

GERENCIAMENTO DE TEMPO COM BASE EM INFORMAÇÃO E METODOLOGIAS ÁGEIS

AGILE AND INFORMATION-BASED TIME MANAGEMENT

Arthur Arioli Bergamaschi – arthurbergmz@gmail.com
Jederson Donizete Zuchi – jederson.zuchi@fatec.sp.gov.br
Faculdade de Tecnologia de Taquaritinga (FATEC) – SP – Brasil

DOI: 10.31510/inf.v15i1.362

RESUMO

O gerenciamento de tempo consiste no conjunto de processos que asseguram a conclusão do projeto em um prazo previsto. Entretanto, segundo uma pesquisa da IBM, 90% dos dados existentes no mundo foram gerados desde 2012, sendo 2.5 bilhões de gigabytes por dia. Tomando como comparação esta ampla quantidade de informações geradas diariamente, a maioria dos projetos acaba perdendo o foco durante o decorrer do desenvolvimento por constantes mudanças no escopo - devido a novas informações levantadas por seu estado atual não ser o suficiente ou por problemas de comunicação, por exemplo, devido ao grande volume de informação que deve ser transitado durante todo o processo. Com isto em mente, o presente estudo foi elaborado por meio de revisão bibliográfica e um questionário, visando demonstrar o impacto causado ao fragmentar-se a ideia geral do projeto em tarefas menores, utilizando o Scrum como estrutura auxiliar.

Palavras-chave: Gerenciamento. Projeto. Tempo. Informação. Scrum.

ABSTRACT

Time management includes processes that ensures the delivering of a project in an expected deadline. However, according to an IBM's research, 90% of the existing data in the world were created since 2012, around 2.5 billions of gigabytes per day. Taking as comparison this huge amount of information daily generated, most of the projects end up losing focus during the development process due to constant scope changes - as new raising of information thanks to the current state of the project not being enough for development, or even due to communication problems, caused by lost of information due to the high rate being transitioned through the whole process. With that in mind, the present study was elaborated through a bibliographical review and a questionnaire, aiming demonstrating the impact caused by fragmenting the big picture of a project into smaller tasks, using Scrum as an auxiliary structure.

Keywords: Management. Project. Time. Data. Scrum.

1 INTRODUÇÃO

A “crise do *software*” (DIJKSTRA, 1972) identifica o momento em que dificuldades no desenvolvimento de *software*, frente à alta demanda, exigiam a resolução de problemas de

maneira eficaz e eficiente, levando à necessidade de criação de técnicas mais adequadas para o desenvolvimento de sistemas.

Como exemplos de novos métodos de gerenciamento, surgiram as metodologias ágeis. Em essência, estes tipos de metodologia têm como base uma ideia simples: permitir que as equipes tenham um feedback quase que imediato do trabalho realizado. “Eles estão na direção certa? O que planejam fazer em seguida é realmente o que deveriam fazer, considerando tudo que descobriram durante aquele ciclo?” (SUTHERLAND, 2015).

Segundo uma pesquisa da IBM (IBM, 2014), 90% dos dados existentes no mundo foram gerados desde 2012, sendo 2.5 bilhões de *gigabytes* por dia. Com esta ampla quantidade de informações geradas diariamente, fica claro que a tentativa de manipulação destes geram conflitos no gerenciamento de tempo em desenvolvimento de projetos. A necessidade de um filtro de informações úteis, assim como dar atenção a certos detalhes e coletar pontos cruciais para o projeto, caracterizam algumas das maiores dificuldades atuais de gestores.

Com isto em mente, este artigo se propõe a esclarecer como melhor gerenciar estes pontos, além de apresentar o Scrum como estrutura auxiliadora, evidenciando através de quais maneiras ela se mostra como um apoio para o gerenciamento de tempo. O referencial teórico introduz conceitos de gerenciamento de projetos de modo geral e o como o gerenciamento de tempo, que influencia tanto em seu êxito, está ligado à diversos fatores dentro de uma organização - como a comunicação eficiente.

Deste modo, é apresentado o Cone da Incerteza, a Teoria do Caos e como eles estão interligados. A partir disto, é exposto o que é desenvolvimento iterativo e incremental e como isto nos auxilia no andamento do planejamento a partir da estrutura Scrum. Por fim, se justifica este presente estudo com um estudo de caso de duas empresas na área de tecnologia da informação e como estas beneficiaram-se da estrutura Scrum para aprimorarem seus resultados, a fim de que os leitores possam inspirar-se em também aprimorar seus conhecimentos e conseguirem novas ideias para implementação de melhorias em seu ambiente de trabalho.

