



**V SINGEP**

**Simposio Internacional de Gest3o de Projetos, Inova3o e Sustentabilidade**  
**International Symposium on Project Management, Innovation and Sustainability**

ISSN: 2317 - 8302

## **Ger4ncia de Projetos 3geis: Um estudo sobre as potencialidades do framework SAFe**

**VINÍCIUS ROGGIA GOMES**

UFSC

vinigomes@outlook.com

**GUILHERME MARTINS ALVAREZ**

UFSC

guilherme.alvarez@gmail.com

**LIA CAETANO BASTOS**

UFSC

lia.c.bastos@ufsc.br



V SINGEP

Simpósio Internacional de Gestão de Projetos, Inovação e Sustentabilidade  
International Symposium on Project Management, Innovation and Sustainability

ISSN: 2317 - 8302

## GERÊNCIA DE PROJETOS ÁGEIS: UM ESTUDO SOBRE AS POTENCIALIDADES DO FRAMEWORK SAFE

### Resumo

Com o crescimento das organizações e as cobranças diárias por resultados, surgem novos desafios no âmbito da gerência de projetos de Tecnologia da Informação. O mercado se tornou mais exigente, então é necessário entregar mais valor aos clientes e responder de forma rápida a mudanças do negócio, reduzindo custos e aumentando a eficiência dos times de desenvolvimento. Assim, faz-se necessário expandir e escalar a estrutura ágil a nível de organização, a fim de que se consiga englobar equipes maiores, sincronizar e alinhar os times de desenvolvimento e atingir todos os níveis hierárquicos dessas instituições. Nesse contexto, utilizar uma metodologia ágil como o SAFe (Scaled Agile Framework) pode ser a solução, pois ele reúne boas práticas ao aplicar o ágil em escala empresarial nos níveis operacional, tático e estratégico. O presente trabalho tem como objetivo, levantar referencial teórico acerca da gerência de projetos ágeis, metodologia *Lean Thinking*, metodologias ágeis, e posteriormente, apresentar o SAFe (Scaled Agile Framework). Verifica-se que quando uma organização progride e seus times crescem, é o momento de pensar em escalar a estrutura ágil, e o SAFe provê um caminho claro de como atuar nessa tarefa.

**Palavras-chave:** Gerência de Projetos; Metodologias Ágeis; SAFe; Lean Thinking.

### Abstract

With the growth of organizations and daily charges for results, new challenges arise in the management of information technology projects. The market has become more demanding, so it is necessary to deliver more value to customers and respond quickly to business changes, reducing costs and increasing the efficiency of development teams. Thus, it is necessary to expand and scale agile structure its organization, so that it can cover larger teams, synchronize and align the development teams and reach all levels of these institutions. In this context, using an agile methodology as SAFe (Scaled Agile Framework) may be the solution, because it brings together best practices in applying agile enterprise-scale in operational, tactical and strategic level. This study aims, raise theoretical framework on the management of agile projects, Lean Thinking, agile methodologies, and subsequently present the SAFe (Scaled Agile Framework). It is found that when an organization progresses and their teams grow, it's time to think about scaling agile structure, and SAFe provides a clear path how to act in this task.

**Keywords:** Project Management; Agile Methodology; SAFe; Lean Thinking.



## 1. Introdução

As metodologias ágeis surgiram em meados dos anos 80, devido as crescentes exigências do mercado por produtividade, inovação, flexibilidade e melhoria na qualidade dos projetos de desenvolvimento de software (MARUPING; VENKATESH; AGARWAL, 2009). Mas apenas em 2001 a expressão “Metodologias Ágeis” se tornou conhecida, quando especialistas em processos de desenvolvimento de software estabeleceram princípios e características comuns entre as metodologias, criando assim o “Manifesto Ágil”.

A necessidade de se gerenciar projetos de software complexos, com incertezas e imprevisibilidade, promoveu uma crescente demanda na utilização dessas novas metodologias. Pois, as abordagens prescritivas e lineares não oferecem as características necessárias para responder de forma ágil as mudanças necessárias, afetando diretamente o custo e prazo do projeto.

De acordo com Vargas (2005), com o intuito de atender as demandas de maneira eficiente, em um ambiente marcado pela velocidade de mudanças, torna-se indispensável a utilização de um modelo de gerenciamento baseado em objetivos e prioridades. O autor ainda afirma que, outro fator que impulsiona o gerenciamento de projetos é o crescimento da competitividade, ou seja, quem for mais rápido e competente, provavelmente conseguirá melhores resultados.

Estamos vivendo uma tendência para o desenvolvimento ágil de aplicações, devido ao ritmo acelerado das mudanças na área da Tecnologia da Informação, pressões por constantes inovações, concorrência acirrada e grande dinamismo no ambiente de negócios (BOEHM, 2006).

As metodologias ágeis vêm sendo aplicadas nos mais diversos tipos de projetos na área de Tecnologia da Informação, em diferentes domínios, nas suas mais diversas formas como, Scrum, XP (*eXtreme Programing*), Kanban, entre outros.

Verifica-se que a adoção dessas metodologias pode trazer benefícios para as organizações, porém a sua aplicação sem o devido conhecimento e cautela, pode criar barreiras para se alcançar os objetivos esperados. Nesse sentido, pode-se citar, a baixa de efetividade na utilização de metodologias ágeis, falta de sincronia e comunicação nos times ágeis e desmotivação dos participantes.

Observando os problemas acima citados, questiona-se como fazer para aumentar a agilidade e efetividade na gerência de projetos utilizando metodologias ágeis? Como auxiliar o time de desenvolvimento de software a obter o melhor proveito dessas metodologias?

Com o crescimento de uma organização e as cobranças diárias por resultados, faz-se necessário escalar a estrutura ágil, a fim de que se consiga englobar equipes maiores e atingir todos os níveis hierárquicos dessas instituições. Nesse contexto, uma metodologia ágil como o SAFe (Scaled Agile Framework) pode ser a solução, pois ele reúne boas práticas ao aplicar o ágil em escala empresarial nos níveis operacional, tático e estratégico.

O SAFe apresenta firmes propósitos, como melhorar a produtividade, competitividade e qualidade do produto, além de aumentar a motivação, engajamento e satisfação dos empregados da organização (SAFe, 2015).

Esse trabalho tem como objetivo, levantar referencial bibliográfico acerca da gerência de projetos, do *Lean Thinking* e das metodologias ágeis. Além disso, é apresentado o framework SAFe, amplamente utilizado para implementar o desenvolvimento Lean/Ágil de sistemas em escala empresarial.

