



**Universidade de Brasília - UnB**

**Faculdade Gama - FGA**

**Engenharia de Software**

**Disciplina: Verificação e validação de software**

**Professor: Ricardo Ajax Dias Kosloski**

**Análise de um processo de Engenharia de Requisitos de software utilizando o  
Framework SAFe.**

Alexandre Torres 13/0099767

Anna Larissa 12/0049465

Daniel Moura 14/0037578

Edson Gomes 09/0037618

Eduardo Gomes 14/0137068

Gesiel Freitas 14/0082531

Hugo Martins 13/0028100

Iolane Caroline 13/0028355

Jhonatan Alves 13/0116408

Kássia Catarine 17/0052087

**Brasília, 24 de maio de 2017**

---

## SUMÁRIO

## 1. INTRODUÇÃO

A cada dia os *software* se tornam cada vez maiores e mais complexos, tendo em vista esse fato, tornou-se necessário a busca por métodos que procurem garantir a qualidade do produto. Os trabalhos de verificação e validação de *software* tem como objetivo a busca por essa qualidade. Os problemas encontrados por essa área de conhecimento se classificam por erros, defeitos, falhas, omissões, fatos incorretos, inconsistências, ambiguidades e informações estranhas. Onde, para encontrar tais problemas, são utilizadas técnicas de verificação e validação.

O presente projeto visa utilizar uma técnica estática de verificação chamada Inspeção de *Software*. Bernard e Price (1994, apud Alves e Gouvea, 2013) afirmam que a técnica de inspeção é eficiente e pode ser utilizadas nas demais fases do processo de desenvolvimento. A inspeção também reduz os custos nas reparações de defeitos e sua eficiência atinge números de 30 a 70% de satisfação.

Kelly (1992, apud Alves e Gouvea 2013) citam que o custo e o esforço empregados na correção dos defeitos utilizando a técnica de inspeção são baixos. “Pesquisas indicam que em média 1,75 horas são utilizadas na correção de um defeito encontrado na inspeção de *software*, 1,46 horas em um defeito encontrado na inspeção de código e 17 horas em um defeito encontrado nos testes”.

O processo de inspeção inclui seis fases que são: planejamento, detecção de defeitos, coleção de defeitos, discriminação de defeitos, retrabalho e continuação. Nessas fases são definidos os aspectos a serem avaliados e os questionários, tal fase é decisiva para garantir qual o nível de cobertura é desejado atingir.

Dessa forma, para o presente projeto foram definidos alguns objetivos de verificação que auxiliarão na análise de um projeto de Elicitação de Requisitos produzido na disciplina de

Requisitos da Universidade de Brasília campus Gama, onde foi utilizado o *Scaled Agile Framework* (SAFe).

## 2. OBJETIVO

Este trabalho tem como objetivo utilizar as técnicas de verificação estática de inspeções, para aferir um processo de elicitação de requisitos desenvolvido em um projeto na disciplina de Requisitos de Software utilizando o SAFe.

Para isso, como objetivo específico, será necessário analisar se existem falhas no processo, analisar o processo de acordo com a técnica de inspeções e propor melhorias ao processo.

## 3. CRONOGRAMA

Para a execução do trabalho, foi criado um cronograma para guiar as datas das atividades, que pode ser visto na imagem a seguir:

☐ Ponto de Controle 1	7.33d?	22/04/2017	24/05/2017	
☐ Definição do escopo	0.33d?	22/04/2017	24/04/2017	
Definir escopo de Verificação e Validação	0.33d?	22/04/2017	22/04/2017	
Definir Técnica de V & V	0.33d?	22/04/2017	22/04/2017	
Planejar Apresentação 1	0.33d?	23/04/2017	23/04/2017	3,4
Apresentação 1	1d?	23/04/2017	24/04/2017	5
Definir ferramenta de V & V	3d?	19/05/2017	22/05/2017	
Elaborar Processo de V & V	5d?	19/05/2017	24/05/2017	4
Ponto de Controle 1	0.33d?	24/05/2017	24/05/2017	2,7,8

Imagem x: Cronograma do ponto de controle 1.