2 REFERENCIAL TEÓRICO

De acordo com PMBOK (PMI, 2012), gerenciamento de tempo é um conjunto de processos requeridos para assegurar a conclusão do projeto no prazo previsto. Para isto, são necessários cuidados constantes de gestão desde o planejamento até a entrega final. Ao saber quais as *features* que deverão ser executadas, é possível planejar o que deve ser feito para evitar atrasos na entrega e como diminuir o risco de erros irreversíveis, diminuindo, consequentemente, os gastos com um projeto, influenciando o êxito do projeto.

“As empresas de tecnologia da informação têm problemas muito comuns, como o tamanho da equipe e sua intercomunicação, que são pontos relativos” (COCKBURN, 2007). Devido a imprevistos, seja por falta de comunicação entre membros da equipe, conhecimento técnico insuficiente ou informações insatisfatórias, grande parte dos projetos ultrapassam o prazo previsto. Como se pode concluir através de uma pesquisa da PMI Brasil (PMI Brasil, 2008), alguns dos problemas mais frequentes são o não cumprimento de prazos, contabilizando 62%; problemas de comunicação, com 58% (Ilustração 1, 2 e 3); mudança de escopo, com 59%; e escopo não definido adequadamente, com 53%.

Ilustração 1 – Riqueza (“temperatura”) do canal de comunicação.

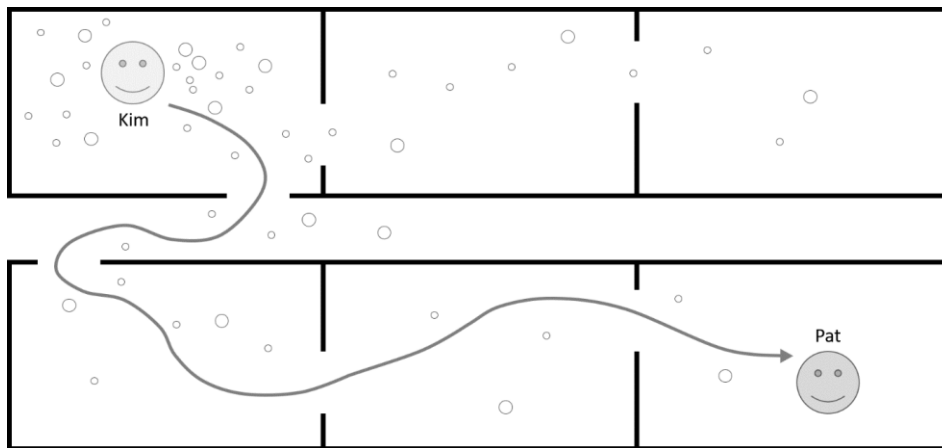


Fonte: COCKBURN (2001), adaptado pelo presente autor.

A Ilustração 1 mostra um gráfico comparando a efetividade dos canais de comunicação com a efetividade da comunicação empregada. COCKBURN (2001) diz que a forma mais efetiva de comunicação é de pessoas tendo uma conversa frente à frente,

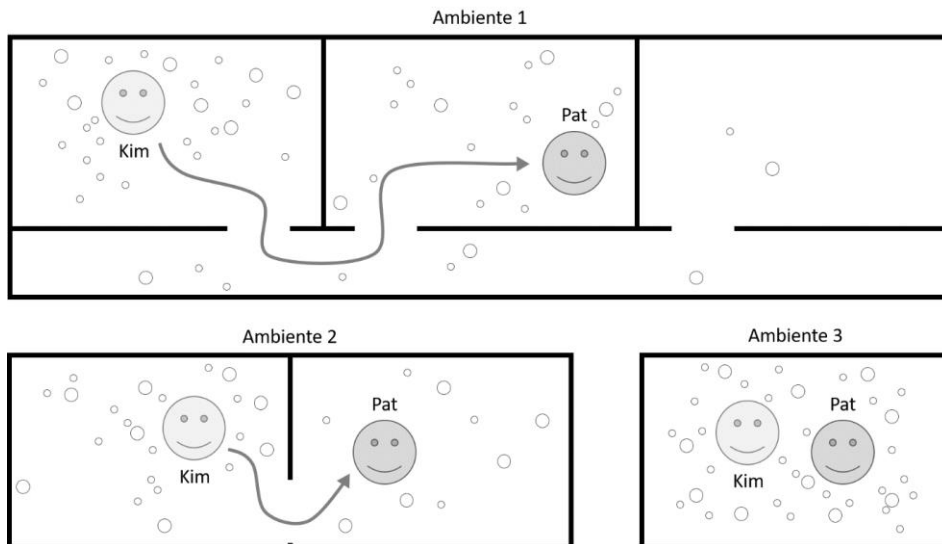
particularmente quando beneficiadas de um quadro branco. Conforme há o afastamento desta situação, a efetividade da comunicação decai.

Ilustração 2 – Comunicação exemplificada: energia e informação sendo repassadas através de barreiras complexas.



