complex activity and highly dependent on the collaborative work of multidisciplinary teams. Team productivity is a determining factor for the success of software development. This work analyzes characteristics that influence the productivity of software teams. A survey evaluated the presence of these characteristics in teams that use the Scrum agile methodology and in teams that use other software methodologies. The research indicates that, in the point of view of software professionals, teams that use Scrum are superior in many of the characteristics that impact their productivity.*

Resumo. *Desenvolver software é uma atividade complexa e altamente dependente da colaboração entre equipes multidisciplinares. A produtividade da equipe é um fator determinante para o sucesso do desenvolvimento de software. Este trabalho analisa características que influenciam a produtividade de equipes de desenvolvimento de software. Uma pesquisa de campo avaliou a presença dessas características em equipes que utilizam a metodologia ágil Scrum e em equipes que utilizam outras metodologias de desenvolvimento de software. A pesquisa indica que, na visão dos profissionais de software, equipes Scrum são superiores em muitas das características que impactam a sua produtividade.*

1. Introdução

Atender a expectativas cada vez mais altas do consumidor é um desafio para organizações produtoras de software. Softwares cada vez mais complexos precisam ser gerados em tempos menores e com maior qualidade. Uma equipe de desenvolvimento de software motivada é essencial para alcançar esses resultados [Beecham et al. 2008].

Uma equipe é formada por pessoas que trabalham juntas em um espaço (concreto ou virtual) compartilhado, com o propósito de realizar um objetivo comum. Membros de uma equipe não realizam suas tarefas individualmente, mas precisam interagir em tarefas que se complementam [Gorla and Lam 2004, Scarnati 2001]. Trabalhar em equipe promove o aprendizado através da interação, resolução de problemas, diálogo, cooperação e colaboração. Um ambiente de equipe desenvolve o lado cognitivo através das interações sociais, com troca de experiências e ideias, gerando novos conhecimentos e experiências [Johnson and Johnson 1995].

Diversas características contribuem para os resultados apresentados por uma equipe de desenvolvimento de software, tais como o conhecimento, a experiência e o comportamento individual de cada membro da equipe, e o contexto organizacional em que a equipe trabalha [Oliveira 2017].

Este trabalho apresenta resultados de uma pesquisa qualitativa sobre características de equipes de software que impactam sua produtividade. A Seção 2 faz uma breve revisão da literatura sobre produtividade em equipes de software. A Seção 3 discute o trabalho em equipe na metodologia Scrum, amplamente utilizada na indústria de software, que foi usada neste trabalho como base para comparação e análise das características de produtividade. A Seção 4 descreve uma pesquisa de campo sobre estas características envolvendo profissionais de software. A Seção 5 apresenta os resultados desta pesquisa, que indicam que equipes Scrum são superiores em muitas dessas características.

2. Trabalho em equipe no desenvolvimento de software

O desenvolvimento de software é altamente dependente de fatores humanos e muitos dos problemas enfrentados em projetos de software estão associados às pessoas [DeMarco and Lister 2013]. Para o desenvolvimento de software com qualidade é necessário a colaboração de uma equipe constituída por diversos profissionais que devem coordenar suas tarefas, criar e seguir planos e manter uma boa comunicação.

Um fator crítico para o sucesso de uma equipe é a sinergia, que significa ações ou esforços simultâneos voltados para o mesmo objetivo. A sinergia, cria um ambiente em que todos podem contribuir para a criação de um ambiente de trabalho eficaz. Em um contexto profissional ela pode ser vista como a combinação de [Luca and Tarricone 2001]:

- Respeito: ter sentimento positivo de consideração, reconhecimento e estima pelas qualidades de cada pessoa e de toda a equipe.
- Cooperação: realizar trabalho em grupo, em um espaço compartilhado, somando os esforços e apoiando o outro na realização de suas tarefas.
- Lealdade: honrar seus compromissos com retidão e responsabilidade, nunca prejudicando a equipe, independente do que ocorrer.
- Comprometimento: respeitar os acordos, formais ou informais, realizando aquilo que foi combinado pela equipe, ainda que não concorde.

Esses atributos são essenciais para uma boa convivência entre indivíduos que formam uma equipe. Além disso, o sucesso de um projeto de software depende de três componentes inter-relacionados: pessoas, processos e tecnologias. As pessoas envolvidas no projeto seguem processos que definem quais tarefas são realizadas e aplicam tecnologias para realizar o trabalho. Essas pessoas precisam ter habilidades técnicas para compreender os processos e aplicar tecnologias adequadas. Todavia, a qualidade do produto será alcançada somente se as pessoas confiarem na equipe e estiverem motivadas para colaborar no trabalho de criação do software [Beecham et al. 2008].

2.1. Desafios para Equipes de Desenvolvimento de Software

A Figura 1 apresenta os principais desafios para a produtividade de equipes e indica que existe uma relação causal entre esses desafios. Esses desafios afetam todos os tipos de equipe e representam questões particularmente importantes para o trabalho de desenvolvimento de software.



Figura 1. Cinco grandes desafios para equipes [Lencioni 2015]

A confiança é o elemento essencial para que qualquer tipo de relação humana seja saudável. No trabalho em equipe, ela é o pilar que sustenta todas as características positivas das pessoas envolvidas em um projeto [Lencioni 2015]. A **ausência de confiança** ocorre quando os membros da equipe não estão dispostos a assumir seus erros, fraquezas e necessidades de ajuda, ou seja, não aceitam demonstrar que são vulneráveis. Atitudes que podem promover um bom relacionamento na equipe e aumentar a confiança são: transparência e coerência nas ações, saber ouvir, assumir falhas e dar *feedbacks* honestos e construtivos. Superar o desafio da ausência de confiança é a base para que uma equipe possa ter alto desempenho [Emelyanova et al. 2019].