☐ <b>Ponto de Controle 2</b>	5.67d?	01/06/2017	26/06/2017	1
☐ <b>Planejar inspeção</b>	1.33d?	01/06/2017	07/06/2017	2
Avaliar andamento do projeto	0.33d?	01/06/2017	01/06/2017	
Planejar Objetivos de V & V	0.33d?	01/06/2017	01/06/2017	4
Elaborar Questionário	3d?	02/06/2017	05/06/2017	13
Elaborar Checklist	5d?	02/06/2017	07/06/2017	13
☐ <b>Executar inspeção</b>	1d?	08/06/2017	13/06/2017	11
Avaliar andamento do projeto	0.33d?	08/06/2017	08/06/2017	
Realizar reunião Overview Meeting	0.33d?	08/06/2017	08/06/2017	11
Examinar artefatos de trabalho	4d?	08/06/2017	12/06/2017	11
Planejar Reunião de inspeção	0.33d?	12/06/2017	12/06/2017	
☐ <b>Realizar reunião de inspeção</b>	0d?	13/06/2017	13/06/2017	18
Elaborar Laudo da reunião de inspeção	0.33d?	13/06/2017	13/06/2017	
Elaborar documento de melhorias	0.33d?	13/06/2017	13/06/2017	
☐ <b>Registrar resultados de inspeção</b>	0.33d?	14/06/2017	15/06/2017	16
Avaliar andamento do projeto	0.33d?	14/06/2017	14/06/2017	
Elaborar documento com o detalhamento dos dados deriv	0.33d?	15/06/2017	15/06/2017	22
☐ <b>Avaliar resultados de inspeção</b>	1.33d?	16/06/2017	22/06/2017	24
Avaliar andamento do projeto	0.33d?	16/06/2017	16/06/2017	
Analisar consistência dos relatórios	0.33d?	18/06/2017	18/06/2017	26
Relatar avaliação dos resultados	0.33d?	20/06/2017	20/06/2017	29
Sugerir melhorias	0.33d?	22/06/2017	22/06/2017	30
Preparar apresentação do Ponto de Controle 2	0.33d?	25/06/2017	25/06/2017	31
Ponto de Controle 2	0.33d?	26/06/2017	26/06/2017	32

Imagem x: Cronograma do Ponto de Controle 2.

## 4. FERRAMENTA

Para o auxílio da execução das tarefas do trabalho, a utilização de uma ferramenta de checklist é essencial para automatizar a verificação dos projetos escolhidos, assim a ferramenta Asana foi escolhida para o armazenamento e criação dos *checklist* utilizados no projeto.

O Asana é uma ferramentas desenvolvida pelos criadores [Dustin Moskovitz](#), [Justin Rosenstein](#), ela foi lançada em 2008, e o seu intuito é disponibilizar para o usuário uma plataforma para criação de notas e listas, além de poder acrescentar fotos, áudios, arquivos, realizar comentários nas notas, criar uma baseline para projetos e trabalhar de forma colaborativa.

A escolha da ferramenta foi baseada na necessidade de disponibilidade em qualquer sistema operacional, necessidade de colaboração entre usuários e a possibilidade de criação de checklist especiais para cada projeto analisado.

Ao analisar possíveis ferramentas de *checklist* como *Evernote*, *Trello*, *Google keep* e entre outras, o Asana foi o que mais se destacou por causa da sua gama de funcionalidades disponíveis de forma gratuita e pela facilidade de manuseio.

## **5. MÉTODOS**

A fim de analisar a qualidade dos processos utilizados na disciplina de Requisitos, baseados no SAFe, será utilizada a técnica de inspeção, guiada por questionários.

O caminho a ser seguido para atingir-se o objetivo do projeto será:

1. Definir processo, grupo de requisitos a ser analisado.
2. Especificar o processo de inspeção.
3. Elaborar o *checklist* de inspeção.
4. Realizar a inspeção.
5. Elaborar o relatório contendo os resultados.
6. Sugerir melhorias para o processo.

### **5.1 Definição do Processo de Requisitos**

Para a execução do projeto de Verificação e Validação (V&V) foi definido um processo de requisitos do primeiro semestre de 2015.