Por meio de uma pesquisa exploratória, envolvendo levantamento bibliográfico, serão elencadas e discutidas algumas questões envolvendo gerência de projetos, metodologias ágeis de desenvolvimento e a utilização do framework SAFe.



O trabalho está organizado em quatro seções principais. Após a introdução, a segunda apresenta a fundamentação teórica relacionada a *Lean Thinking* e Metodologias Ágeis, como Scrum (vão apresentar somente uma? Então porque metodologias ágeis no plural). A terceira seção descreve o framework SAFe e orienta como ele pode ser utilizado nas organizações. Por fim, na última seção, são apresentadas as considerações finais.

## 2. Referencial Teórico

Nesta seção é apresentado os principais temas relacionados com o desenvolvimento de projetos com metodologias ágeis, com o objetivo de embasar a análise sobre o framework SAFe e o escalonamento de projetos ágeis.

### 2.1 *Lean Thinking* (Mentalidade Enxuta)

De acordo com Poppendieck e Poppendieck (2007) a origem do *Lean Thinking* (ou Mentalidade Enxuta) se deu no chão de fábrica japonês, a partir do Sistema Toyota de Produção. Ohno (1988) esclarece que se trata de um sistema para absoluta eliminação do desperdício, baseado no fluxo Just-in-Time e na autonomia (princípio de *Jidoka*), ou seja, automação inteligente.

Ao longo dos anos, o *Lean Thinking* mudou seu foco inicial ligado a redução de desperdícios e custos para uma abordagem que busca aumentar o valor para os clientes (HINES; HOLWEG; RICH, 2004). Nesse sentido, o termo valor pode ser considerado como a capacidade de entregar o produto ou serviço desejado pelo cliente, a um preço adequado e com o menor tempo entre o momento em que o cliente faz seu pedido e a entrega.

O *Lean Thinking* fornece uma maneira de especificar valor, alinhar as ações de criação de valor, realizar essas atividades sem interrupção e executá-las de forma mais eficaz. Além disso, esse paradigma proporciona uma maneira de tornar o trabalho mais satisfatório, fornecendo feedback imediato sobre os esforços realizados para converter o desperdício em valor (WOMACK; JONES, 2005)

A aplicação desta metodologia tem proporcionado ótimos resultados para as empresas que a praticam, ainda que poucas tenham conseguido repetir integralmente o sucesso e a eficiência operacional alcançados pela Toyota (POPPENDIECK; POPPENDIECK, 2007).

Moreira, Alves e Sousa (2010) afirmam que esse paradigma tem como objetivo alcançar uma alta produtividade e sincronizar as demandas e a sua produção. De acordo com Womack e Jones (1996), para atingir estes objetivos, cinco princípios foram estabelecidos:

- **Especificar o valor** - entender o ponto de vista dos clientes e o que é valor para eles. Tem como objetivo construir uma comunicação clara e uma compreensão compartilhada, a fim de entregar o que o cliente necessita;
- **Identificar o fluxo de valor** - tem como objetivo remover os desperdícios de processos, ou seja, identificar os processos que geram valor, os que não geram valor, mas são importantes para a manutenção dos processos e da qualidade, e por fim, aqueles que não agregam valor algum, devendo estes serem eliminados;
- **Fluxo Contínuo de valor** - a fim de eliminar os desperdícios, os processos têm que ser alterados e reorganizados de forma que o produto ou serviço flua através de todos os passos, da forma mais eficaz e eficiente possível.
- **Produção Puxada** – a partir da compreensão das exigências e necessidades dos clientes, pode-se construir fluxos de processos que atendam essas demandas. Assim, entregando



o que o cliente precisa, quando precisa e no lugar que precisa. O consumidor passa a puxar o fluxo de valor, portanto não se deve produzir um bem ou serviço até que o cliente o solicite;

- **Perfeição** - todos os envolvidos nos fluxos de valor devem ter a perfeição como objetivo. O aperfeiçoamento contínuo, objetivando um estado ideal deve nortear os esforços. O mundo muda constantemente, assim os processos precisam continuar a satisfazer as necessidades e as demandas de mudança.

LIB (2015, p.1) conclui que o *Lean Thinking* tem como um dos principais resultados o “aumento da capacidade de oferecer os produtos que os clientes querem, na hora em que eles precisam, aos preços que estão dispostos a pagar, com custos menores, qualidade superior e *lead times* mais curtos, garantindo, assim, maior rentabilidade para o negócio”.

## 2.2 Metodologias Ágeis

Nos últimos anos, metodologias ágeis surgiram como uma alternativa as formas tradicionais de desenvolvimento de software. Um aspecto que contribuiu para o seu crescimento foi a contínua necessidade de mudança, impulsionada pelas constantes alterações de mercado (IANSITI; MACCORMACK, 1997; LEE; XIA, 2005).

Na visão de Maruping, Venkatesh e Agarwal (2009), ágil é um método de gestão e desenvolvimento de software que utiliza uma abordagem iterativa e incremental voltada para processos empíricos, dividindo o problema em produtos menores e visando entregar um incremento funcional de software.

Para Highsmith (2004, p. 16) a agilidade em TI (Tecnologia da Informação) é “ a habilidade de criar e responder a mudanças, buscando a obtenção de lucro em um ambiente turbulento”.

Em 2001, foi concebido o Manifesto Ágil, que relaciona 12 princípios que fundamentam o desenvolvimento ágil de software. Esse documento foi assinado por Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas no ano de 2001.

De acordo com BOEHM (2002), para o Manifesto ágil, os indivíduos e suas interações, a colaboração com o cliente, o funcionamento do software e sua capacidade em responder de maneira efetiva a mudanças, são mais importantes que a utilização de ferramentas, processos, planos, documentação e contratos.

As metodologias ágeis visam a redução dos riscos relacionados a incertezas de projeto, a colaboração do time de desenvolvimento com os especialistas de negócios, responder de forma eficiente as mudanças de projeto e a satisfação dos clientes (PAULK, 2002).

Beck (1999) afirma que metodologias ágeis são utilizadas para fornecer flexibilidade no desenvolvimento de software, assim, permitindo que as equipes de desenvolvimento de software lidem com um ambiente imprevisível e cheio de mudanças.

Existem diversas metodologias e frameworks para o desenvolvimento ágil de software, entre eles podemos citar: eXtreme Programming (XP) (BECK, 1999), Scrum (SCHWABER; SUTHERLAND, 2013), Lean Development (POPPENDIECK; POPPENDIECK, 2007), Feature Driven Development (FDD) (COAD et al., 1999), entre outras. Uma das mais utilizadas é o Scrum, a qual será apresentada em seguida.



