

Modelagem e Simulação de uma Equipe de Desenvolvimento Scrum

Lucas Silveira Kupssinskü¹, Gustavo Souza¹, José Ernesto da Silva Barbosa¹

¹PPGCA – Programa de Pós Graduação em Computação Aplicada
Universidade do Vale do Rio dos Sinos (UNISINOS) – São Leopoldo, RS – Brazil
lucas.kup@gmail.com, gtvosouza@hotmail.com, ernestosbarbosa@outlook.com

Abstract. *This paper aims to apply modeling and simulation techniques to the team scaling problem in agile software development teams. To achieve this, a real development team was studied using a database with historical data raging two years. Using this data, a simulation in Arena software was created, making the number of team members a variable that can be adjusted and that indicators of the team goal can be observed. The results suggest that it is possible to apply simulation as a tool to aid managers in allocation and definition of development team.*

Resumo. *Este artigo propõe aplicar técnicas de modelagem e simulação ao problema de dimensionamento de equipes de desenvolvimento ágil. Dessa forma, foi estudado uma equipe de desenvolvimento real com base em dados históricos de dois anos. Utilizando esses dados, foi gerada uma simulação no software Arena, possibilitando que a quantidade de integrantes do time possa ser ajustada na simulação e que indicadores dos objetivos do time possam ser observados. Os resultados atingidos na pesquisa, sugerem que é possível aplicar a simulação como ferramenta de apoio aos gestores que possuem a tarefa de alocação e definição de times de desenvolvimento.*

1. Introdução

Dimensionar uma equipe de desenvolvimento de sistemas não é uma questão trivial. Uma equipe ágil mal dimensionada pode gerar ociosidade dos membros do time, aumentar a quantidade de retrabalho ou mesmo não cumprir com as expectativas. Muitas vezes a tarefa de determinação do time, cabe a profissionais experientes que, tomam suas decisões de forma intuitiva, carecendo de um método definido ou de ferramentas para auxiliar nesta atividade.

Diante desse problema, o presente trabalho propõe uma forma de analisar o desempenho de equipes de desenvolvimento ágil que utilizam Scrum, possibilitando simulação de cenários hipotéticos para chegar a um dimensionamento de equipe mais próximo ao ideal. A metodologia escolhida para abordar essa questão é a modelagem do processo de desenvolvimento e simulação computacional com base em dados históricos. Dessa forma, o processo de desenvolvimento da empresa foi observado, e os gargalos nos quais as atividades podem gerar filas de espera foram mapeados para o software Arena. Parâmetros que foram considerados relevantes para o processo de desenvolvimento, tais como Tamanho das Histórias, Número de Manutenções criadas por dia e número de urgências atendidas pela equipe, foram coletados em uma base histórica de dois anos, fornecida pela empresa alvo do estudo. Esses dados foram aproximados em distribuições

probabilísticas que foram consideradas na simulação do modelo.

O modelo foi validado contra os resultados dos últimos dois *Sprints* desenvolvidos pela empresa, sugerindo que a simulação pode gerar resultados representativos para avaliação do dimensionamento de uma equipe de desenvolvimento ágil, considerando que existam dados históricos para a construção do modelo.

O artigo está organizado na seguinte estrutura, o capítulo 2 apresenta uma breve revisão bibliográfica, juntamente com o Problema, Justificativa, Hipótese e Objetivo da Pesquisa, o capítulo 3 apresenta a metodologia utilizada bem como os dados levantados e a forma de trabalho da empresa alvo do estudo, o capítulo 4 discorre sobre os testes e resultados atingidos e o capítulo 5 apresenta considerações finais e ideias de novas pesquisas nessa mesma área.

2. Desenvolvimento

[Ahimbisibwe et al. 2015] destaca que um dos fatores críticos para o sucesso de um projeto é o tamanho da equipe de desenvolvimento. [Pendharkar and Rodger 2009] complementa essa informação indicando também que definir o tamanho de um time continua sendo um desafio para os gestores, que muitas vezes tem que apelar para abordagens intuitivas devido a carência de métodos ou ferramentas para apoio nessa decisão, sendo que em alguns casos a metodologia é escolhida baseado no tamanho do time [Keshta and Morgan 2017].