A falta de confiança na equipe gera o **medo do conflito**, já que em um conflito normalmente alguém “ganha” e o outro “perde”. Porém, não precisa ser necessariamente assim. Conflitos podem gerar uma situação de ganho mútuo. Para isso é necessário que ele esteja no meio termo entre o conflito construtivo e o destrutivo. Ele não pode ser totalmente construtivo, porque pode gerar concordância cega e bajulação, e não pode ser totalmente destrutivo, gerando discussão gratuita e desnecessária, sem nenhum benefício para a equipe. Quando existe confiança na equipe, o medo do conflito diminui e os conflitos passam a ser construtivos, servindo como forma de crescimento e aprendizagem para todos os membros da equipe.

Quando a equipe busca, a todo custo, evitar o conflito, ela reduz a comunicação entre os membros e gera a **falta de comprometimento** com o trabalho. Comprometimento é diferente de consenso, ou seja, a equipe toda deve ter compromisso em realizar as tarefas planejadas, mesmo que nem todos concordem com elas. Sem o comprometimento dos membros da equipe com o planejamento feito para o projeto, as chances de alcançar o objetivo estabelecido diminuem. A falta de comprometimento pode ser resolvida com confiança e valorização de conflitos construtivos.

Se a equipe não consegue estabelecer o comprometimento com os planos do projeto, surge na equipe a **fuga da responsabilidade**, que é a situação em que cada membro evita se responsabilizar por seus atos. O senso de responsabilidade de cada membro em uma equipe deve ir além de apenas fazer a sua parte, e envolve se preocupar com as ações feitas pelos demais membros da equipe. Para minimizar a fuga de responsabilidade, é preciso fortalecer o comprometimento dos membros com a equipe.

A **falta de atenção aos resultados** é uma consequência dos desafios anteriores e ocorre quando os membros se distraem das metas da equipe, seja por problemas pessoais ou por possuírem outras prioridades. Metas autoexplicativas e fáceis de lembrar, associadas a tarefas diárias que tenham uma relação óbvia com a realização de metas, contribuem para superar esse desafio. Cada pessoa deve sentir que suas tarefas diárias ajudam a equipe a atingir seu objetivo final.

2.2. Motivação de Equipes de Desenvolvimento de Software

A motivação é um fator determinante para a produtividade individual e da equipe [Oliveira 2017]. Um colaborador desmotivado pode contaminar toda a equipe, pois sua baixa produtividade pode atrasar o projeto, gerando descontentamento pela sobrecarga de trabalho dos demais membros na tentativa de cumprir os prazos. Além disso, o resultado final do projeto não atende as expectativas, pois quando se trabalha sem motivação, dá-se pouca atenção à qualidade do produto.

Embora não esteja diretamente contemplada na pirâmide de desafios para equipes proposta por [Lencioni 2015] (Figura 1), a falta de motivação é um dos grandes desafios enfrentados pelos gestores de equipes de software e pode ser gerada por diversos fatores, tais como [Beecham et al. 2008, Ortiz and Farina 2015, Emelyanova et al. 2019]:

- Falta de oportunidade para crescimento profissional.
- Acúmulo de funções ou sobrecarga de trabalho.
- Falta de reconhecimento ao trabalho realizado na equipe.
- Dificuldade na gestão do tempo.
- Falta de participação nas decisões da equipe.
- Falta de preparação técnica e de treinamento para as atividades.
- Falta de competência ou de colaboração na equipe.

Uma organização que busca uma equipe motivada, deve despertar “motivação intrínseca” em seus funcionários, que significa realizar uma atividade ou tarefa exclusivamente pelo fato dela ser interessante ou agradável, e não por possíveis consequências. Já a “motivação extrínseca” é o ato de realizar algo em razão de fatores externos, como medo, imposição, dinheiro ou recompensas [Deci and M. 2000, Locke and Schattke 2019]. Tendo prazer na realização da tarefa, o colaborador irá se dispor a trabalhar com muito mais eficiência e capricho.

Um fator determinante para a motivação de uma equipe de desenvolvimento de software é a colaboração, que define uma forma de se trabalhar em equipe em que as pessoas agem juntas buscando completar os objetivos do projeto, e não objetivos individuais ou de subgrupos [Whitehead 2007]. A colaboração pode ser vista como a combinação de atividades que envolvem [Fuks et al. 1991]:

- Comunicação: troca de informação, conhecimentos, mensagens e negociação.
- Coordenação: gestão de pessoas, suas tarefas e recursos disponíveis.
- Cooperação: produção conjunta de uma dada tarefa em um espaço compartilhado.

As atividades de colaboração precisam ser realizadas contínua e iterativamente quando se trabalha em equipe. A comunicação discute e negocia as tarefas do projeto, criando compromissos que são gerenciados pela coordenação, que organiza as tarefas e

os possíveis conflitos existentes para facilitar a cooperação. Durante a cooperação é necessário renegociar situações imprevistas, o que novamente demanda comunicação. Isso gera uma percepção em cada membro que permite obter respostas para suas próprias ações e também para as ações dos demais participantes da equipe.

A colaboração é uma condição essencial para a motivação da equipe. Quando não há colaboração, a equipe se desmotiva e perde o foco nos objetivos compartilhados [Batra et al. 2017].

3. Trabalho em equipe no Scrum

O Scrum é uma metodologia de desenvolvimento de software representativa das metodologias ágeis, muito disseminadas na atualidade [Sutherland 2020]. A filosofia desse tipo de metodologia dá grande destaque para valores humanos. O desenvolvimento ágil não rejeita processos, ferramentas, documentações, negociações e planejamento, mas considera que tudo isso é secundário se estiver em conflito com as pessoas e suas interações no projeto. Alguns dos principais fatores que potencializam a produtividade no trabalho em equipe em projetos que utilizam o Scrum são [O’Connell and Molloy 2019]:

- **Familiaridade:** a filosofia Scrum incentiva a interdependência e a confiança entre os membros da equipe. Para fomentar a confiança, equipes Scrum tendem a ser estáveis, de forma que a familiaridade decorre do tempo de convivência da equipe.
- **Facilitação:** o papel do Scrum Master é fundamental para facilitar o desenvolvimento de soluções rápidas e com qualidade. Ele tem experiência com o processo e as tecnologias e coloca todo o seu conhecimento a favor da equipe do projeto.
- **Tamanho da equipe:** a equipe no Scrum normalmente é pequena, com até nove pessoas. Isso possibilita que seus membros possam interagir e se comunicar sempre que necessário para realizar o trabalho. Trabalhar em equipe é indispensável para o sucesso do projeto.
- **Auto-organização:** uma equipe Scrum trabalha para a realização de objetivos específicos. São auto-organizadas e auto-motivadas e possuem as capacidades necessárias para tomar decisões importantes com autoridade.
- **Abertura para pedir/oferecer ajuda:** embora a equipe tenha a capacidade para a tarefa de desenvolvimento de software, determinados problemas são complexos e exigem diferentes expertises. A equipe Scrum usa a familiaridade e a confiança para que cada membro possa demandar ajuda em tarefas complexas, ou oferecer ajuda, se perceber que pode ajudar na dificuldade de um colega.
- **Remoção de impedimentos:** a equipe Scrum organiza reuniões diárias e remove impedimentos que podem prejudicar a realização de suas metas. A base para remoção de impedimentos é a comunicação constante entre os membros da equipe.
- **Melhoria contínua:** a função da reunião *Sprint Retrospective* é garantir um aprimoramento do trabalho, tornando-o mais eficiente, por meio de uma retrospectiva buscando levantar pontos que necessitam aprimoramento e elaborar planos de ação para colocar essas melhoras em prática.
- **Ritmo sustentável:** a equipe Scrum deve manter um ritmo de trabalho sustentável e constante, sem as práticas de horas-extras e aceleração forçada neste ritmo, pois isso gera estresse e afeta negativamente o resultado da equipe.

- **Motivação:** o fato de agir em consonância com os fatores descritos anteriormente, e de trabalhar em uma organização que apoia essa forma de comportamento, contribui para a confiança e para a motivação da equipe.

Considerando-se esses fatores, pode-se concluir que a filosofia de trabalho da metodologia Scrum busca aprimorar o trabalho em equipe durante o processo de desenvolvimento de software [Lindsjörn et al. 2016]. Uma das questões trabalhadas na presente pesquisa é se esses fatores são percebidos pelos membros das equipes Scrum como pontos positivos para o trabalho em equipe e se esta percepção é diferente da percepção apresentada por membros de equipes que usam outras metodologias de desenvolvimento de software. A próxima seção apresenta a análise desta questão.

4. Avaliação das Características de Equipes na Comunidade de Software

Para validar as características que influenciam o trabalho de equipes de desenvolvimento de software identificadas foi realizada uma pesquisa envolvendo uma população de setenta participantes que trabalham em equipes de desenvolvimento de software em organizações situadas na região metropolitana de Goiânia.

A pesquisa foi baseada em um questionário enviado para diversas organizações. De um total de 70 profissionais ligados a essas organizações que responderam o questionário, 37 utilizam o Scrum para desenvolvimento de software, enquanto 33 utilizam outros métodos de desenvolvimento, diferentes do Scrum. O modelo de questionário utilizado, bem como as respectivas repostas, estão descritos em [Moraes 2020].

A primeira parte da pesquisa explora os cinco grandes desafios para o trabalho em equipe sintetizados na Figura 1. A base da pirâmide de desafios é a ausência de confiança. A questão proposta em relação a esta característica avalia se ela realmente ocorre, no ponto de vista dos participantes, e se há diferenças entre equipes que utilizam e que não utilizam o Scrum para o desenvolvimento de software. A forma como os participantes da pesquisa avaliam suas respectivas equipes é mostrada na Figura 2.

As repostas obtidas desta questão mostram que a grande maioria de membros de equipes de desenvolvimento de software confiam, parcial ou totalmente, em seus colegas de equipe. Independente de utilizarem ou não o Scrum, os resultados se mantiveram bem próximos para as três alternativas da questão. Há indícios, portanto, de que essa é uma característica de equipes de desenvolvimento de software, não importando a metodologia usada pela equipe.

A análise das repostas também confirma a importância do estudo da falta de confiança em equipes de desenvolvimento de software, pois mais da metade dos respondentes, tanto no grupo Scrum quanto no grupo de outras metodologias, responderam que não podem confiar totalmente em suas respectivas equipes de desenvolvimento de software.

Em relação ao desafio "medo de conflitos", a questão verifica se ocorrem conflitos constantes entre membros da equipe e, em caso positivo, se esses conflitos são utilizados pela equipe como forma de crescimento e aprendizado. A maneira como os participantes da pesquisa avaliam os conflitos dentro de suas equipes é representada na Figura 3.

As repostas mostram que mais de 70% dos participantes de equipes Scrum utilizam os conflitos que ocorrem nos projetos como forma de crescimento e aprendiza-