## 2.3 Scrum

Para Schwaber (1997), o Scrum é baseado nas teorias empíricas de controle de processos, ou empirismo. Para o empirismo, o conhecimento provém unicamente da experiência e de tomadas de decisão apoiadas no que se pode captar do mundo externo.

O Scrum trabalha com a complexidade de desenvolvimento de software através do controle de inspeção, adaptação e visibilidade de requisitos de um processo, fazendo o uso de uma série de regras e práticas (BALLE, 2011). Rising e Janoff (2000) afirmam que o Scrum utiliza uma abordagem iterativa e incremental para aprimorar a previsibilidade e o controle de riscos.

Barbosa (2010), afirma que para alcançar seus objetivos, o método Scrum emprega uma estrutura na qual, no início de cada iteração, a equipe analisa o que deve ser feito e então seleciona aquilo que acreditam que pode se tornar um incremento de valor ao produto no término da iteração. Então, a equipe dedica-se totalmente para desenvolver a iteração e ao final, apresenta o incremento de funcionalidade construído para os *stakeholders*.

De acordo com Pereira et. al. (2007), o ciclo de desenvolvimento do Scrum é denominado de Sprint, com durações de 2 a 4 semanas. A Figura 1 ilustra a Sprint e seus eventos.

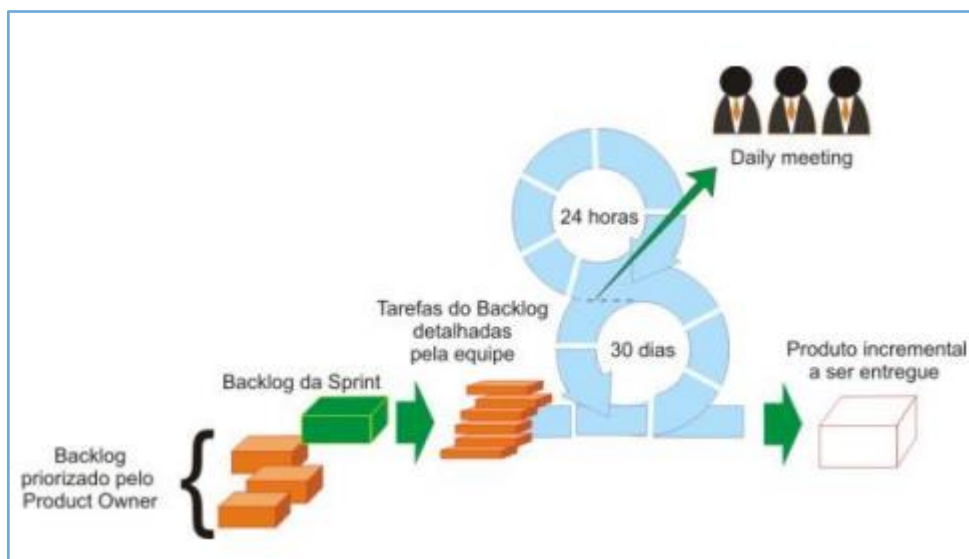


Figura 1. Ciclo do Scrum (adaptado de The Scrum Development Process)  
Fonte: Pereira et. al., 2007

Segundo Schwaber e Sutherland (2013), o Scrum prescreve quatro eventos formais dentro dos limites da Sprint, estes são:

- Reunião de planejamento da Sprint;
- Reunião diária;
- Reunião de revisão da Sprint;
- Retrospectiva da Sprint.

De acordo com Pereira et. al. (2007) antes de cada Sprint, realiza-se uma Reunião de planejamento (Sprint Planning Meeting), na qual o time de desenvolvimento tem contato com o cliente (*Product Owner*), a fim de priorizar o trabalho que precisa ser feito, selecionar e estimar as tarefas que o time pode desempenhar dentro da Sprint.



Ao longo da Sprint, o time monitora o andamento do desenvolvimento realizando Reuniões Diárias (Daily Meeting), com não mais que 15 minutos de duração, e observa o seu progresso utilizando um gráfico chamado *Sprint Burndown*.

Ao final de cada Sprint, é realizada uma Reunião de Revisão (*Sprint Review*) do produto entregue, verificando se tudo que foi acordado entre o time de desenvolvimento e o *Product Owner* realmente foi implementado. Em seguida, realiza-se a Reunião de Retrospectiva (*Sprint Retrospective*), uma reunião de lições aprendidas, com o objetivo de melhorar o processo, time de desenvolvimento e/ou produto para próxima Sprint (RISING; JANOFF, 2000).

### 3. SAFe (Scaled Agile Framework)

De acordo com Da Silva et al. (2016, p. 2) o SAFe é um framework que propõe boas práticas de desenvolvimento ágil, tendo uma estrutura baseada em práticas e padrões propostos e consolidados em outros frameworks, como: *Lean*, *XP (eXtreme Programming)*, *Scrum* e *Kanban*.

Segundo Turetken et al. (2016, p. 2) o SAFe apresenta diversos benefícios, como aceleração do *time-to-market*, aumento da produtividade e da qualidade, e redução de riscos e custos do projeto.