A pesquisa por métodos para melhor estimar o tamanho de times de desenvolvimento já se justifica pelas afirmações de PENDHARKAR et al (2009), porém sua importância aumenta ainda mais se levarmos em consideração que, tipicamente, mais de 50% do custo da software acaba se concentrando nas atividades de manutenção [Glass 2001] e já existem resultados empíricos que sugerem que o tamanho da equipe de desenvolvimento é correlacionada positivamente com o esforço de manutenção [Aziz et al. 2009].

Dessa forma, partimos da hipótese que, dado uma base histórica sobre a performance de uma equipe, é possível modelar os resultados de um time de desenvolvimento ágil via simulação possibilitando que diferentes composições de times sejam avaliadas, de modo que o gestor possa saber quais seriam as consequências de uma eventual realocação de recurso.

Definido o problema de pesquisa, o presente artigo tem como objetivo apresentar os resultados de uma simulação de um time de desenvolvimento Scrum com foco no dimensionamento de equipes de desenvolvimento ágil utilizando dados históricos no software de simulação Arena.

Essa abordagem tem a vantagem de possibilitar o uso do tamanho do time como uma das variáveis do processo de simulação, possibilitando que o indivíduo que está executando a simulação observe o comportamento e os resultados do time variando o número de integrantes. Autores como [Powell et al. 2004] salientam em suas pesquisas que não tinham conhecimento de estudos que permitissem observar o comportamento do

sistema considerando o tamanho do time como variável.

É necessário salientar que diversos fatores presentes em um ambiente real de desenvolvimento não são escopo desse trabalho e foram desconsideradas para simplificação do modelo. Alguns fatores que merecem destaque são: realocação de recursos durante o *Sprint*, comunicação interna da equipe de desenvolvimento, diferença do nível de maturidade entre os integrantes da equipe. A diferenciação de nível de maturidade foi desconsiderada pois, conforme aprofundado na próxima sessão, os dados de desenvolvimento levantados durante a janela de tempo de dois anos, englobam o trabalho de pessoas de diferentes níveis de maturidade, portanto tomamos como premissa que os dados já expressam o nível de maturidade médio dos profissionais que trabalham na empresa estudada

3. Método proposto

Para fins de caracterização, a empresa que cedeu os dados para o estudo (denominada a partir de agora como Empresa X) desenvolve um sistema de gestão utilizado por empresas de diversos segmentos em todo o Brasil, contando com cerca de 5000 clientes e 500 funcionários.

Com intuito de definir o modelo para desenvolvimento do simulador, o processo de desenvolvimento da Empresa X foi observado, percebendo-se que a metodologia de desenvolvimento Scrum é aplicada com algumas particularidades que serão destacadas a seguir.

O processo de desenvolvimento da Empresa X é baseado em Scrum, uma metodologia de desenvolvimento representativa na categoria de metodologias ágeis. O processo Scrum inicia a partir da visão do Product Owner, considerando o produto que ele deseja desenvolver e de uma base de requisitos devidamente priorizados, o produto é desenvolvido utilizando um processo iterativo e incremental a partir do trabalho do time em períodos pré-fixados de tempo, os *Sprints* [Sharma and Hasteer 2016].

A figura 1 ilustra o processo padrão do Scrum. A etapa desse processo que foi expandida e modelada é o *Sprint*. Analisando a colocação do *Sprint* dentro do processo Scrum, é possível perceber que ele possui como entrada o *Sprint Backlog* e como saída o “Produto ou Funcionalidade Concluída”, esses pontos necessitam ser definidos no simulador.

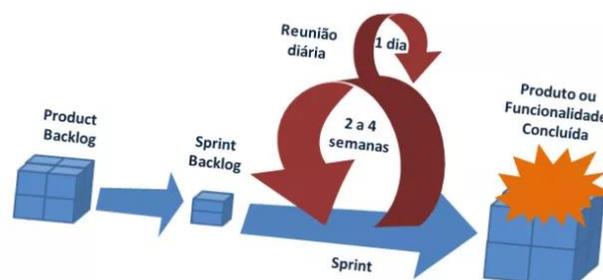


Figura 1. Processo Scrum

Contudo, apesar da figura 1 representar bem o fluxo de trabalho de um time Scrum, a Empresa X fez algumas adaptações no processo padronizado com intuito de tornar ele mais aderente a sua realidade, a principal modificação é a possibilidade de

outras equipes acessarem o time Scrum para levar problemas urgentes ou manutenções, conforme ilustrado na figura 2 a seguir.