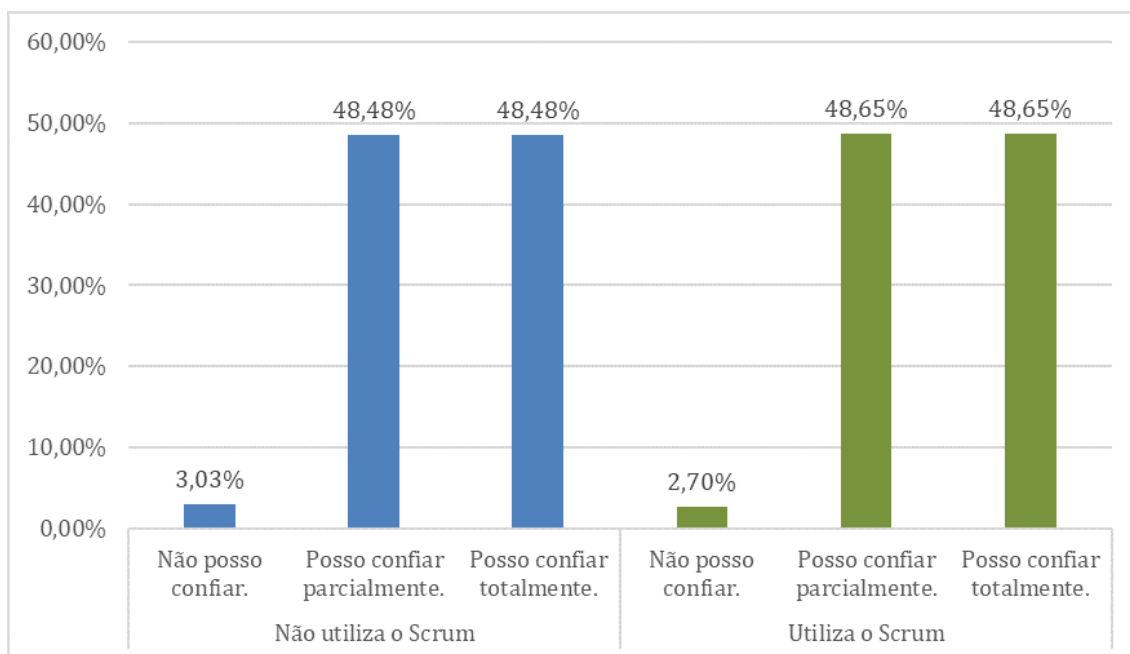


Figura 2. Você sente que pode confiar em seus colegas de equipe?

gem. Isso pode ser atribuído à comunicação promovida no Scrum, por meio das diversas reuniões realizadas no decorrer de um projeto. Elas estimulam o diálogo entre membros da equipe e, apesar das possíveis desavenças, geralmente chega-se a um acordo. Por exemplo, na Retrospectiva da Sprint, durante a auto-inspeção da equipe, podem surgir conflitos. No entanto, a reunião é utilizada para acertar os pontos de vistas divergentes e identificar ações para serem planejadas e melhoradas para a próxima Sprint.

A análise das respostas a esta questão indica, ainda, que a valorização de conflitos construtivos é uma meta a ser atingida pelas equipes de desenvolvimento de software que não usam Scrum, pois mais da metade dos respondentes deste grupo afirma que os conflitos são evitados, ou que nenhum proveito é obtido de sua ocorrência nos projetos.

A questão seguinte busca a percepção dos participantes sobre o comprometimento da equipe com o projeto de software. A Figura 4 mostra as estatísticas de respostas.

É possível observar o mesmo padrão para os dois tipos de equipes. A resposta com maior percentual entre os participantes foi “todos realizam o que foi proposto”, independentemente do método de desenvolvimento utilizado. Com essa análise é possível inferir que a maior parte das pessoas que participam de equipes de desenvolvimento de software está disposta a realizar a atividade combinada, mesmo que não esteja de acordo com essa atividade.

Todavia, o desafio de obter comprometimento das equipes com o planejamento do projeto é uma questão relevante para ambos os grupos, já que a soma das respostas com menor percentual supera a quantidade de repostas que apontam para o total comprometimento da equipe com o que foi planejado para o projeto. Ou seja, a maioria dos participantes das equipes, tanto no Scrum como em outras metodologias, consideram que o grau de comprometimento demonstrado pela equipe não é ideal.

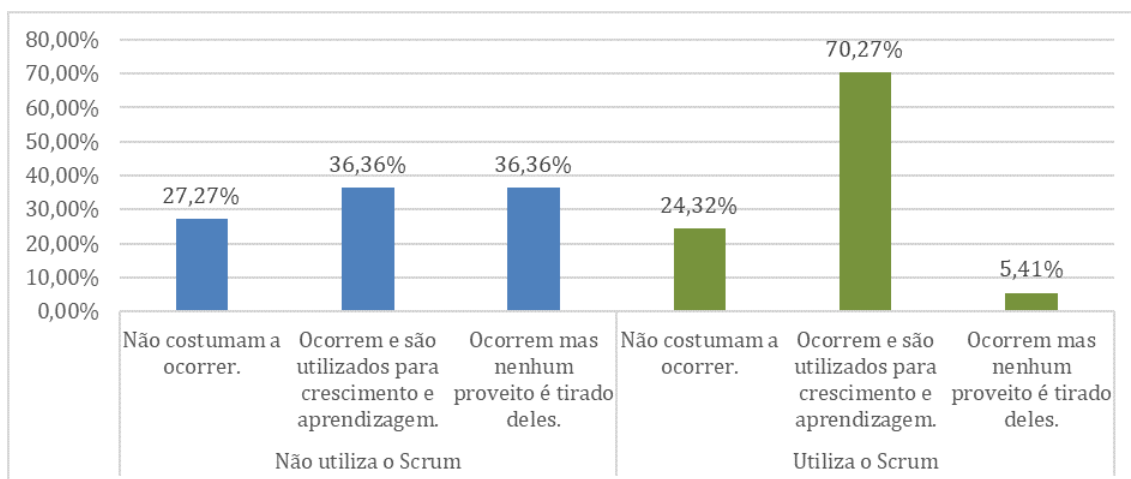


Figura 3. Ocorrem conflitos que geram crescimento e aprendizagem na equipe?

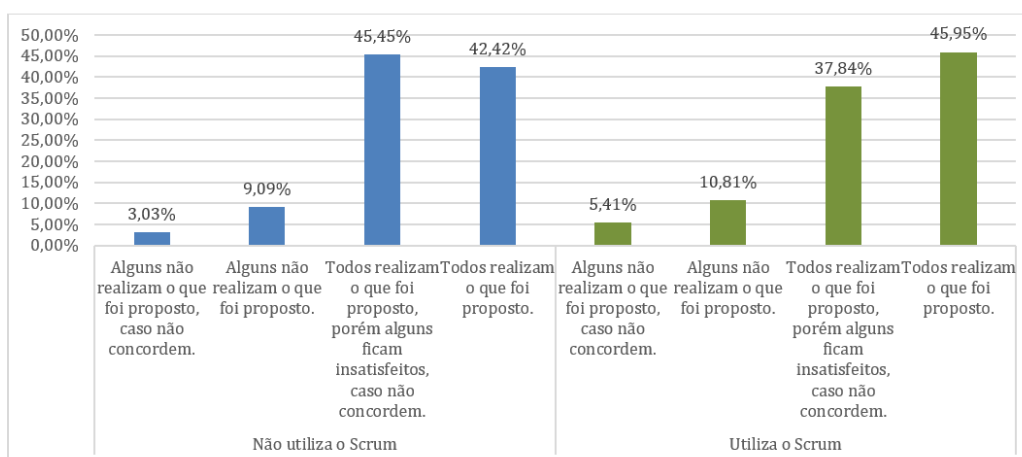


Figura 4. Todos realizam o trabalho planejado, ainda que não concordem?