Fonte: COCKBURN (2001), adaptado pelo presente autor.

Ilustração 3 – Comunicação exemplificada: pessoas em três ambientes diferentes.



Fonte: COCKBURN (2001), adaptado pelo presente autor.

A Ilustração 2 e Ilustração 3 mostram como a “Comunicação Osmótica” (COCKBURN, 2001) é afetada quando duas pessoas que estão separadas por distância, paredes ou cubículos. A comunicação torna-se mais difícil e as informações se perdem ao

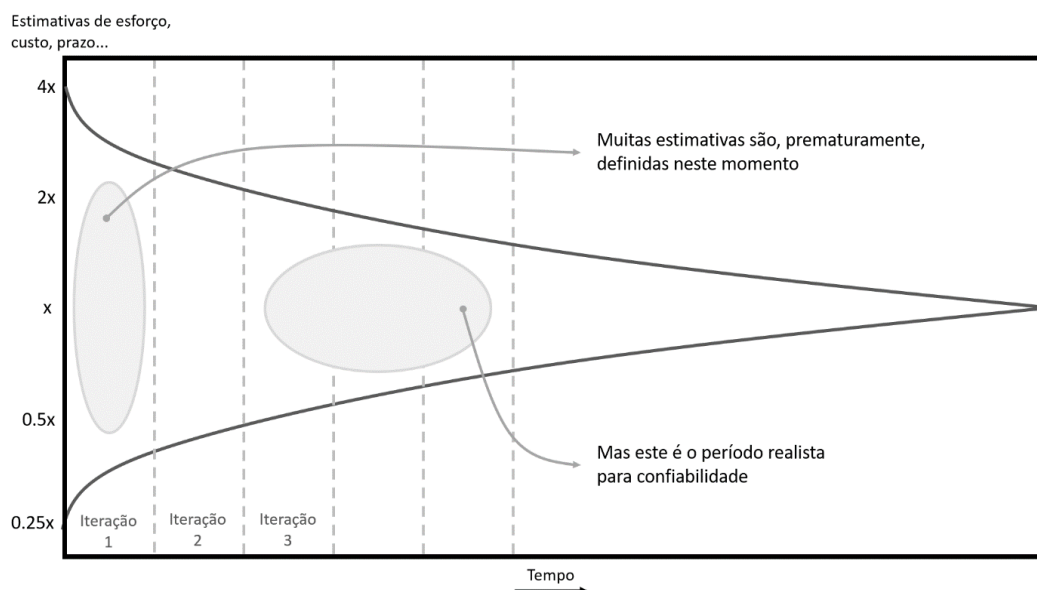
longo do caminho. Em um mesmo ambiente, como é possível observar no Ambiente 3 da Ilustração 3, a comunicação flui sem nenhum problema por ambas não terem obstruções no caminho dos dados transcorridos.

Os projetos também acabam tendo, segundo o *Exceeding Value* (The Standish Group International, 2014), 80% de funcionalidades e funções entregues de baixa a zero utilização, sendo apenas 20% de uso constante ou muito frequente. Segundo o princípio de Pareto, estes últimos 20% acabam representando cerca de 80% do valor do projeto. “Fazer as pessoas priorizarem de acordo com o valor as obriga a produzir aqueles 20% primeiro” (SUTHERLAND, 2014). É mais sensato, portanto, utilizar-se do tempo do início do projeto para coletar informações úteis e entender as necessidades do projeto, para que não se invista tempo em refatoração no produto final, por exemplo, devido à alguma funcionalidade que passou despercebida na elaboração do escopo.

2.1 Cone da incerteza e a teoria do caos

O Cone da Incerteza (MCCONNELL, 2006) é um gráfico onde se pode verificar a precisão das estimativas do projeto de acordo com seu ciclo de vida, conforme mostra a Ilustração 4.

Ilustração 4 – Cone da Incerteza.



Fonte: LARMAN (2003).

Um desafio enfrentado por quem tenta planejar adequadamente um projeto de *software* é o fechamento de um contrato com um cliente, por exemplo. Este gráfico ajuda a entender quando será o momento certo para realizá-lo, para que seja possível evitar dificuldades inesperadas.

No começo de um projeto, onde ainda não se dispõe de um escopo sólido – ou seja, com as funcionalidades primordiais definidas, grandes problemas serão enfrentados durante a produção, como apresentar um produto incompleto ou não ser possível atender o que fora requisitado por seus stakeholders, levando a uma elevada taxa de replanejamentos.