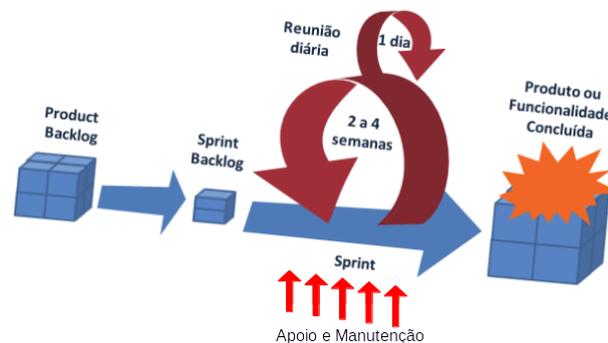


Figura 2. Processo Scrum da Empresa X

A figura 2 difere da figura 1 apenas nas setas destacadas em vermelho ilustrando “Apoio e Manutenção”, que são conceitos presentes na empresa modelada, sendo apoio definido como interrupções realizadas por equipes externas ao time Scrum e Manutenção são os chamados de manutenção corretiva urgentes que podem surgir durante o período de um *Sprint*.

O conceito de apoio oferecido pela empresa é bastante genérico, porém em contato com o time foi levantado que, na prática, existem dois momentos principais nos quais esses apoios ocorrem, na etapa de análise de sistemas para resolver dúvidas não documentadas sobre o funcionamento do software e dúvidas sobre programação, levantadas normalmente pela equipe de atendimento ao cliente que acaba necessitando interromper o time Scrum. Dessa forma o apoio é caracterizado por ser uma demanda que surge durante o dia a dia e fica de fora do planejamento da equipe de desenvolvimento, esse tipo de atividade entra na fila de desenvolvimento e consome tempo da equipe da mesma forma que as demais atividades, adicionando mais um fator de imprevisibilidade ao processo.

De posse dessas informações referentes ao processo da empresa, foram acessadas as bases históricas de informação da empresa, de modo que possamos estudar o comportamento estatístico das atividades de manutenção, apoio e tamanho de atividades. Essas informações então serão utilizadas como entrada para uma simulação de filas, na qual se espera verificar se a equipe está bem dimensionada ou se existe algum possível gargalo que pode ser evitado com um novo arranjo da equipe.

4. Testes e Resultados

A sessão de resultados está subdividida em Dados do Processo, Modelo e Simulações, nos quais são apresentados, respectivamente, as informações coletadas junto a Empresa X e as distribuições probabilísticas aproximadas, o modelo implementado no software Arena e as simulações realizadas no modelo.

4.1 Dados do Processo

Para construir o modelo, foram levantados os seguintes dados referentes ao processo: Número de Apoios por Dia, Número de Manutenções por dia e Tamanho das Atividades do *Sprint* e a razão de apoios que viram manutenção. A seguir será mostrada a análise desses dados feita com apoio do software Input Analyzer que é distribuído em conjunto

ao Arena. A seguir encontram-se tabulados os dados utilizados e os resultados das distribuições aproximadas.

Tabela 1. Resumo dos Dados Coletados

	Apoios por Dia	Manutenções por Dia	Tamanho das Atividades (horas)
Observações	511	331	95
Mínimo	1	1	1
Máximo	26	7	125
Média	5.31	1.64	21.1
Desvio padrão	2.95	0.908	21.7
Distribuição	Normal	Beta	Weibull
Expressão	NORM(5.31, 2.95)	0.5 + 7 * BETA(1.16, 5.96)	0.999 + WEIB(19.4, 0.923)
Erro Quadrático	0.002864	0.001123	0.002636

A Tabela 1 resume os dados coletados junto a Empresa X, foram 937 atividades de desenvolvimento analisadas (cada uma delas consta como uma observação na primeira linha da tabela), as informações de Mínimo, Máximo, Média e Desvio padrão são computados em horas gastas em cada uma das atividades, a distribuição se refere a distribuição de probabilidades que mais se ajusta ao dado, o Erro Quadrático se refere ao erro encontrado entre a distribuição probabilística e o dado cole