O objetivo da próxima questão é compreender como o participante avalia a responsabilização em sua equipe. A “Fuga da responsabilidade” preocupa-se com a capacidade das pessoas de assumirem seus erros. Mesmo cometendo erros é necessário ser transparente com a equipe, a fim de procurar formas de contornar a situação e com isso alcançar bons resultados. Os participantes da pesquisa avaliaram a responsabilidade em suas equipes conforme mostra a Figura 5.

As repostas indicam que, para a maioria dos participantes, nem sempre os membros da equipe se responsabilizam por seus atos, incluindo erros e falhas. Esse tipo de conduta pode causar problemas em um projeto. É importante ser sincero com a equipe, e assim resolver o erro sem comprometer o projeto. Quando se olha somente para as equipes que utilizam o Scrum, esse quadro tende a melhorar um pouco, uma vez que quase metade respondeu “Sempre se responsabilizam”.

Geralmente, equipes que não utilizam o Scrum fazem divisões de papéis, como por exemplo, subequipe de requisitos, de desenvolvimento e de testes. Essas entregam o resultado de seus trabalhos para outras de forma independente e autônoma. Em equipes Scrum,

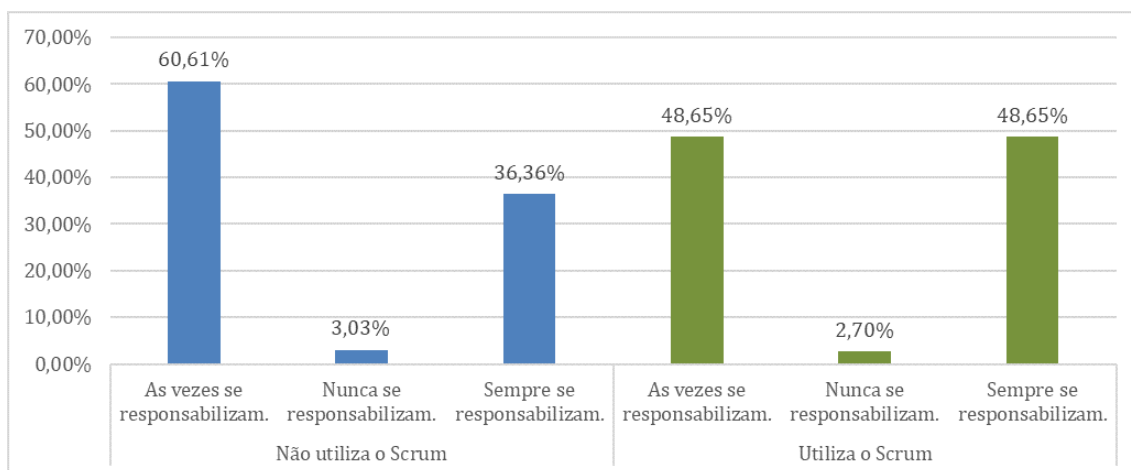


Figura 5. Os integrantes da equipe se responsabilizam por seus atos no projeto?

a mesma equipe deve fazer todo o trabalho para produzir o software durante a *Sprint*, tornando a equipe responsável por todas essas tarefas. Além disso, as reuniões diárias ajudam a responsabilização individual pelas ocorrências no projeto [Sutherland 2020].

Apesar disso, a análise das estatísticas de resposta para essa questão confirma que o desafio de obter responsabilização individual sobre as ações realizadas no projeto está presente em ambos os grupos de respondentes.

Ainda em relação à característica de fuga da responsabilidade, a questão seguinte busca saber se existe um senso de responsabilidade coletivo na equipe. Este senso de responsabilidade pode ser constatado em uma equipe que avalia suas ações como um todo e não individualmente. A questão investiga se os membros da equipe apenas fazem sua parte, ou se também se preocupam com o sucesso dos demais. Caso algo não saia como planejado, a equipe toda se responsabiliza e procura soluções para resolver os problemas, sem apontar culpados. A visão dos participantes sobre o senso de responsabilidade nas equipes é mostrada na Figura 6.

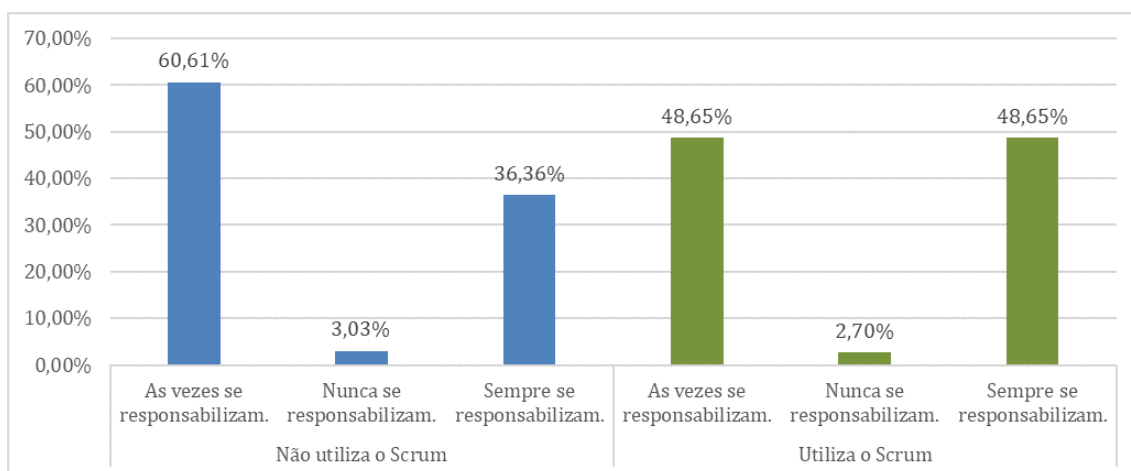


Figura 6. A equipe costuma se responsabilizar por seus atos?