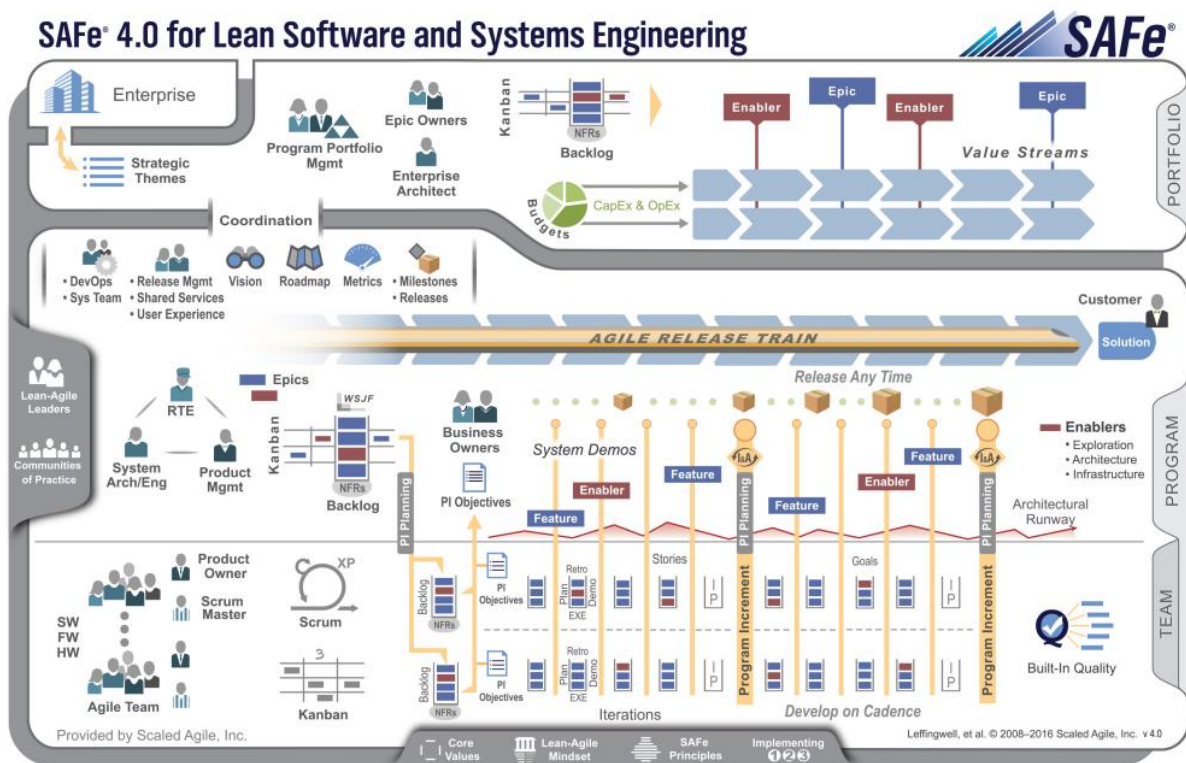


Figura 2. Big Picture para organizações menores (até 100 pessoas)  
Fonte: SAFe (2016b)

SAFe é um framework para atender toda a organização. Na versão atual (4.0) existem 2 modelos. Um modelo é direcionado para organizações menores, com 100 pessoas ou menos, conforme apresentado na Figura 2, é dividido em 3 níveis como nas versões anteriores do SAFe. O outro modelo foi planejado para soluções maiores, incluindo mais de 100 pessoas por fluxo de valor, conforme apresentado na Figura 3, este é dividido em 4 níveis.

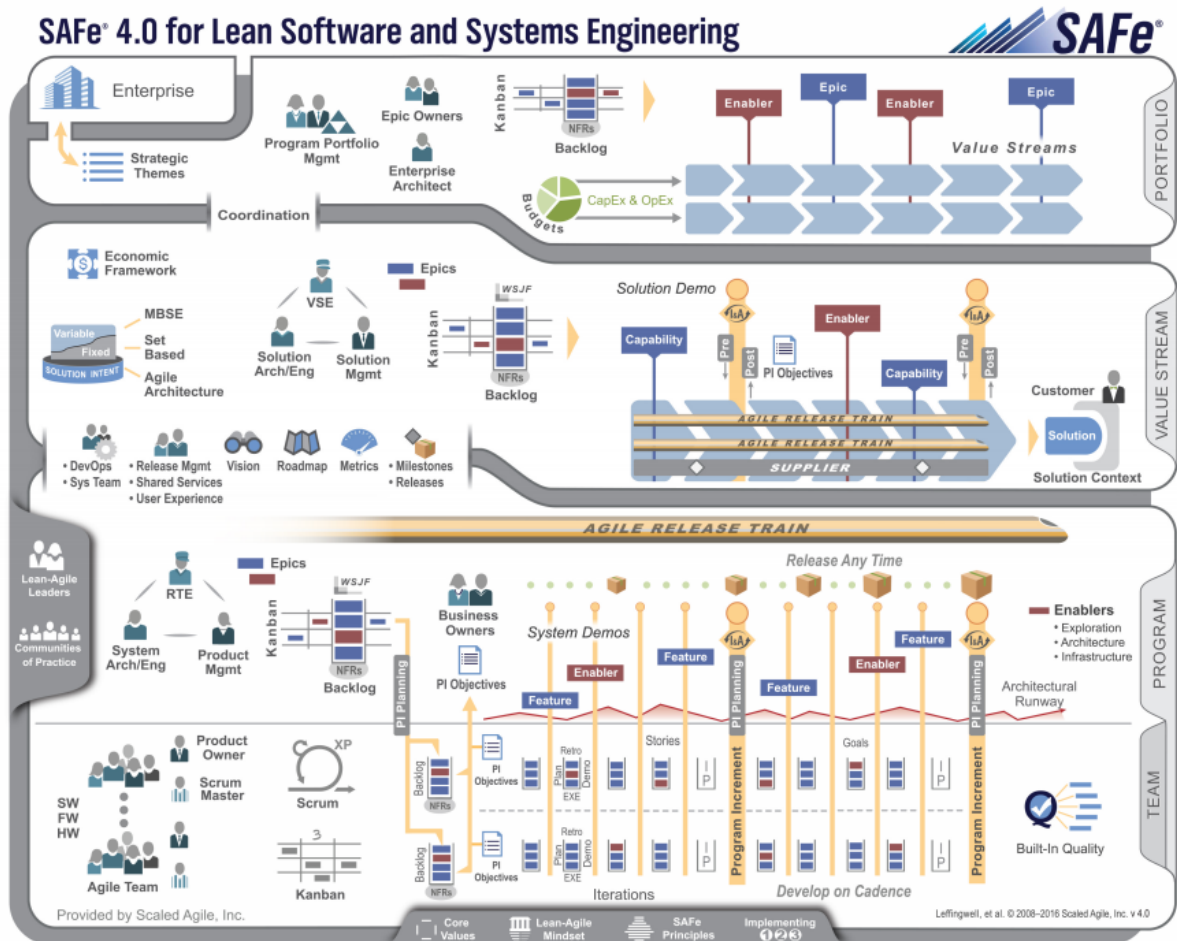


Figura 3. Big Picture para organizações maiores (a partir de 100 pessoas)  
Fonte: SAFe (2016b)

Os 4 níveis existentes no SAFe são, *Team*, *Program*, *Value Stream* e *Portfolio*. Os níveis do SAFe, fornecem abstrações organizacionais com papéis e práticas destinadas a orientar a agilidade em grandes empresas de software.

### 3.1 Team

O nível *Team* fornece modelos organizacionais, artefatos, papéis e processos para as atividades das equipes ágeis. Conforme visto na figura 2 e 3, o nível *Team* trabalha muito parecido com o Scrum.

Da Silva et al. (2016, p. 3) afirmam que no nível *Team* são levantadas as histórias de usuários, que representam as funcionalidades do sistema, e são definidos os critérios de aceitação das histórias, que são as necessidades que a funcionalidade deve atender para que a história seja aceita pelo cliente.

Nesse nível estão grupos multifuncionais de indivíduos que trabalham juntos, os quais tem a capacidade e autoridade para definir, construir e testar soluções de valor numa série de iterações de tamanho fixo (TURETKEN et al., 2016, p. 4).