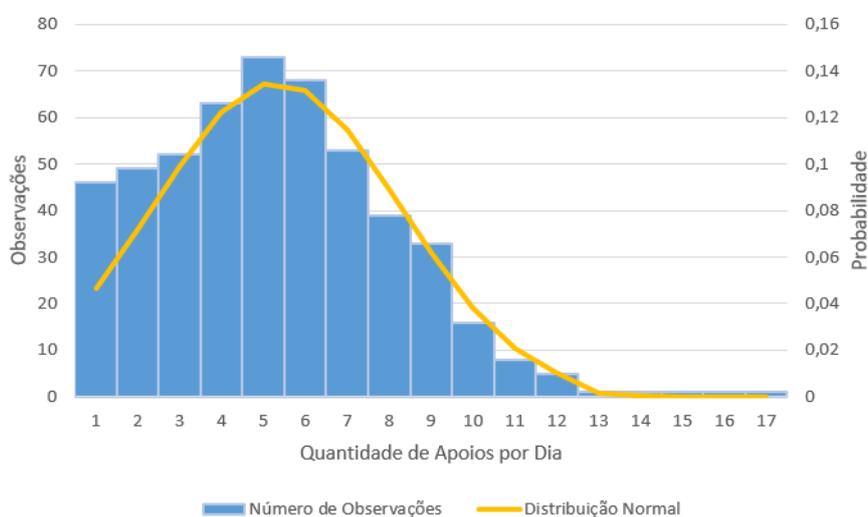


Figura 3. Distribuição de apoios por dia

O dado de Apoio por Dia, refere-se a quantidade de vezes que o time é interrompido por dia por alguma atividade externa qualquer, essa interrupção demanda que o técnico pare de trabalhar nas atividades previstas e de um tratamento adequado para essa atividade. A aproximação realizada sobre as informações de apoios por dia pode ser vista na figura 3 na forma gráfica. De modo que a distribuição de probabilidade para a quantidade de apoios que será aberta em um dia segue um comportamento aproximado por uma exponencial.

Outra informação importante para a simulação do processo é a quantidade de manutenções que entram na fila da equipe por dia. Manutenções são caracterizadas como chamados de correção do sistema e recebem prioridade sobre as demais atividades, dessa forma a equipe deve ser capaz de atender além das histórias do *Sprint* os chamados de manutenção que são criados conforme distribuição plotada abaixo.

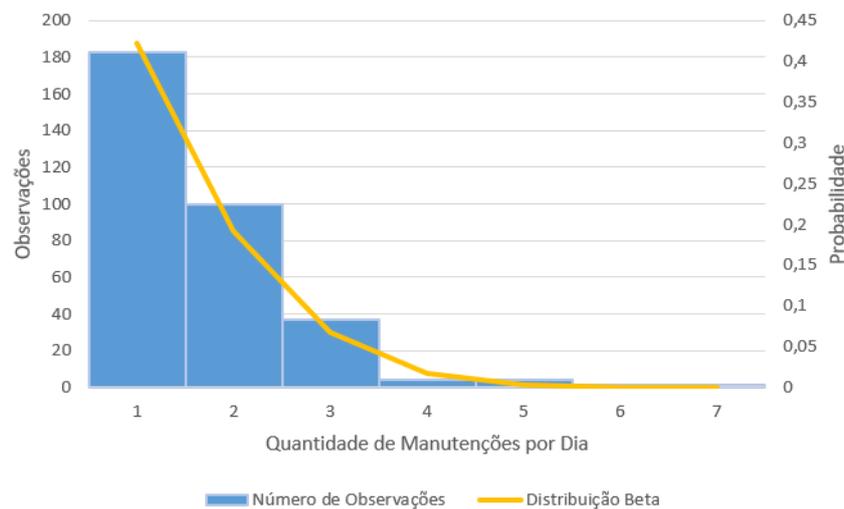


Figura 4. Distribuição de manutenções por dia

As atividades do backlog que entram nas *Sprints* tem tamanhos variados, a medição de tamanho em desenvolvimento de software pode ser realizada de diversas maneiras, porém, nesse artigo, será considerada a mesma medida empregada pela Empresa X, o número de horas previsto para execução da atividade. A figura 4 resume de forma gráfica o tamanho das atividades do backlog que entram nos *Sprints*, esses dados contam com o tamanho planejado em horas para essas atividades. Esse comportamento foi aproximado via distribuição de Weibull.