#### **5.1.1 Processo de Requisitos**

O processo adotado segue o template do SAFe *framework* Portfólio SAFe, os quais contém os níveis de Portfólio, Programa e Time, que são os exigidos na disciplina de requisitos e adotados pelos alunos.

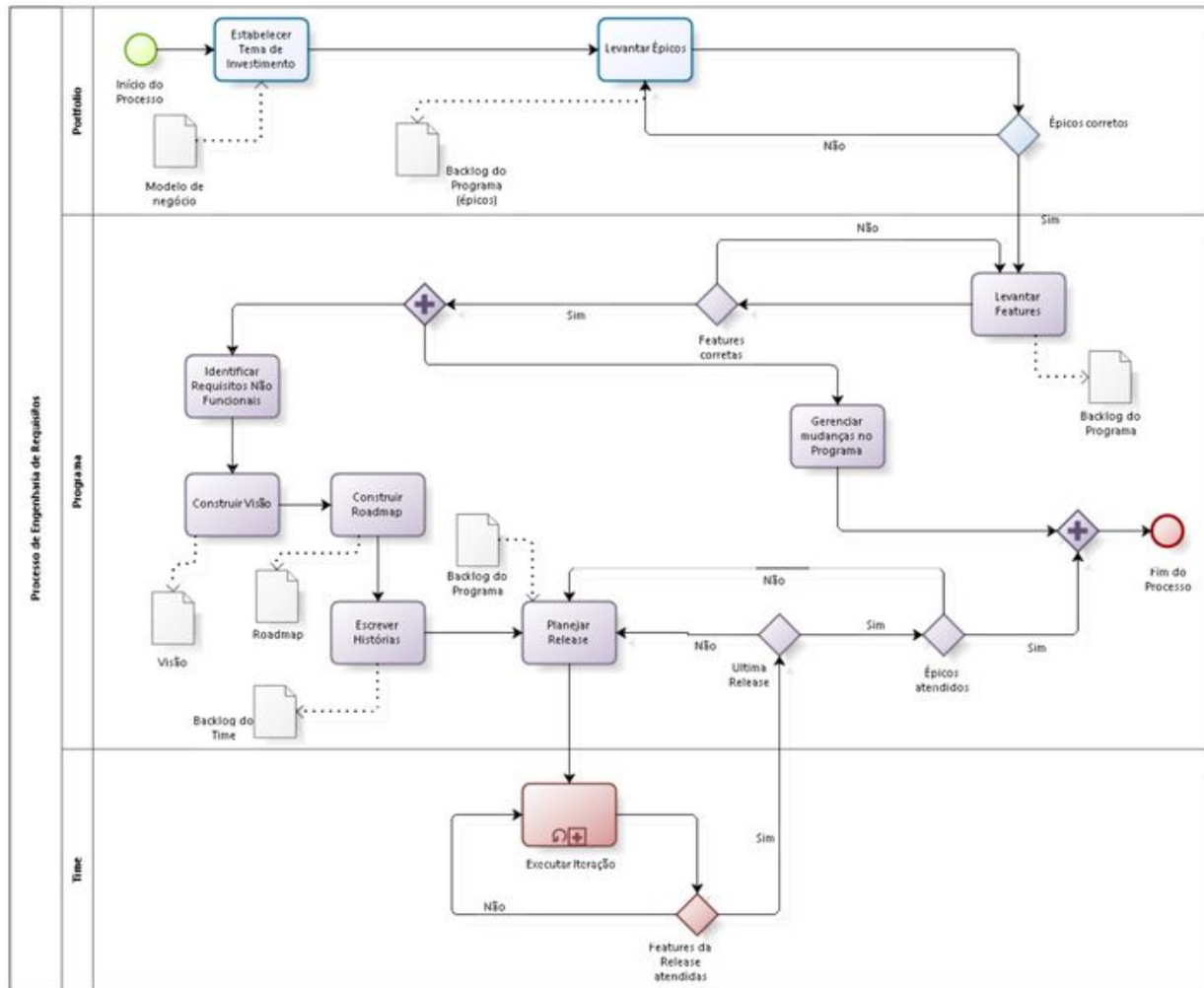


Imagem xx processo de requisitos

Como pode ser visto o nível de time tem um subprocesso executar interação, como pode ser visto abaixo.