No nível *Team*, realiza-se o planejamento, execução e retrospectiva de uma iteração, onde as histórias de usuário serão implementadas, podendo variar de uma a duas semanas. Essa iteração é chamada de Sprint (DA SILVA ET AL., 2016, p. 3).

O conteúdo elegível de cada Sprint é de responsabilidade do PO (*Product Owner*), que é o encarregado de priorizar, classificar e refinar o *backlog* do time, organizando a lista das histórias de usuário (GUPTA e tal., 2016, p. 5362).

Assim como no Scrum, no SAFe o time planeja e executa uma iteração com tamanho fixo, de acordo com os objetivos da iteração. Cada iteração prove um incremento valioso de uma nova funcionalidade, conquistada através de um padrão que se repete. Planejar a iteração, comprometer-se com alguma funcionalidade, executar a iteração construindo e testando histórias, demonstrar a nova funcionalidade, realizar uma retrospectiva, e repetir o mesmo processo para a próxima iteração, esse é o padrão iterativo e incremental (SAFE, 2016c).

É importante lembrar que num time ágil, além dos papéis de time e do PO, existe um papel fundamental chamado de Scrum Master. O Scrum Master tem como principais objetivos ajudar o time ágil a se auto organizar e auto gerenciar, que de acordo com Lopez (2015, p. 3) ele faz isso ensinando e treinando o time ágil sobre os princípios e práticas da metodologia Scrum.

### 3.2 Program

O principal objetivo no nível *Program* é organizar as equipes ágeis em grande escala, a fim de otimizar a entrega de valor dos requisitos. Além disso, o nível *Program* também alinha as equipes com uma visão estratégica e um *roadmap* para cada investimento (TURETKEN, 2016, p. 5).

No nível *Program*, times, funções/papéis e atividades são organizadas em torno da metáfora ART (*Agile Release Train*), um time de times ágeis que proporciona um fluxo contínuo de lançamento de incrementos de valor. ARTs são organizações virtuais formadas para abranger limites funcionais, eliminar *handoffs* e passos desnecessários, e acelerar a entrega de valor através da implementação do SAFe e princípios e práticas do *Lean-Agile* (SAFE, 2016d).

A metáfora do trem vem da ideia que um trem parte de uma estação e chega na próxima estação com um horário previsível. A metáfora comunica conceitos como, confiabilidade no cronograma, velocidade padrão, planejamento previsível e etc. Vemos esses conceitos no ART pois ele é constante e relativamente pequeno.

No SAFe o valor é entregue por ARTs de longa duração, cada um dos quais realiza uma porção do fluxo de valor, ou até mesmo todo o fluxo de valor. São entregues incrementos de valor nos PIs (*Program Increments*) de 8 a 12 semanas de duração. Cada PI é formado de múltiplas iterações, onde cada iteração desenvolve um incremento, significativo e valioso do sistema desenvolvido. Cada ART é composto de 5 a 12 times ágeis (50 – 125+ pessoas) e inclui todos os papéis e infraestrutura necessária para entregar software totalmente testado e funcionando (SAFE, 2016d).

Os ARTs são auto organizados e auto gerenciados pelos times ágeis, mas é necessário alguém para criar o ART e dirigir o mesmo, ou seja, alinhar os diversos times para uma missão em comum, este papel é realizado pelo RTE (*Release Train Engineer*), o Scrum Master do trem, que garante que o trem ande suavemente e dentro dos trilhos (SAFE, 2016d).

Cada PI começa com uma *Planning Meeting*, onde todos os passageiros do trem, ou membros dos times se juntam para verificar a visão ou *Road Map* do trem e os objetivos de cada PI. Cada time informa o que pode fazer nesse PI e comunica os outros times do trem. Desta



forma sabem o que esperar de cada time, cada ART, e o fluxo de valor como um todo. Além disso identificam dependências entre os times e os riscos que terão que lidar (SAFE, 2016e).

Lean, Agile e SAFe, todos dedicam foco a entrega contínua de valor ao cliente. Até o fim de cada PI, os times estão envolvidos no desenvolvimento das *features*. Cada PI conta, então PI depois de PI, os times estão focados na entrega. Com isso, os times perdem o foco em outra coisa, a inovação. Para resolver isso, o SAFe apresenta as IP (*Innovation and Planning Iterations*), que faz parte da camada *Team*, esta iteração é dedicada a inovação e planejamento. No momento inovação, é dado tempo para os times se engajarem em *hackathon* e explorarem ideias criativas, já no momento de planejamento, é investido tempo para fazer ajustar o trem, como, demonstração das conquistas da PI, manutenção do trem por retrospectiva, e planejar a próxima PI juntos (SAFE, 2016f).

### 3.3 Value Stream

O nível *Value Stream* é um nível opcional do SAFe, que se destina a construção de soluções grandes e complexas, soluções que normalmente requerem múltiplos ARTs. O nível fornece uma série de constructos que são novos no SAFe 4.0. Ele foi projetado em grande parte para empresas que enfrentam os maiores desafios de sistemas, que envolvem habilidades multidisciplinares, como software, hardware, elétrica, eletrônica, mecânica, e muitos outros. A construção desses sistemas, conta muitas vezes com centenas, talvez milhares, de profissionais, bem como fornecedores, internos e externos. O fracasso de um sistema desses ou até mesmo de um subsistema tem consequências econômicas e sociais inaceitáveis (SAFE, 2016g).

O principal objetivo desse nível é descrever as abordagens *Lean* e ágeis para desenvolvimento de sistemas que escalam para o desafio de definir, construir e implantar sistemas de missão crítica (SAFE, 2016g).

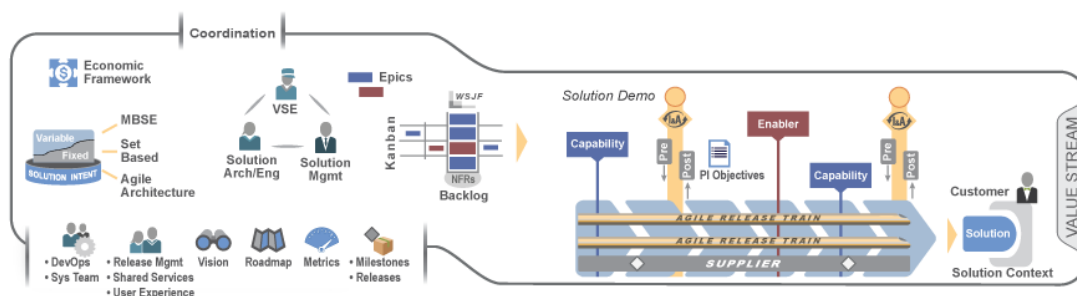


Figura 4. Value Stream level

Fonte: SAFe (2016a)

Na construção de sistemas com tamanho desafio, se faz necessário novos constructos, artefatos e coordenação. Um destes é o framework econômico, que vem com o objetivo de fornecer limites financeiros para o fluxo de valor na tomada de decisões (SAFE, 2016g).