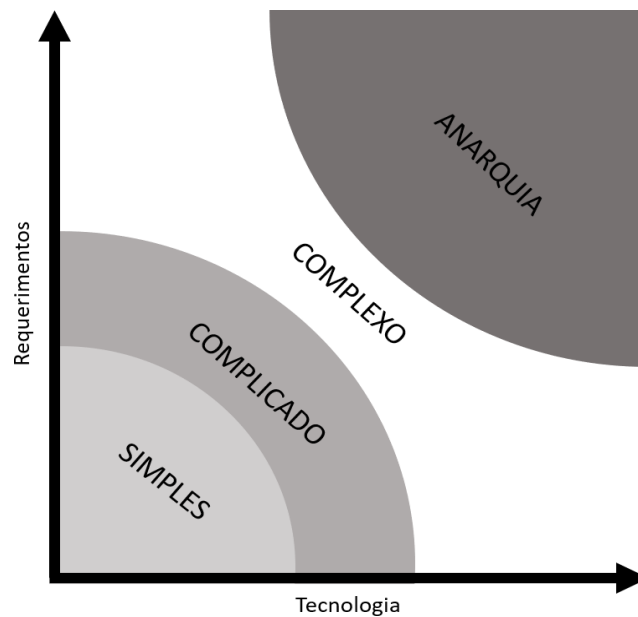
COHEN (2015) evidencia que “a Teoria do Caos trata de sistemas complexos e dinâmicos, rigorosamente deterministas, mas que apresentam um fenômeno primordial de instabilidade chamado de “sensibilidade às condições iniciais” que (...) torna-os não previsíveis na prática a longo prazo”.

Um sistema é um conjunto de elementos interconectados, de modo a formar um todo organizado (REISSWITZ, 2008). “Um sistema complexo é quando suas propriedades não são uma consequência natural de seus elementos constituinte” (MOSCHIN, 2015), como visto em rede sociais ou em uma colônia de animais, por exemplo. Já um sistema dinâmico “nasce da exigência de construir um modelo geral, para que todos seus elementos possam evoluir segundo uma regra, ligando o estado presente aos estados passados” (ASSIS; SOUZA, 2011).

“Em sistemas complexos e dinâmicos, como é o desenvolvimento de um *software*, determinados resultados podem ser instáveis no que se diz respeito à evolução temporal, como função de seus parâmetros e variáveis” (COHEN, 2015). Quaisquer simples erros na determinação do estado inicial e atual de um sistema podem ser ampliados pela não-linearidade ou pelo excesso de interações entre componentes, levando, deste modo, a resultados imprevisíveis. É o que se pode chamar, portanto, de “caos determinístico” (SCHUSTER, 1995) ou, simplesmente, “caos” (LORENZ, 2005, apud DANFORTH, 2013).

Através da Ilustração 5, é possível identificar o quão próximo um projeto está ao caos ao observar as zonas de decisão. Quanto mais requerimentos e tecnologia certo projeto demandar, mais perto da zona anárquica (ou zona de anarquia) estará. “Os métodos tradicionais de planejamento, visão e negociação são insuficientes neste contexto” (STACEY, 2001). Conseqüentemente, se o projeto entrar na zona anárquica, dificilmente será possível prosseguir sem bom plano de negócios, organização e flexibilidade.

Ilustração 5 – Matriz de Stacey.



Fonte: STACEY (2001), adaptado pelo presente autor.

Segundo o autor (STACEY, 2001), quanto mais próximo à zona simples, mais fácil conseguir entrar em consenso com os envolvidos no desenvolvimento e mais clara a certeza de que o projeto está andando como deveria.

Jim Highsmith (HIGHSMITH, 2009) julgou o que seria essencial em uma organização: a capacidade de criar alterações e responder favoravelmente a quaisquer mudanças. Estes benefícios são apenas possíveis em um ponto específico de equilíbrio entre estabilidade e flexibilidade. Seguindo esta ideologia, introduzimos, a partir deste momento, os princípios ágeis e a estrutura Scrum.

2.2 Princípios ágeis

Os princípios ágeis de desenvolvimento de *software* foram apresentados pela Aliança Ágil, que surgiu em 2001 com 17 especialistas em processos de desenvolvimento de *software*, ao estabelecerem o Manifesto Ágil.

A partir deste momento, foram definidas certas prioridades. Uma delas é ter a satisfação do cliente por meio de entregas contínuas e antecipadas de um produto válido, buscando ver indivíduos interagindo sobre o produto acima de processos e ferramentas. Outra é ter o *software* funcionando, sem a necessidade de documentação extensa e confusa. Ter

maior aproximação do cliente, ao contrário de constantes negociações de contratos, também é de extrema importância. Por fim, é ter e respostas rápidas às mudanças, ao invés de seguir planos previamente definidos, é vital para o bom desempenho do andamento do desenvolvimento. (SUTHERLAND, 2014).

2.2.1 Desenvolvimento iterativo e incremental

O desenvolvimento iterativo e incremental é um processo de desenvolvimento de *software* que prevê planejamento em estágios, desenvolvendo e entregando o produto em incrementos de *software* funcionais, promovendo previsões regulares para reflexão da equipe, beneficiando-se de múltiplos ciclos chamados “PDCA” (planejar, desenvolver, checar e agir) (IMAI, 2014).