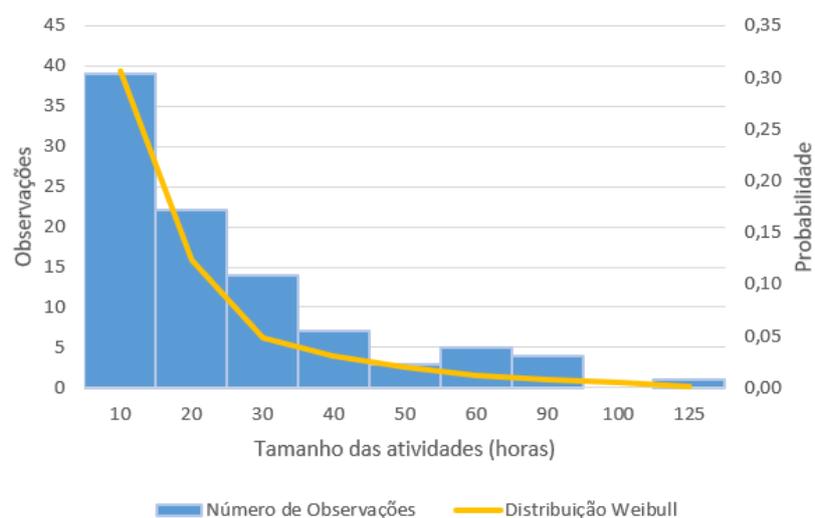


Figura 5. Distribuição do tamanho das atividades

Outra informação importante para a simulação do processo usando filas é o tamanho (em horas) das atividades da *Sprint*, observando a figura 5 podemos ver a

distribuição de probabilidades e o histograma dos dados coletados sobre o tamanho das atividades plotados nos mesmos eixos. O desvio padrão desse dado é maior do que os demais, caracterizando que existe grande diversidade de tamanho nas atividades da *Sprint*.

4.2 Modelo

De posse dos dados e do processo da empresa, utilizou-se o software Arena para realizar a simulação do processo. O Arena é um software largamente utilizado em Pesquisa Operacional para otimização de processos baseados em filas, é um software pago, porém fornece versão gratuita para fins acadêmicos.

Conforme a figura 6 ilustra, o modelo do Arena consta basicamente com três pontos de entrada para o processo do *Sprint*, um deles é o próprio “*Sprint*”, que representa as atividades planejadas da equipe, o segundo é representado pelo bloco “*Extra Sprint*” que representam os chamados de manutenção corretiva e o terceiro é a solicitação de apoio. O segundo e terceiro blocos de entrada obedecem às distribuições estatísticas apresentadas na sessão anterior.