Segundo SAFe (2016g), esse nível é similar ao nível *Program*, pois é organizado em torno de PIs, o qual é sincronizado com todos os ARTs no fluxo de valor.

De acordo com SAFe (2016g), esse nível utiliza algumas práticas como:

- *MBSE (Model-Based System Engineering)* - descreve como requisitos e design podem ser desenvolvidos, documentados e mantidos em modelos mais flexíveis e acessíveis;
- *Set-Based Design* – prática que objetiva adiar ao máximo a tomada de cada decisão, de maneira que seja feita apenas quando seria irresponsável protelar ainda mais, e,



- *Agile Architecture* - apoia o ato de equilíbrio entre o design emergente, aquele que é feito na hora pelas equipes ágeis, e a arquitetura intencional, que é criado de forma colaborativa com os líderes de alto nível técnico e as equipes.

A solução é o principal constructo desse nível, sendo explorado a intenção e contexto da solução. Outra função do nível de *Value Stream* é coordenação de múltiplos ARTs, para que todos entreguem a solução juntos. Para garantir isso, todos os ARTs estão sincronizados em um único PI, com iterações sincronizadas para facilitar a colaboração entre os ARTs e até mesmo entre os fornecedores. No início de cada PI, o planejamento é feito com todos os ARTs ao mesmo tempo. No fim de cada PI, uma demonstração da solução é realizada, integrada de todos ARTs. Após isso é feito uma inspeção e adaptação que é realizada para melhorar o processo de toda a cadeia de valor (SAFE, 2016g).

### 3.4 Portfolio

O nível mais alto do SAFe é o nível de portfólio, onde os programas estão alinhados à estratégia de negócios da empresa ao longo de linhas de fluxo de valor. Este nível é necessário para as empresas, que necessitam de modelos de governança e de gestão (TURETKEN, 2016, p. 5).

Em empresas maiores, pode haver vários *Portfolios*, um para cada instância do SAFe, cada um com sua própria estratégia e recursos. Nesse nível o *Lean Thinking* é essencial para a ampliação das práticas ágeis da empresa. Para organizar as tarefas nesse nível, é utilizado o sistema Kanban que faz o trabalho visível para todos os interessados e prove um limite de WIP (*Work-In-Progress*) (SAFE, 2016h).

SAFe (2016i) afirma que no nível de *Portfolio* o épico no sistema Kanban passa por 6 estágios no caminho da implementação (ou rejeição) conforme figura 5.

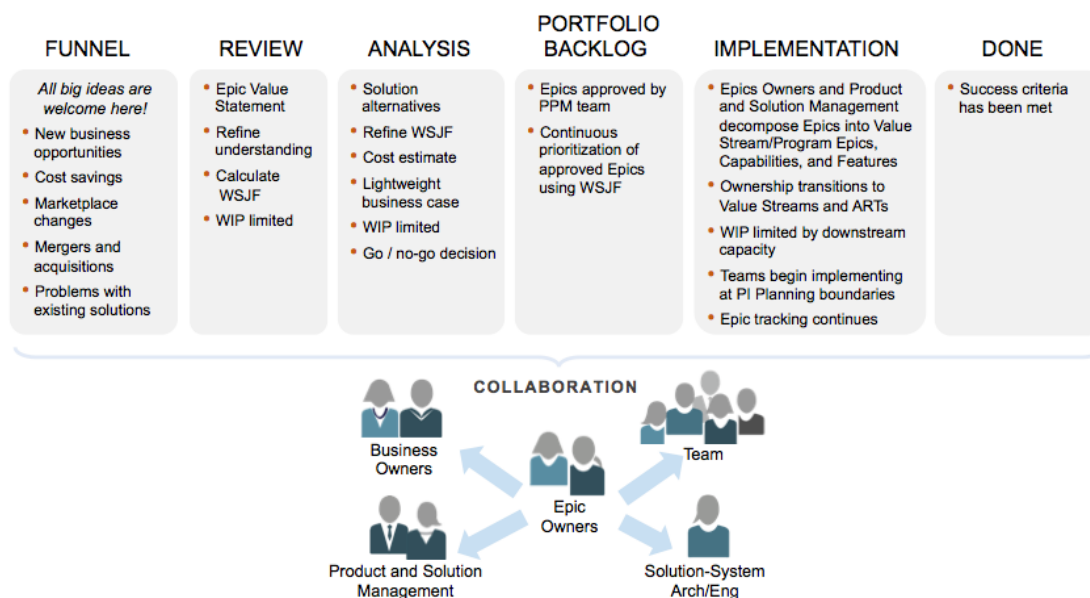


Figura 5. Kanban  
Fonte: SAFe (2016i)

1. *Funnel* - onde todas as grandes ideias são bem-vindas, são vistas novas oportunidades de negócio, mudanças de mercado, aquisições;
2. *Review* - refina o entendimento das ideias, é calculado o esforço e custo;



3. *Analysis* - avalia soluções alternativas, estima custos verificando a viabilidade e benefícios;
4. *Portfolio Backlog* - armazena os épicos aprovados até haver capacidade disponível;
5. *Implementing* - quando a capacidade se torna disponível, onde a implementação inicia;
6. *Done* - quando o épico alcança seus critérios de sucesso.

Esse nível também é responsável por levantar indicadores, tais como KPIs (*Key Performance Indicators*), a satisfação do funcionário, do cliente, com a qualidade dos lançamentos de software e do número de lançamentos por ano (SAFE, 2016h).

Outro ponto muito importante e discutido nesse nível é o orçamento. É normal vermos empresas tradicionais de desenvolvimento de software terem problemas para conciliar a gestão orçamentaria e uma cultura ágil de desenvolvimento de software. De acordo com SAFe (2016j), o framework fornece estratégias para resolver esses problemas, provendo para as empresas o melhor dos dois mundos, um processo de desenvolvimento que é muito mais sensível às necessidades do mercado, juntamente com gestão profissional e responsável dos gastos de investimento.