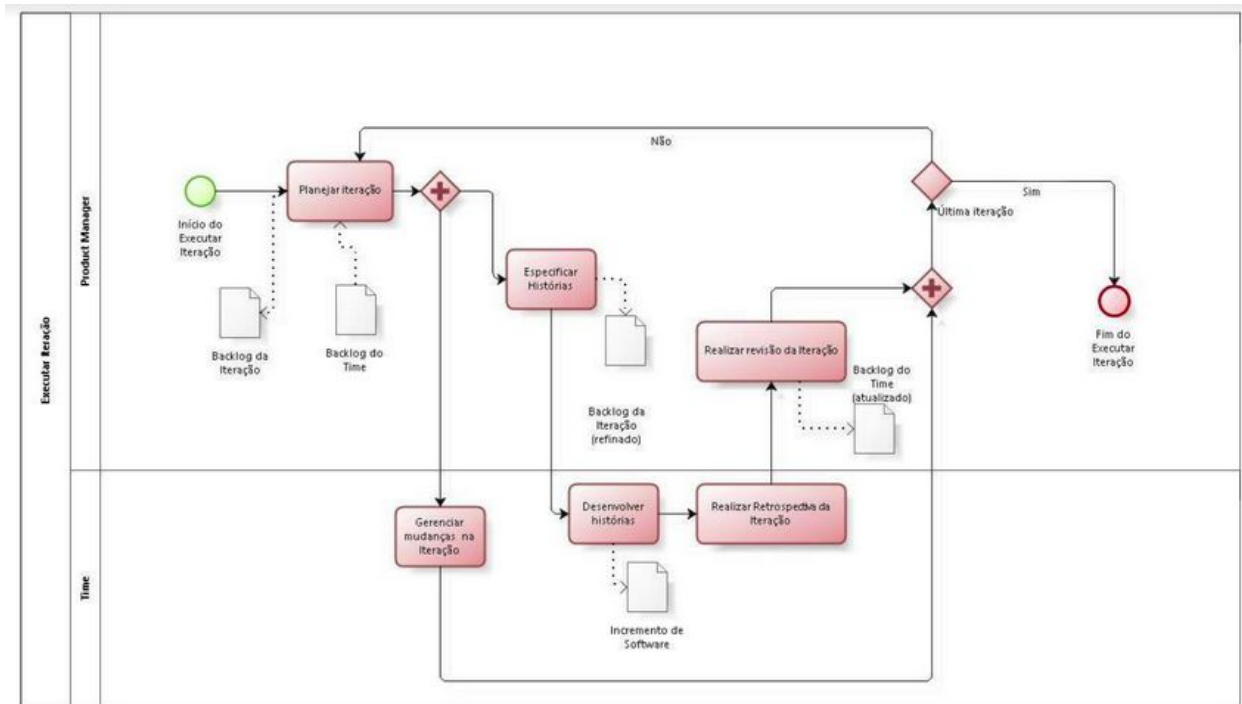


Imagem xx subprocesso executar interação

## 5.2 Especificação do Plano de Inspeção

### 5.2.1 Plano de inspeção

O plano de inspeção é um documento que provê instruções de como a inspeção do produto ocorrerá. Ele provê detalhes sobre quais características devem ser testadas, verificadas e ou validadas a fim de garantir a qualidade do produto. Também especifica métricas e os indicadores a serem utilizados a fim de garantir a conformidade do produto inspecionado com os padrões definidos.

Portanto, percebe-se que neste plano devem ser detalhadas características específicas da operação de inspeção a ser realizada, assim como características relevantes do produto que será inspecionado.