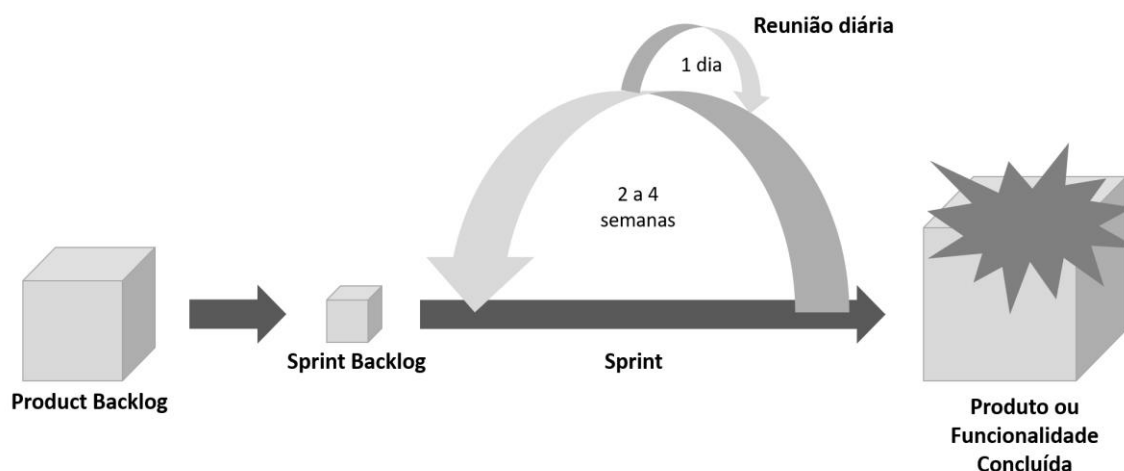
Em essência, a metodologia propõe que, ao começar um projeto, paradas regulares sejam realizadas para verificar se o que está sendo feito está alinhado com o proposto e se os resultados são os que seus clientes desejam, verificando se existem maneiras de aprimorar a forma de como se está trabalhando para obter melhores resultados e quais seriam os obstáculos enfrentados pelos envolvidos. Isto é chamado de “ciclo de inspeção e adaptação” (SUTHERLAND, 2014).

2.2.2 Estrutura scrum

A estrutura Scrum (SUTHERLAND, 2014) foi idealizada por Jeff Sutherland, formalizado por Ken Schwaber, para ser uma forma mais rápida, eficaz e confiável de criar *softwares* para o setor de tecnologia. Até então, a maior parte do desenvolvimento de programas era feita utilizando como base o método em cascata, levando a processos imprevisíveis, não-flexíveis e, em geral, dificilmente resultava em um produto satisfatório. “O Scrum é uma mudança radical das metodologias prescritivas” (SCHWABER, 2016), já que sua estrutura se tornou uma das melhores formas no setor tecnológico para criar novas soluções, permitindo alta flexibilidade, para que empresa e cliente possam buscar, juntos, o melhor produto possível.

Todo o processo sugerido pela estrutura Scrum foi exemplificado por LARMAN (2003) e pode ser observado na Ilustração 6.

Ilustração 6 – Processo Scrum exemplificado.



Fonte: LARMAN (2003), adaptado pelo presente autor.

Sem a necessidade inicial de um escopo totalmente elaborado, o Scrum funciona com apenas as funções primordiais do *software* definidas – o chamado *Product Backlog* (SUTHERLAND; SCHWABER, 2017).

Devido a sua estrutura altamente modificável, é possível adicionar novas *features* durante o desenvolvimento, definindo prioridades e prazos. Por meio dos *Sprints* (SUTHERLAND, 2014) é possível entregar um *Increment* (SUTHERLAND; SCHWABER, 2017).

Sprints são um conjunto de *features* definidas (SUTHERLAND; SCHWABER, 2017) que deverão ser realizadas até o *Product Backlog* estar sem mais *features* previstas. Ocorrem em um período de, geralmente, duas a quatro semanas.

Segundo os mesmos autores, um *Increment* é o produto gerado pela *Development Team*, com as funcionalidades necessárias para que ele cumpra as funções planejadas até aquele momento. A partir do *Increment* é possível validar, por exemplo, a eficiência do produto, sua usabilidade e aceitação no mercado.