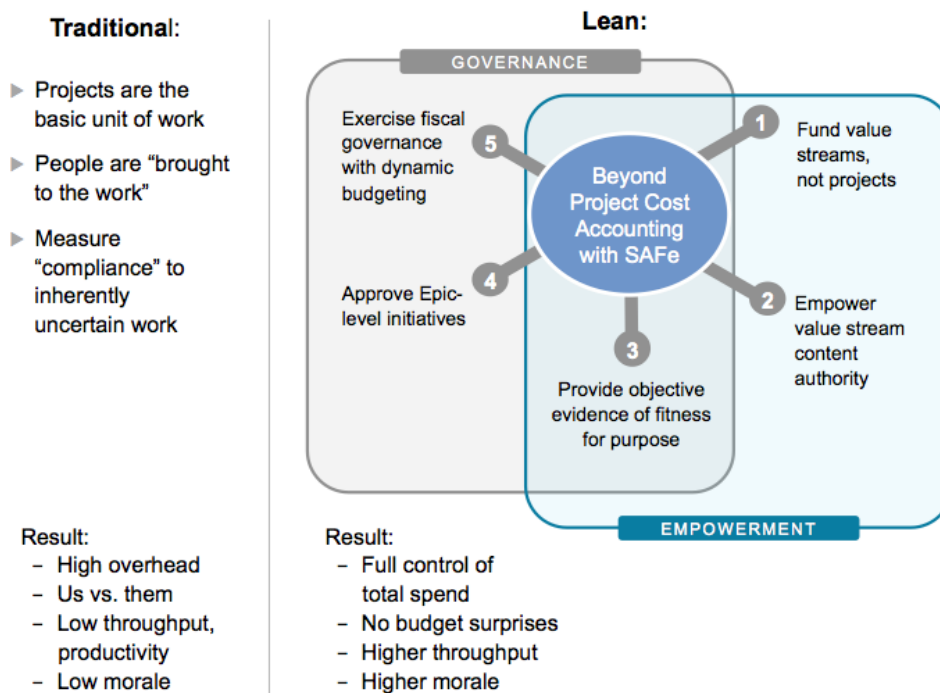


Figura 6. Budget  
Fonte: SAFe (2016j)

SAFe (2016j) afirma que claramente, a gestão orçamentaria tradicional é um grande inibidor para práticas *Lean* e ágeis. Para resolver isso, SAFe fornece cinco práticas integradas conforme figura 6, as quais são detalhadas abaixo:

1. O primeiro passo é aumentar a autonomia e diminuir a sobrecarga movendo as decisões de gastos do dia-a-dia para mais perto da solução. Isso é feito através de orçamentos para cada fluxo de valor. Assim os *stakeholders* tem autonomia para alocar orçamento onde faz sentido para o contexto atual do *backlog* e *roadmap*. Além disso, como os fluxos de valor são de longa duração, as pessoas envolvidas trabalham juntos por um longo período, aumentando o espírito de equipe, conhecimento, competência e produtividade. Mesmo assim o orçamento



ainda é controlado. Na maioria dos casos, as despesas são fixadas através de um PI onde fica fácil de prever, assim todas as partes interessadas sabem o gasto previsto para o próximo período, independentemente dos recursos.

2. A empresa ainda deve ter certeza de que os fluxos de valor estão construindo a coisa certa. Essa é uma das razões pelas quais projetos foram criados, entretanto no SAFe isso não é feita com projetos, mas por meio do empoderamento e responsabilidades da gestão de produtos e soluções. Para dar visibilidade a todo mundo, todos os trabalhos a serem feitos, serão feitos, contidos e priorizados no *backlog* do produto e da solução.

3. Outro ponto importante é fornecer evidências objetivas do propósito do orçamento daquele fluxo de valor. É possível ter um rápido *feedback* do investimento através da demonstração da solução, ou até mesmo a cada iteração através da demonstração do sistema, certificando os principais interessados que está sendo feito a coisa certa do jeito certo.

4. Épicos devem passar por aprovação orçamentária adicional, pois são grandes e impactantes o suficiente para afetar outros fluxos de valor e ARTs, bem como podem ter um elevado custo. Estes podem ser financiados por reserva orçamentária ou até mesmo feito através de realocação de pessoal ou orçamento de outro fluxo de valor, mas essa decisão exige análise e tomada de decisão em nível estratégico.

5. Por fim, deve-se exercitar a governança fiscal com orçamento dinâmico. Os fluxos de valor são em grande parte auto organizados e auto gerenciados, mas eles não causam a sua própria existência e nem se financiam. Para esse fim, a gestão de programa e portfolio tem autoridade para definir e ajustar os orçamentos do fluxo de valor dentro do portfólio. A fim de responder à mudança, no entanto, o financiamento irá variar ao longo do tempo com base na dinâmica do negócio.

#### 4. Conclusões

Este trabalho está focado na área de gerência de projetos ágeis, mais precisamente nas metodologias e no framework SAFe. Por meio de revisões sistemáticas da literatura, pôde-se perceber que a área está sendo bastante pesquisada, vemos isso em obras como "Scrum - a arte de fazer o sobro de trabalho na metade do tempo" de Jeff Sutherland, "The lean Startup" de Eric Ries e "Scaled Agile Framework (SAFe) Distilled: A Practical Guide to Scaling Agile in the Enterprise" de Richard Knaster e Dean Leffingwell.

Para obter uma análise mais adequada da gerência de projetos ágeis, deve-se levar em consideração o contexto atual do desenvolvimento de software e a aplicação de metodologias ágeis nas empresas.

É importante ter em mente que o SAFe é um *template* e cada organização pode customizar de acordo com a sua realidade e para cada situação. Verifica-se que quando uma organização cresce e os times crescem é o momento de pensar em escalar a estrutura ágil, e o SAFe provê um caminho claro de como começar a fazer isso. O SAFe tem vários benefícios, como a comunicação entre diversas camadas, entregas incrementais, a visão do portfólio alinhada com o desenvolvimento, a aproximação das pessoas e divisão das responsabilidades através do *Agile Team*, etc. Um diferencial do SAFe é que ele traz a camada de *Portfolio*, ou seja, a camada de tomada de decisões, diferente dos outros frameworks que se preocupam apenas com o desenvolvimento.

O desenvolvimento desse artigo, possibilitou a identificação de alguns pontos de sugestão para trabalhos futuros. Como o artigo trata do *framework* SAFe, seria interessante abordar formas de implantação do mesmo em organizações. Ainda sobre o SAFe, seria adequado desenvolver estudo comparativo entre ele e outros frameworks como por exemplo o LeSS (Large Scale Scrum) ou o Nexus.



## Referências

BALLE, A. **Análise de metodologias ágeis: conceitos, aplicações e relatos sobre XP e scrum**. Rio Grande do Sul, 2011. Trabalho de Graduação de Ciências da Computação - Universidade Federal do Rio Grande do Sul.