## 5.2.2 Processo

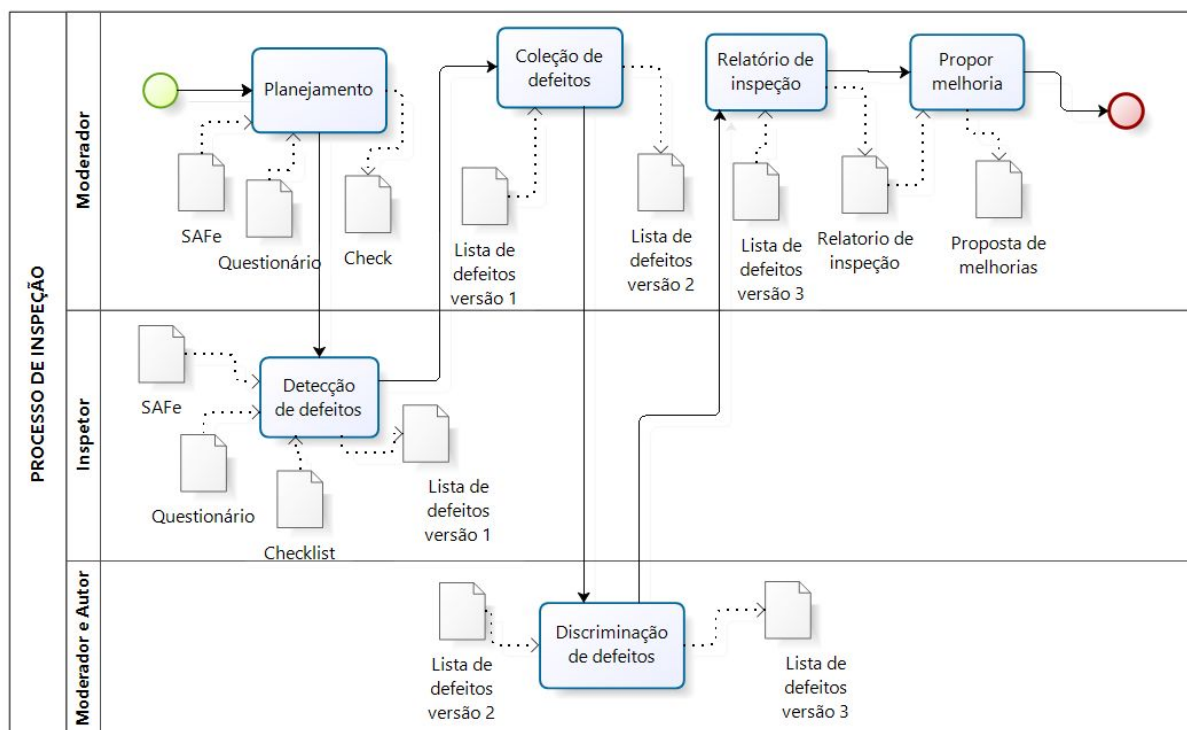


Figura 5.2.2.1: Processo de inspeção

## 5.2.3 Descrição da Atividades

### 5.2.3.1 Planejamento:

No processo de inspeção a ser realizado, utilizará a leitura baseada em *checklist*, onde o inspetor será guiado por questões pré definidas a respeito do processo realizado a partir do *framework* SAFE. Sua utilização é essencial para que o caminho onde a inspeção será feita seja identificado e inspecionado de forma correta, assim atestando a sua qualidade de acordo com os parâmetros utilizados no *checklist*. A avaliação e as perguntas contidas no *checklist* se baseará em campos contido no processo executado pelo portfólio SAFE, e a análise será por meio dos artefatos, dos papéis e das atividades.

**Leitura baseada em checklist:** O *checklist* conterá todos os itens para verificação do processo a fim de atestar a qualidade do processo de requisitos avaliado neste estudo, através dos artefatos, papéis e as atividades constantes nele seguindo o SAFe parâmetro de validação.

**Responsável:** Moderador

#### 5.2.3.2 Detecção de Defeitos

Nesta fase os defeitos encontrados são listados. Os defeitos encontrados podem ser do tipo descrito abaixo:

**Omissão:** A omissão ocorre quando há uma falta de informação ou definições que são relevantes para o projeto;.

**Responsável:** Inspetores

#### 5.2.3.3 Coleção de Defeitos:

Nesta fase são listados todos os defeitos encontrados e os defeitos repetidos são descartados.

**Responsável:** Moderador

#### 5.2.3.4 Discriminação de Defeitos:

Nesta fase as discrepâncias listadas são analisadas junto ao autor de cada artefato para avaliar se a discrepância apontada é um defeito de fato.

**Responsável:** Moderador e autor.

#### 5.2.3.5 Relatório da inspeção:

Nesta fase os dados coletados são descritos e analisados de acordo com as questões de pesquisa definidas no projeto.