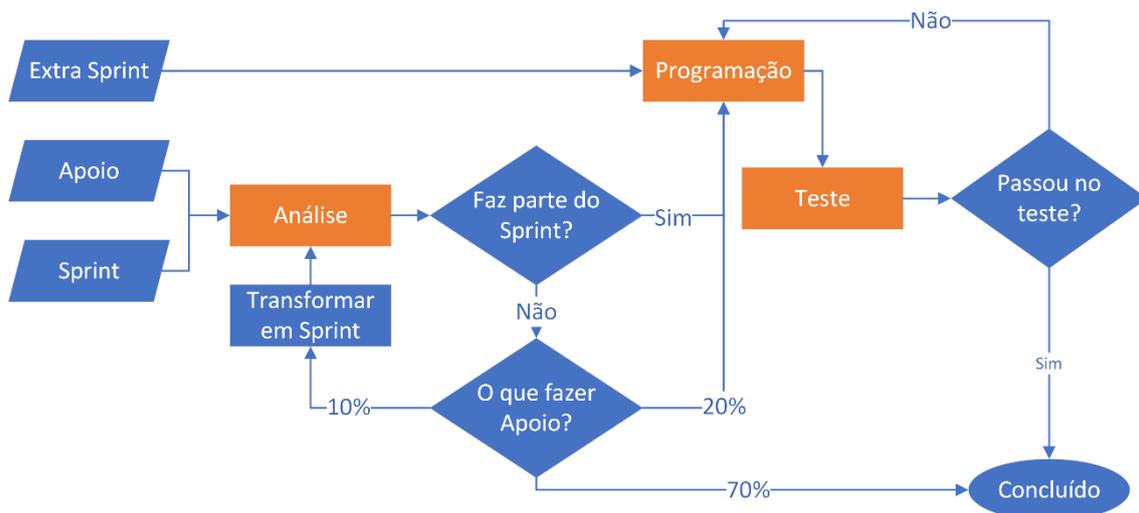


Figura 6. Modelo desenvolvido no Arena

Após os blocos de entrada, ocorre a definição da prioridade das atividades, conforme comentado em sessões anteriores, apoios e manutenções devem ter prioridade sobre as atividades do *Sprint*, para modelar esse comportamento no Arena, foi realizada uma atribuição de prioridade para cada uma das atividades e as filas que se formam no Analista de Sistemas, na Programação e no Teste são consideradas filas prioritárias, nas quais as atividades com maior prioridade são selecionadas primeiro.

Outra informação importante considerada no modelo, refere-se aos resultados dos testes, a atividade de testes acaba passando para concluído cerca de 90% das atividades, as demais voltam para a fila do programador para retrabalho, esse fato é modelado considerando o condicional após a atividade de teste com a razão de aprovação de 1 para 10.

4.3 Resultados

A modelagem realizada, permite estudar o comportamento das filas de atividades para análise, programação e teste, dessa forma foi possível estudar esses gargalos durante

o decorrer do *Sprint*. Filas ociosas ou com grande tempo de espera indicam uma possibilidade de melhoria no dimensionamento da equipe.

Foram realizadas simulações de 100 *Sprints* com a equipe contando com um analista de sistemas, três programadores e um testador, optou-se por essa composição de equipe pois essa é a composição da equipe de desenvolvimento da Empresa X, cujos dados foram considerados nas sessões anteriores. A partir dessas simulações, foram monitoradas as formações de fila de atividades, nas figuras do analista de sistemas, programador e testador. O comportamento dessas filas está tabulado na a seguir.

Tabela 2. Resumo das filas de atividade após 100 simulações

	Tempo de Espera Médio (horas)	Tempo de Espera Máximo (horas)	Tamanho Médio (atividades)	Tamanho Máximo (atividades)
Analista	4,9086	19,5027	6,0436	22
Programador	1,8458	15,6332	4,4534	18
Testador	2,0537	15,0396	4,1958	21

As informações da tabela 2, mostram que os tempos médios de espera são consideravelmente diferentes para as filas de atividade de analista, programador e testador. Como existem mais pessoas trabalhando no papel de programador, o tempo médio de espera na fila de atividades do programador é menor, ficando em 1,84 dias aproximadamente, sendo que o Analista de Sistemas, por desempenhar diversas atividades, acaba demorando em média 4,9 dias para começar a trabalhar em sua atividade. As demais informações tabuladas mostram uma certa semelhança de comportamento entre os três papéis, de modo que o tempo de espera máximo na fila também é o maior para o Analista, ficando em 19,5 dias, isso significa que no pior caso o Analista só conseguiu começar a avaliação de uma atividade no último dia do *Sprint*, pois os *Sprints* possuem 20 dias.

Outro fator interessante da simulação é o seu comportamento dinâmico, como as atividades são essencialmente sequenciais, percebe-se que no começo do *Sprint* o Analista fica bastante sobrecarregado, gerando as maiores filas nos primeiros dias de atividade. Conforme o *Sprint* vai passando, o gargalo também segue o fluxo, ocasionando uma fila grande em programação que, no final do *Sprint*, acaba gerando uma grande fila no testador, que teve momentos de ociosidade nos primeiros dias. Esse tipo de comportamento, sugere uma sazonalidade de demandas durante o *Sprint*, se por um lado o Analista fica sobrecarregado no começo do *Sprint*, no final é o testador que acaba deixando suas atividades formarem fila.