Há três papéis para a estrutura Scrum funcione corretamente. O primeiro papel é o *Product Owner* (SUTHERLAND; SCHWABER, 2017) – responsável por definir o que precisa ser feito, quais funcionalidades serão implementadas e definir quais itens irão compor o *Product Backlog*. O segundo papel é do *Scrum Master* (SUTHERLAND; SCHWABER, 2017), que procura assegurar que a equipe respeite e siga os valores e práticas do Scrum, além de assegurar que a equipe não se comprometa excessivamente com relação àquilo que é capaz de realizar durante um *Sprint*. Ele atua, também, como facilitador do *Daily Scrum*, e é

responsável por tentar remover quaisquer obstáculos que sejam levantados pela equipe durante estas reuniões. O terceiro papel, por fim, é a **Development Team** (SUTHERLAND; SCHWABER, 2017), que é a equipe multidisciplinar de desenvolvimento do projeto que irá, de fato, produzir o *Increment* da *Sprint*.

A estrutura Scrum também apresenta três artefatos. O primeiro artefato é o **Product Backlog** (SUTHERLAND; SCHWABER, 2017), que é uma lista de funcionalidades a serem feitas ao longo da vida útil do projeto, com *features* previamente priorizadas. O segundo artefato é o **Sprint Backlog** (SUTHERLAND; SCHWABER, 2017), onde são definidas as funcionalidades de maior prioridade, com a equipe comprometendo-se a finalizá-las até o próximo *Sprint*. O terceiro artefato é um método de acompanhamento do progresso da *Sprint* atual (SUTHERLAND; SCHWABER, 2017), escolhida pela *Development Team*. Um exemplo de método de acompanhamento de progresso é o **Burndown Chart**, que oferece uma resposta visual de tarefas restantes e concluídas. Ele sempre almeja a chegar a 0 (zero) – ou seja, sem tarefas restantes – o quanto antes do *Sprint* acabar.

Há, também, quatro reuniões. A primeira reunião é a **Sprint Planning** (SUTHERLAND; SCHWABER, 2017), que é quando o *Project Owner*, o *Scrum Master* e a *Development Team* se reúnem para discutir as funcionalidades que serão implementadas no *Sprint* que será iniciado, considerando suas dificuldades e prioridades, buscando seus objetivos até o próximo *Sprint* a ser realizado. Aqui, a equipe faz diversas perguntas ao *Project Owner*, afim de quebrar as funcionalidades em tarefas técnicas. Estas tarefas darão origem ao *Sprint Backlog*.

A segunda reunião é a **Daily Scrum** (SUTHERLAND; SCHWABER, 2017) é uma reunião rápida e diária – onde todos da *Development Team* discutem o que foi feito desde a última *Daily Scrum* que fora realizada, dizem no que estão trabalhando atualmente e se estão sendo impedidos de fazer seu trabalho devido a alguma razão ou se precisam de algum outro tipo de ajuda. Entretanto, não há espaço para discussões técnicas.

A terceira reunião é a **Sprint Review** (SUTHERLAND; SCHWABER, 2017), que ocorre ao final de cada *Sprint*. É aqui onde a equipe mostra o que foi concluído ao *Project Owner*, quais foram suas dificuldades, o que foi proveitoso e quais serão os próximos passos a serem tomados no projeto.

Por fim, através da quarta reunião – a chamada **Sprint Retrospective** (SUTHERLAND; SCHWABER, 2017), é possível inspecionar necessidades de adaptações

durante o processo de desenvolvimento. Ela sempre acontece depois de uma *Sprint Review* e antes de uma nova *Sprint Planning*.

3 PROCEDIMENTO METODOLÓGICO EM PRÁTICA

Juntamente com o desenvolvimento deste estudo, realizou-se uma pesquisa via e-mail (o questionário e as respostas dos entrevistados podem ser acessados por meio do endereço: <https://gist.github.com/arthurbergmz/a2fa3e4e86fe5cc8f91e5e522326b331>) com duas empresas diferentes no ramo de tecnologia da informação que utilizam o *Scrum* para gerenciar seus projetos, com seus respectivos *Scrum Masters*, visando o gerenciamento da informação. A Empresa A possui aproximadamente um ano e seis meses e é uma *startup*, possuindo apenas um aplicativo como produto, enquanto a Empresa B possui 29 anos, consolidada no mercado por prestar serviços de consultoria e desenvolvimento de *software*.

4 RESULTADOS ANALISADOS

Apesar dos entrevistados identificarem seus pontos em relação à estrutura de maneiras diferentes, ainda é possível elucidar as melhorias que o *Scrum* traz para o gerenciamento de tempo.

De acordo com o entrevistado, a Empresa A costumava gerenciar a empresa sem nenhum tipo de metodologia e, conforme seu crescimento, mostrou-se necessário um melhor gerenciamento geral, decidindo-se implantar o *Scrum* como estrutura de desenvolvimento ágil. A empresa procurava por entregas mais rápidas e consistentes, assim como mais clareza nos processos realizados em cada entrega. Após a implantação - que ocorreu em março de 2017, o entrevistado relatou melhorias tanto na clareza dos processos em realização e nos que ainda seriam realizados, quanto clareza nos objetivos da própria empresa. Também foram notadas melhorias na comunicação entre todas as áreas da empresa, possibilitando equipes mais motivadas e engajadas. Como consequência, as equipes de desenvolvimento ganharam maior confiança da gerência.