As respostas indicam que existe um senso de responsabilidade maior para equi-

pes que utilizam o Scrum. Provavelmente, isso ocorre devido ao espírito de trabalho em equipe que é promovido no Scrum. Na filosofia Scrum, as tarefas devem ser de responsabilidade de todos, e existe uma única equipe, sem subdivisões.

Mesmo tendo uma avaliação melhor em relação à fuga de responsabilidade coletiva pelo projeto, este desafio deve ser considerado nas equipes Scrum, uma vez que mais da metade dos respondentes neste grupo não consideram que a equipe sempre se responsabiliza pela condução do projeto. No grupo de outras metodologias, a situação é mais grave, pois quase 65% dos participantes indicam que a equipe apresenta algum grau de fuga de responsabilidade coletiva.

A questão seguinte avalia a opinião dos participantes da pesquisa sobre o foco da equipe nos resultados finais de um projeto. No topo da pirâmide de desafios aparece o item “Falta de atenção aos resultados”. Para assegurar tal atenção aos resultados, é fundamental que as metas do projeto sejam autoexplicativas e fáceis de lembrar. Além disso, é necessário que as tarefas realizadas diariamente tenham uma ligação óbvia com o cumprimento das metas. A avaliação dos participantes quanto ao foco nos resultados finais é representada na Figura 7.

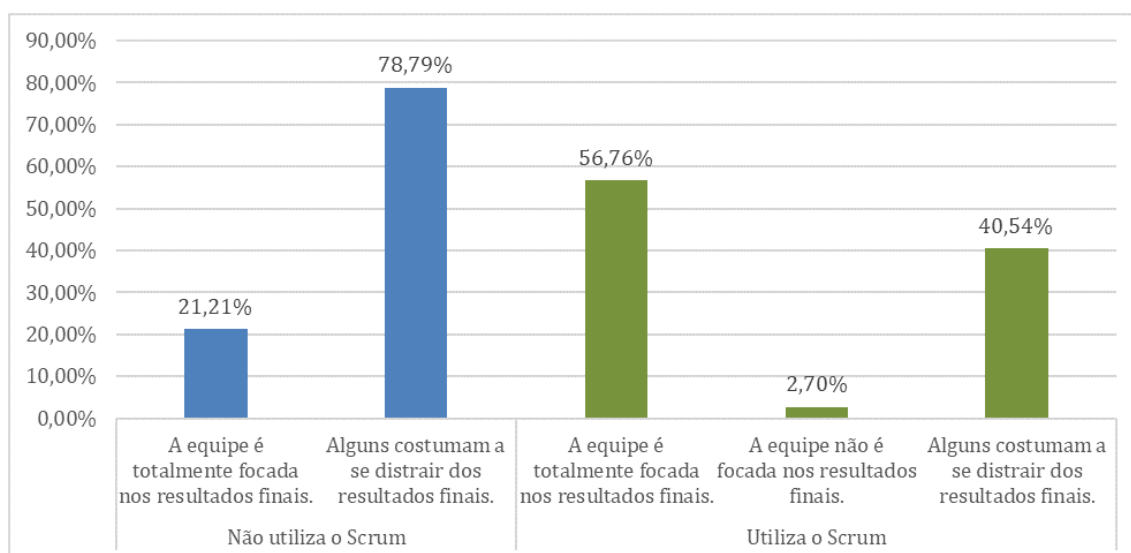


Figura 7. A equipe se distrai dos resultados finais no decorrer do projeto?

As respostas obtidas dos participantes da pesquisa indicam que as equipes que utilizam o Scrum são mais focadas nos resultados finais de um projeto. Os diversos eventos obrigatórios no Scrum, com destaque para a Reunião de Planejamento da Sprint e a Reunião Diária, ajudam a equipe a focar nos resultados finais. Em particular, na reunião diária discute-se o que será realizado nas próximas 24 horas, criando uma relação clara entre alcançar a meta final do projeto e as atividades realizadas diariamente. Apesar disso, mais de 40% dos participantes do grupo Scrum entendem que a equipe não foca totalmente nos resultados finais do projeto.

No grupo de respondentes que utilizam outras metodologias de desenvolvimento de software, as estatísticas de resposta desta questão são muito expressivas, já que quase 80% das respostas indicam que alguns membros da equipe apresentam falta de atenção aos resultados finais do projeto. Logo, é um desafio que precisa ser enfrentado para

melhorar a produtividade das equipes de software.

A questão seguinte trata da motivação dos participantes de equipes de desenvolvimento de software, avaliando se o participante se sente motivado no seu trabalho. A Figura 8 apresenta as estatísticas de resposta para essa questão.