Outra informação retirada das simulações refere-se à utilização dos recursos. Para possibilitar uma análise sobre o correto dimensionamento de uma equipe de desenvolvimento, é possível analisar qual é o percentual do tempo que analistas, programadores e testadores permaneceram efetivamente realizando atividades, essa informação está compilada na imagem a seguir

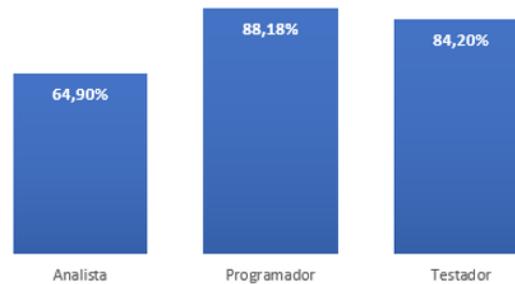


Figura 7. Percentual do tempo em atividade

Em uma primeira avaliação da figura 7, podemos pensar que o analista está sendo subutilizado, pois apenas 64,9% do seu tempo é gasto em atividades do *Sprint*. Porém, esse indicador não consegue mostrar a questão da sazonalidade das suas atividades, no início do *Sprint* ele possui todas as atividades em fila para avaliação enquanto programadores e testadores apenas aguardam suas atividades entrarem na fila.

5. Conclusão

O presente artigo buscou propor o uso de modelagem e simulação para abordar o problema do dimensionamento de equipes de desenvolvimento ágil por parte de gestores, hoje essa escolha é feita sem ferramental exigindo que o profissional faça escolhas baseado apenas em experiência profissional, de fato é possível criar um modelo que descreva uma equipe de desenvolvimento ágil com base nos dados históricos produzido pelo time, porém essa pesquisa apresentou também outros resultados igualmente interessantes.

Durante as simulações, ficou clara a natureza sazonal das atividades de análise, programação e teste. De modo que é difícil criar um dimensionamento estático da equipe que atenda a todo o período do *Sprint* de forma ideal, um exemplo disso é o caso do Analista de Sistemas, embora ele fique bastante ocupado no começo do projeto, da metade para o final do pacote sua fila de atividades acaba sumindo, totalizando apenas 64,9% do tempo em atividade. Uma sugestão é que nessa etapa do *Sprint*, o Analista receba treinamento para Testes, de modo que ele possa auxiliar o testador na etapa final da *Sprint*, otimizando o uso dos Recursos.

A metodologia proposta nesse artigo pode ser utilizada em outras empresas, desde a empresa possua base histórica com informações sobre tipo e tamanho das atividades, dadas essas premissas é possível implementar uma simulação usando filas considerando as distribuições probabilísticas que mais se aproximam aos dados da empresa em questão.

A simulação de equipes de desenvolvimento ágil pode proporcionar uma ferramenta importante para auxiliar em decisões administrativas. Seja em uma etapa inicial de dimensionamento do time de desenvolvimento ou em uma posterior avaliação sobre realocação de recursos. Este artigo não esgota esse tema de pesquisa, de modo que novos estudos se fazem necessários nessa área, alguns temas relevantes seriam, simular uma equipe na qual os integrantes possam exercer diversas atividades diferentes durante o *Sprint* ou também criar uma simulação que leve em considerações aspectos afetivos e

de relacionamento entre os integrantes do time.

Referências

Ahimbisibwe, A., Cavana, R. Y. and Daellenbach, U. (2015). A contingency fit model of critical success factors for software development projects: A comparison of agile and traditional plan-based methodologies. *Journal of Enterprise Information Management*, v. 28, n. 1, p. 7–33.

Aziz, J., Ahmed, F. and Laghari, M. S. (2009). Empirical analysis of team and application size on software maintenance and support activities. In *Information Management and Engineering, 2009. ICIME'09. International Conference on*.

Glass, R. L. (2001). Frequently forgotten fundamental facts about software engineering. *IEEE software*, n. 3, p. 110–112.

Keshta, N. and Morgan, Y. (oct 2017). Comparison between traditional plan-based and agile software processes according to team size & project domain (A systematic literature review). In *2017 8th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. IEEE. <http://ieeexplore.ieee.org/document/8117128/>, [accessed on Oct 4].

Pendharkar, P. C. and Rodger, J. A. (2009). The relationship between software development team size and software development cost. *Communications of the ACM*, v. 52, n. 1, p. 141–144.

Powell, A., Piccoli, G. and Ives, B. (2004). Virtual teams: a review of current literature and directions for future research. *ACM SIGMIS Database: the DATABASE for Advances in Information Systems*, v. 35, n. 1, p. 6–36.

Sharma, S. and Hasteer, N. (2016). A comprehensive study on state of Scrum development. In *Computing, Communication and Automation (ICCCA), 2016 International Conference on*.