No caso da Empresa B, a cultura ágil, utilizando a estrutura *Scrum*, foi implantada há 8 anos, enquanto a empresa já possuía 21 anos de existência. Como a Empresa B também

presta consultorias em outras áreas além de desenvolvimento de *software*, como de *Business Intelligence* e implantação de novas tecnologias, os problemas relatados foram um pouco diferentes.

Previsões erradas, desde previsões de negócios às previsões de entregas dos projetos, assim como improdutividade por parte das equipes, eram obstáculos para a Empresa B. Com a implantação da estrutura na empresa, o conhecimento sobre os processos e problemas do cliente se tornaram melhores e mais claros, ajudando a evitar problemas de gerenciamento de escopo. Graças ao acompanhamento constante dos processos e da visão dos clientes, aproveitava-se do feedback gerado nas reuniões do Scrum. A hierarquia das equipes de desenvolvimento também diminuiu, melhorando os relacionamentos, reduzindo os atrasos de entregas e os problemas de improdutividade.

O processo de implantação da estrutura também se deu de diferentes formas nas duas empresas. A Empresa A passou por dificuldades na mudança de cultura da empresa, já que as equipes realizavam seu trabalho daquela maneira há muito mais tempo, portanto, acreditavam que aquela já era a melhor alternativa. Já a Empresa B não enfrentou grandes dificuldades na implantação da estrutura, já que o Scrum é indicado para empresas do ramo.

Já como facilidade, a Empresa A possuía o apoio do *Chief Information Officer* (CIO), ou chefe do departamento de tecnologia da informação, e apesar da resistência dos desenvolvedores ao aderirem o processo cultural, todos sabiam que seria a melhor escolha para o gerenciamento. As facilidades para a implantação da estrutura na Empresa B vieram diretamente do foco da empresa e de como as equipes veem as suas relações com o cliente, muitas vantagens nessas áreas fizeram com que a empresa toda abordasse a estrutura com empenho.

Entre as vantagens citadas com a implantação, destacam-se a transparência dos processos, que trouxe visões reais dos problemas, como o fato de que as previsões não são perfeitas e tornou possível explicar para o cliente a dificuldade em realizar algum de seus requisitos, ou até mesmo a impossibilidade. Além dessas vantagens, foi possível reagir aos problemas muito antes ou desistir de ideias que não faziam sentido. Ao ouvirem de todos os envolvidos, a resolução de dificuldades tornou-se mais fácil. Ter o cliente mais próximo do desenvolvimento foi crucial para o êxito, pois é plausível que o cliente conheça melhor as dificuldades do processo, permitindo que as equipes conheçam melhor suas necessidades. Este relacionamento criou um espaço fértil para soluções, de fato, na empresa.

Com essa análise foi possível perceber as vantagens e melhorias que metodologias ágeis e, em especial, a estrutura Scrum, causam em qualquer tipo de gerenciamento, não apenas em tempo, mas também no relacionamento com o cliente, por meio de uma melhor gestão de comunicação, com a análise constante de *feedbacks*. Já nos gerenciamentos de tempo e risco a melhoria é contínua, pois são permitidos acompanhamentos que trazem melhores previsões de negócios e entregas, tomando posições proativas em relação ao gerenciamento de riscos, evitando problemas ou trazendo soluções rápidas e eficazes.

Além de todas estas áreas, a estrutura é aliada em outro tipo de gerenciamento: o de informação. Como já citado anteriormente, as reuniões definidas ajudam a impedir que ocorra dispersão do foco, os assuntos são tratados cada um em seu momento e a análise contínua de *feedbacks* evitam perda de informações importantes e excesso de informações desnecessárias.

Gerenciar tempo, normalmente, leva à paralelização das tarefas, causando o que pode ser chamado de “troca de contexto”, o que é extremamente prejudicial às tarefas de alta demanda criativa e/ou complexas, como o próprio desenvolvimento de *software*. Já para tarefas administrativas, de gestão ou comerciais, priorização e algum nível de paralelização podem ser utilizados.

O Scrum, por fim, é uma estrutura para desenvolvimento ágil, com foco em gerenciamento de tempo, mas que traz consigo melhorias em todas as áreas de gerenciamento.

5 CONCLUSÃO

O foco inicial deste projeto era mostrar como o gerenciamento da informação influencia no gerenciamento de tempo, comunicação e todas as outras áreas de gerenciamento de projeto. O presente autor acredita que com metodologias ágeis, ao fragmentar-se a ideia geral de um projeto em tarefas menores - como através dos *Sprints* e *Increments* na estrutura Scrum, por exemplo, é possível que o contexto anárquico seja lidado com mais facilidade, ao contrário de metodologias que não oferecem tal benefício.