BARBOSA, Vladimir. **Comparação entre Metodologias Ágeis e Tradicionais para o Desenvolvimento de Software**. Unipac - Universidade Presidente Antônio Carlos, Faculdade de Tecnologia e Ciências de Conselheiro Lafaiete, 2010.

BECK, K. Embracing change with extreme programming. **IEEE Computer Society**, v. 32, 70–77, 1999.

BOEHM, B., **A View of 20th and 21st Century Software Engineering**. In: Proceedings of the 28th international conference on Software engineering. ACM, 2006. p. 12-29.

BOEHM, B. Get ready for agile methods, with care. **IEEE Computer Society**, v. 35, n. 1, p. 64-69, 2002.

COAD, Peter; DE LUCA, J.; LEFEBRE, E. **Java modelling in color**. Prentice-Hall, Englewood Cliffs, NJ, 1999.

DA SILVA, Dyego A., DE OLIVEIRA, Edgard C., CANEDO, Edna D., MARTINS, Hugo F. Application of a hybrid process software requirements management. **Iberian Conference on Information Systems and Technologies**, CISTI, 2016-July, art. no. 7521442, 2016.

GUPTA, R.K., REDDY, P.M. Adapting agile in a globally distributed software development **Proceedings of the Annual Hawaii International Conference on System Sciences**. art. no. 7427851, pp. 5360-5367, 2016.

HIGHSMITH, Jim. **Agile Project Management: Creating Innovative Products**. 2004.

IANSITI, Marco; MACCORMACK, Alan. Developing products on Internet time. **Harvard business review**, v. 75, n. 5, p. 108-117, 1997.

LEE, Gwanhoo; XIA, Weidong. The ability of information systems development project teams to respond to business and technology changes: a study of flexibility measures. **European Journal of Information Systems**, v. 14, n. 1, p. 75-92, 2005.

LIB - Lean Institute Brasil. 2015. Disponível em <<http://www.lean.org.br/o-que-e-lean.aspx>>. Acesso em: 28 ago. 2016.

MARUPING, Likoeb M.; VENKATESH, Viswanath; AGARWAL, Ritu. A control theory perspective on agile methodology use and changing user requirements. **Information Systems Research**, v. 20, n. 3, p. 377-399, 2009.



MOREIRA, Francisco; ALVES, Anabela C.; SOUSA, Rui M. Towards eco-efficient lean production systems. In: **Balanced Automation Systems for Future Manufacturing Networks**. Springer Berlin Heidelberg, 2010. p. 100-108.

OHNO, Taiichi. **The Toyota Production System: Beyond Large-Scale Production**. Productivity Press, Portland, 1988.

PAULK, Mark C. Agile methodologies and process discipline. **Institute for Software Research**, p. 3, 2002.

PEREIRA, P.; TORREÃO, P.; MARÇAL, A.S. **Entendendo o Scrum para gerenciar projetos de forma ágil**. Mundo PM-Project Management. Curitiba, ed. 14, abril. 2007.

POPPENDIECK, Mary; POPPENDIECK, Tom. **Implementing lean software development: From concept to cash**. Pearson Education, 2007.

RISING, Linda; JANOFF, Norman S. The Scrum software development process for small teams. **IEEE software**, v. 17, n. 4, p. 26, 2000.

SAFe - Scaled Agile Framework. Site do framework. 2016. Disponível em <<http://www.scaledagileframework.com/whats-new-in-safe-40/>>. Acesso em: 27 ago. 2016a.

SAFe - Scaled Agile Framework. Site do framework. 2016. Disponível em <<http://www.scaledagileframework.com/posters/>>. Acesso em: 27 ago. 2016b.

SAFe - Scaled Agile Framework. Site do framework. 2016. Disponível em <<http://www.scaledagileframework.com/team-level/>>. Acesso em: 27 ago. 2016c.

SAFe - Scaled Agile Framework. Site do framework. 2016. Disponível em <<http://www.scaledagileframework.com/program-level/>>. Acesso em: 27 ago. 2016d.

SAFe - Scaled Agile Framework. Site do framework. 2016. Disponível em <<http://www.scaledagileframework.com/pi-objectives/>>. Acesso em: 27 ago. 2016e.

SAFe - Scaled Agile Framework. Site do framework. 2016. Disponível em <<http://www.scaledagileframework.com/innovation-and-planning-iteration/>>. Acesso em: 27 ago. 2016f.

SAFe - Scaled Agile Framework. Site do framework. 2016. Disponível em <<http://www.scaledagileframework.com/value-stream-level/>>. Acesso em: 27 ago. 2016g.

SAFe - Scaled Agile Framework. Site do framework. 2016. Disponível em <<http://www.scaledagileframework.com/portfolio-level/>>. Acesso em: 27 ago. 2016h.

SAFe - Scaled Agile Framework. Site do framework. 2016. Disponível em <<http://www.scaledagileframework.com/portfolio-kanban/>>. Acesso em: 27 ago. 2016i.

SAFe - Scaled Agile Framework. Site do framework. 2016. Disponível em <<http://www.scaledagileframework.com/budgets/>>. Acesso em: 27 ago. 2016j.



**V SINGEP**

**Simpósio Internacional de Gestão de Projetos, Inovação e Sustentabilidade**  
**International Symposium on Project Management, Innovation and Sustainability**

ISSN: 2317 - 8302

SCHWABER, Ken. Scrum development process. In: **Business Object Design and Implementation**. Springer London, 1997. p. 117-134.

SCHWABER, Ken; SUTHERLAND, Jeff. The Scrum Guide. The Definitive Guide to Scrum: The Rules of the Game. 2013. Disponível em <<http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-US.pdf#zoom=100>>. Acesso em: 31/08/2016

SILVA, E. L. da; MENEZES, E. M. Metodologia da pesquisa e elaboração de dissertação. 3. ed. rev. atual. Florianópolis: Laboratório de Ensino a Distância da UFSC, 2001.

TURETKEN, O., STOJANOV, I., TRIENEKENS, J.J.M. Assessing the adoption level of scaled agile development: a maturity model for Scaled Agile Framework. **Journal of Software: Evolution and Process**, 2016

VARGAS, Ricardo. **Gerenciamento de Projetos – Estabelecendo diferencias competitivos**. Rio de Janeiro: Brasport, 2005.

WOMACK, James P.; JONES, Daniel T. **Lean Thinking: Banish Waste and Create Wealth in Your Corporation**. Simon & Schuster, New York, USA, 1996.

WOMACK, James P.; JONES, Daniel T. Lean consumption. **Harvard business review**, v. 83, n. 3, p. 58-68, 2005.