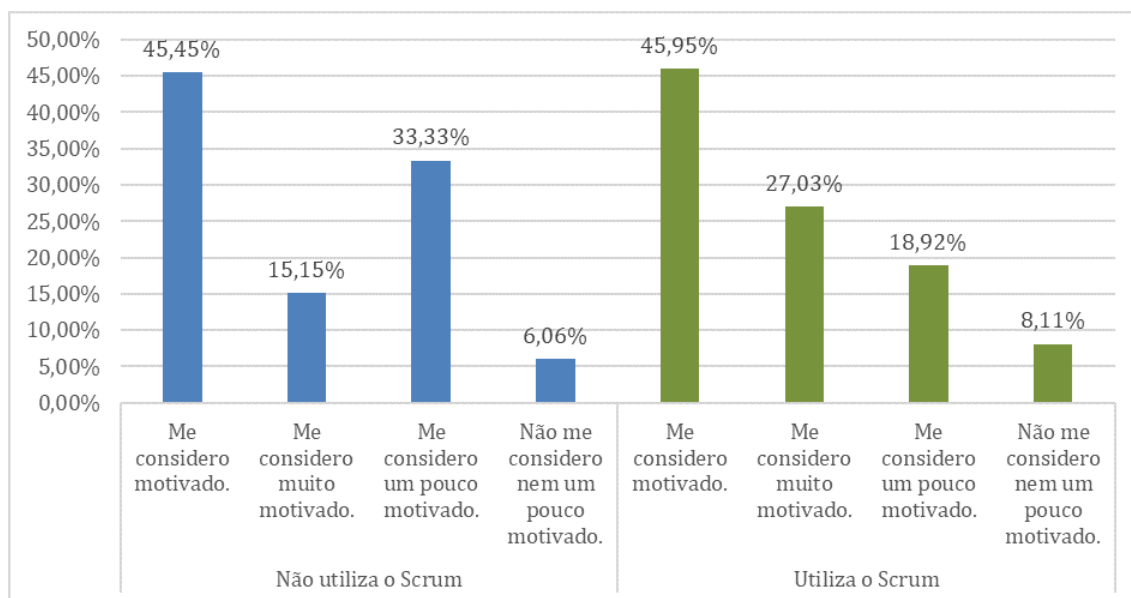


Figura 8. Você se considera motivado em seu trabalho na equipe de software?

As respostas mostram que participantes de equipes Scrum se sentem relativamente mais motivados do que os membros de equipes que utilizam outras metodologias de desenvolvimento. A alternativa “Me considero muito motivado” recebeu quase 12% a mais de respostas entre os participantes de equipes Scrum, enquanto a alternativa “Me considero um pouco motivado” recebeu cerca de 15% a mais de respostas entre os respondentes que não utilizam Scrum. As outras duas alternativas obtiveram uma quantidade de respostas semelhantes entre os dois grupos de respondentes. A maior motivação entre os participantes de equipes Scrum pode ser consequência dos resultados obtidos das respostas anteriores, em que membros de equipes Scrum tiveram melhores percepções de suas respectivas equipes.

Vale destacar que um número considerável de participantes dos dois grupos se considerou “um pouco motivado” ou “nem um pouco motivado” no trabalho. Isso confirma a importância de pesquisas voltadas para a compreensão de fatores humanos em equipes de desenvolvimento de software, já que a motivação, como discute a Seção 2.2, é fator determinante para os resultados dessas equipes.

5. Resultados da Pesquisa

A Tabela 1 sintetiza os resultados obtidos da análise das respostas da pesquisa de campo realizada. Das seis características desejáveis em equipes de desenvolvimento de software avaliadas na pesquisa, duas apresentaram resultados semelhantes, independentemente do método de desenvolvimento utilizado pelas equipes, enquanto quatro características apresentaram resultados superiores em equipes que utilizam o Scrum, de acordo com a opinião dos participantes da pesquisa.

Característica de Equipes de Software	Resultado: Scrum X Outras Metodologias
Confiança nos Colegas de Equipe	Semelhante
Valorização de Conflitos Construtivos	Superior
Comprometimento com os Planos	Semelhante
Senso de Responsabilidade (Individual e Coletivo)	Superior
Foco nos Resultados do Projeto	Superior
Motivação do Participante	Superior

Tabela 1. Comparação de Características em Equipes de Software

Os pontos identificados como superiores em equipes de desenvolvimento que adotam a metodologia Scrum foram:

- Os conflitos são melhor administrados em equipes Scrum, e são utilizados para fomentar o crescimento e o aprendizado da equipe.
- Os membros de equipes Scrum se responsabilizam mais por seus atos no projeto, ainda que sejam erros.
- Existe um maior senso de responsabilidade individual e coletiva por parte das equipes Scrum em relação às ocorrências do projeto.
- Membros de equipes Scrum são mais focados nos resultados finais e sofrem menos distrações que os participantes de outras metodologias.
- Os participantes de equipes Scrum se sentem mais motivados em seu trabalho na equipe de desenvolvimento de software.

Já as características "confiança" e "comprometimento" apresentaram resultados similares nas respostas dos participantes do grupo que participa de equipe Scrum e do grupo que adota outras metodologias de desenvolvimento. Há indícios, portanto, de que essas características são ortogonais ao método de desenvolvimento utilizado pela equipe.

Vale ressaltar que os critérios utilizados nas questões para qualificar o trabalho em equipe foram semelhantes para os dois tipos de equipes de desenvolvimento de software. Os valores são dados em porcentagem devido às diferentes quantidades de participantes que utilizavam (37) e que não utilizavam (33) o Scrum. Esta diferença, no entanto, não representa ameaça à validade das conclusões do presente estudo, já que a dimensão dos dois grupos é bastante similar.

Nessa pesquisa não foi avaliado se os participantes que afirmaram trabalhar em equipes adeptas da metodologia Scrum de fato seguem todos os princípios e processos desta metodologia. Dessa forma, as conclusões da pesquisa precisam ser confirmadas por outros trabalhos semelhantes e, idealmente, aplicando-se a pesquisa em equipes que possuem evidências objetivas de aplicação de todos os princípios propostos na metodologia. Considerando-se, por outro lado, que o propósito da presente pesquisa é investigar o ponto de vista dos praticantes em relação às características de suas respectivas equipes, as conclusões são válidas, já que o próprio participante avaliou cada característica para responder ao questionário de acordo com a sua percepção em relação à sua equipe.

Outro ponto que merece destaque é que, a fim de evitar vieses que pudessem comprometer a validade da pesquisa, o questionário manteve o anonimato dos respondentes.

Com isso, não é possível verificar se os participantes que responderam a pesquisa em cada equipe das organizações selecionadas para participar desta pesquisa possuem uma boa representatividade em relação à equipe como um todo. A anonimização também torna impossível distinguir a qual equipe da organização cada participante pertence.