Ou seja, apesar de não ser possível diminuir o nível caótico de um projeto devido sua complexidade, é possível lidar melhor com todo o processo em si ao visar cada passo do desenvolvimento sendo executado individualmente, ao invés de visar apenas o produto final. E, para isso, é necessário tratar cada *Sprint* finalizada com a mesma importância de um produto final.

Deste modo, ao se trabalhar apenas com informações sucintas, coesas e necessárias em cada momento específico do desenvolvimento, o progresso geral e o controle das informações tornam-se simples e eficazes, com direções mais claras e um maior entendimento do produto.

REFERÊNCIAS

ASSIS, L. F. de; SOUZA, W. A. de. **Introdução aos sistemas dinâmicos discretos**. 63ª Reunião Anual da Sociedade Brasileira para o Progresso da Ciência, Goiânia - GO, 2011. ISSN 2176-1221. Disponível em: <<http://www.sbpcnet.org.br/livro/63ra/resumos/resumos/5577.htm>>. Acesso em 14 de maio de 2018.

COCKBURN, A. **Agile Software Development**, editora Addison-Wesley Professional, 1ª edição, 2001. 304 páginas.

COHEN, R. **Métricas para Help Desk e Service Desk**, editora Novatec, 1ª edição, 2015. 232 páginas.

DANFORTH, C. M. **Chaos in an Atmosphere Hanging on a Wall**, Mathematics of Planet Earth, 2013. Disponível em: <<http://mpe2013.org/2013/03/17/chaos-in-an-atmosphere-hanging-on-a-wall/>>. Acesso em 14 de maio de 2018.

DIJKSTRA, E. W. **The Humble Programmer**, ACM 15, v. 10, p. 859–866, 1972. Disponível em: <<https://www.cs.utexas.edu/users/EWD/ewd03xx/EWD340.PDF>>. Acesso em 9 de maio de 2018.

HIGHSMITH, J. **Agile Project Management: Creating Innovative Products**, editora Addison-Wesley Professional, 2ª edição, 2009. 392 páginas.

IBM. **Spectrum Scale Slidecast**, 2015. 6 páginas. Disponível em: <<https://pt.slideshare.net/insideHPC/ibm-spectrum-scale-slidecast>>. Acesso em 7 de maio de 2018.

IMAI, M. **Gemba Kaizen. Uma Abordagem de Bom Senso à Estratégia de Melhoria Contínua**, editora Bookman, 2ª edição, 2014. 424 páginas.

LARMAN, C. **Agile and Iterative Development: A Manager's Guide**, editora Addison-Wesley Professional, 1ª edição, 2003. 368 páginas.

MCCONNELL, S. **Software Estimation: Demystifying the Black Art (Developer Best Practices)**, editora Microsoft Press, 1ª edição, 2006. 308 páginas.

MITCHELL, M. **Complexity: A Guided Tour**, editora Oxford University Press, 1ª edição, 2011. 349 páginas.

MOSCHIN, J. **Gerenciamento de Parada de Manutenção - Um Projeto de Sucesso ao Alcance de Suas Mãos**, editora Brasport, 1ª edição, 2015. 278 páginas.

PMI. **A Guide to the Project Management Body of Knowledge (PMBOK Guide)**, editora Project Management Institute, 5ª edição, 2012. 589 páginas.

PMI Brasil. **Benchmarking GP**, 2008. 127 páginas. Disponível em: <<https://pt.slideshare.net/FelipeGuedesPinheiro/pequisa-benchmarking-em-gerenciamento-de-projetos-brasil-2008>>. Acesso em 14 de maio de 2018.

REISSWITZ, F. **Análise de Sistemas - Vol 1**, editora Clube de Autores, 1ª edição, 2012. 103 páginas.

SCHUSTER, H. G. **Deterministic Chaos: An Introduction**, editora Wiley-VHC, 3ª edição, 1995. 320 páginas.

STACEY, R. **Complex Responsive Processes in Organizations: Learning and Knowledge Creation**, editora Routledge, 1ª edição, 2001. 272 páginas.

STANDISH Group, **Exceeding Value**, 2014. 4 páginas. Disponível em: <https://www.standishgroup.com/sample_research_files/Exceeding%20Value_Layout.pdf>. Acesso em 9 de maio de 2018.

SUTHERLAND, J. **Scrum: a arte de fazer o dobro do trabalho na metade do tempo**, editora Leya Brasil, 1ª edição, 2014. 240 páginas.

SUTHERLAND, J; SCHWABER, K. **Scrum Guides**. 2017. Disponível em: <<https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf>>. Acesso em 14 de maio de 2018.