Por fim, cabe ressaltar que a presente pesquisa apresenta indícios que embasam as conclusões e resultados apresentados. Todavia, não se pode afirmar, com certeza definitiva, que a superioridade identificada nas equipes de desenvolvimento adeptas do Scrum se deve, exclusivamente, ao fato de adotarem as práticas desta metodologia, já que outros fatores poderiam influenciar esse melhor desempenho das equipes Scrum. Por exemplo, fatores organizacionais poderiam ser a causa da superioridade das equipes Scrum, e não as práticas da metodologia de desenvolvimento de software propriamente dita.

6. Conclusões

Este trabalho apresenta um estudo sobre fatores humanos relacionados à produtividade de equipes de desenvolvimento de software. O estudo identifica características que influenciam positiva e negativamente o trabalho desse tipo de equipe e discute propostas para lidar com cada tipo de característica em equipes de desenvolvimento de software.

Para confirmar a validade de características identificadas nos estudos bibliográficos realizados, o artigo apresenta os resultados de uma pesquisa de campo realizada com a participação de setenta participantes de equipes de desenvolvimento de software de diferentes organizações.

Dois grupos de participantes foram identificados: pessoas que trabalham em equipe de desenvolvimento com base na metodologia de desenvolvimento ágil Scrum e pessoas que trabalham com outras metodologias de desenvolvimento. Com isso foi possível comparar as respostas dadas pelos participantes dos dois grupos, permitindo concluir que as pessoas que trabalham com Scrum tendem a perceber melhores qualidades e características desejáveis em suas respectivas equipes de desenvolvimento de software.

Uma possível explicação para o melhor desempenho da metodologia ágil Scrum em relação à avaliação de características de produtividade é a existência, nesta metodologia, de práticas diretamente relacionadas a fatores humanos e de trabalho cooperativo, tais como facilitação do trabalho em equipe, auto-organização e motivação, remoção de impedimentos, melhoria contínua, e manutenção de ritmo sustentável de trabalho. Assim, a presente pesquisa apresenta indícios de que os componentes e práticas da metodologia Scrum podem contribuir para a produtividade e para a qualidade do trabalho em equipes de desenvolvimento de software.

Como trabalhos futuros pode-se destacar a análise de outras características relacionadas a fatores humanos em equipes de desenvolvimento de software que não foram contempladas no presente trabalho, como por exemplo, fatores motivacionais. Além disso, a aplicação das mesmas questões propostas neste trabalho em equipes de desenvolvimento de software de outras regiões é uma extensão útil para confirmar a generalizabilidade das conclusões aqui apresentadas.

Referências

Batra, D., Xia, W., and Zhang, M. (2017). Collaboration in agile software development: Concept and dimensions. *Communic. of the Association for Information Systems*, 41.

- Beecham, S., Baddoo, N., Hall, T., Robinson, H., and Sharp, H. (2008). Motivation in software engineering: A systematic literature review. *Information and Software Technology*, 50(9-10):860–878.
- Deci, E. L. and M., R. R. (2000). Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary Educational Psychology*, 25(1):54–67.
- DeMarco, T. and Lister, T. (2013). *Peopleware: Productive Projects and Teams*. Addison-Wesley Professional, 3 edition.
- Emelyanova, T., Ilchenko, E., Varlamova, Z., and Paklina, L. (2019). Teamwork management in the field of software development. *Proc. of the Ecological-Socio-Economic Systems: Models of Competition and Cooperation (ESES 2019)*, pages 490–495.
- Fuks, H., Raposo, A. B., Gerosa, M. A., and Lucena, C. J. P. (1991). Applying the 3c model to groupware development. *Int. Journal of Cooperative Information Systems*.
- Gorla, N. and Lam, Y. W. (2004). Who should work with whom?: building effective software project teams. *Communications of the ACM*, 47(6).
- Johnson, D. W. and Johnson, R. T. (1995). *Conflict, Cooperation, and Justice*, chapter Social Interdependence - Cooperative Learning in Education. JosseyBass Publishers.
- Lencioni, P. (2015). *Os 5 Desafios das Equipes*. Rio de Janeiro: Sextante.
- Lindsjørn, Y., I.K.Sjøberg, Dingsøy, T., R.Bergersen, G., and Dybå, T. (2016). Teamwork quality and project success in software development: A survey of agile development teams. *Journal of Systems and Software*, 122:274–286.
- Locke, E. A. and Schattke, K. (2019). Intrinsic and extrinsic motivation: Time for expansion and clarification. *Motivation Science*, 5(4):277–290.
- Luca, J. and Tarricone, P. (2001). Does emotional intelligence affect successful teamwork? In *Proc. of the 18th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education*. University of Melbourne.
- Moraes, D. A. (2020). Estudo sobre equipes de desenvolvimento de software. Trabalho de conclusão de curso, Escola de Ciências Exatas e de Computação. Pontifícia Universidade Católica de Goiás (PUCGO).
- O’Connell, T. and Molloy, O. (2019). Trust: Promoting interdependence, knowledge sharing & collaboration in scrum teams. In *Proceedings of the 2019 International Conference on Software Engineering Research & Practice*, pages 10–18. CSREA Press.
- Oliveira, E. C. C. d. (2017). *Fatores de influência na produtividade de desenvolvedores de organizações de software*. PhD thesis, Universidade Federal do Amazonas.
- Ortiz, F. C. and Farina, E. (2015). A motivação dos profissionais que desenvolvem projetos utilizando as tecnologias da informação: um estudo empírico. *Revista Eletrônica Acadêmica Drummond*, 6(7):37–55.
- Scarnati, J. T. (2001). On becoming a team player. *Team Performance Management: An International Journal*, 7(1/2):5–10.
- Sutherland, J. J. (2020). *SCRUM: guia prático*. Rio de Janeiro: Sextante.
- Whitehead, J. (2007). Collaboration in software engineering: A roadmap. *Future of Software Engineering (FOSE '07)*, pages 214–225.