**Responsável:** Moderador

#### 5.2.3.6 Propor melhoria:

Nesta fase os dados presentes no relatório de inspeção são utilizadas para propor uma possível melhoria.

---

**Responsável:** Moderador

## **5.4 Checklist**

O *checklist* conterá todos os itens para verificação do processo a fim de atestar a conformidade do processo de requisitos avaliado neste estudo com o SAFe. A análise da conformidade será realizada por meio de avaliações dos artefatos, papéis e as atividades constantes no processo avaliado seguindo o SAFe parâmetro de validação.

### **5.4.1 Itens do SAFe usados para Criar o CheckList**

#### **5.4.1. 1 Itens para checklist Nível de Portfólio**

##### **5.4.1.1.1 Artefatos**

**Epics** - captura e reflete as novas capacidades de negócios que só podem ser fornecidas pela cooperação entre fluxos de valor.

**Enabler Epics** - reflete as iniciativas arquitetônicas e outras técnicas que são necessárias para permitir novos recursos e capacidades .

**Temas Estratégicos** - forneça objetivos empresariais específicos e detalhados que liguem o portfólio à estratégia empresarial em evolução.

**Backlog de portfólio** - é o *backlog* de nível mais alto na SAFe. Ele detém os *Epics Business* e *Enabler* aprovadas necessárias para criar um conjunto de soluções de portfólio que forneça a diferenciação competitiva e / ou as eficiências operacionais necessárias para o sucesso do negócio.

#### 5.4.1.1.2 Papéis

**Gerente de Portfólio** - é uma função que representa os indivíduos com o mais alto nível de tomada de decisão e responsabilidade financeira para um portfólio SAFe. Este grupo é responsável por três áreas principais: estratégia e financiamento de investimentos, orientação do programa *Agile* e governança *Lean*.

**Epic Owners (Dono dos épicos)** - assumir a responsabilidade pela coordenação dos épicos do portfólio através do sistema *Portfolio Kanban*.

**O Enterprise Architect (Arquiteto empresarial)** - funciona em fluxos e programas de valor para ajudar a fornecer uma direção técnica estratégica que possa otimizar os resultados do portfólio. O arquiteto empresarial muitas vezes pode atuar como um proprietário épico para épicos facilitadores.

#### 5.4.1.1.3 Atividades

As atividades fornecem um fluxo de valor para as partes interessadas, uma vez que estes estão em um fluxo definido e cada fluxo é uma série de etapas de desenvolvimento.

### 5.4.1. 2 Itens para checklist Nível de Programa

#### 5.4.1.2.1. Artefatos

**Features** - é um serviço de sistema que atende a necessidade de um *stakeholder*. Cada característica inclui uma hipótese de benefícios e critérios de aceitação, e é dimensionada ou dividida conforme necessário para ser entregue por um único *Agile Release Train* (ART) em um único Incremento de Programa (PI).

**Program Epics** - estes épicos são especificamente para uma ART.

**Program Backlog** - é a área de espera para os próximos recursos e facilitadores, cada um dos quais pode abranger várias ARTs e tem como objetivo avançar a solução e construir sua pista arquitetônica.

**Program Kanban** - gerencia o fluxo de recursos e facilitadores através da ART.

**PI Objectives** - são uma descrição resumida dos objetivos comerciais e técnicos específicos que uma ART pretende alcançar no próximo Incremento de Programa (PI).

**Architectural Runway** - consiste no código existente, componentes e infra-estrutura técnica necessários para suportar a implementação de recursos prioritários e de curto prazo, sem redesenho e atraso excessivos.

#### 5.4.1.2.2 Papéis

**System Architect/Engineer** - é uma equipe individual ou pequena de disciplina cruzada que realmente aplica sistemas de pensamento. Eles definem a arquitetura geral do sistema, ajudam a definir requisitos não funcionais, determinam os principais elementos e subsistemas e ajudam a definir as interfaces e colaborações entre eles.

**Product Management** - a voz interna do cliente e trabalha com proprietários de produtos e clientes para entender e comunicar suas necessidades, definir recursos do sistema e participar da validação. Eles são responsáveis pelo atraso do programa.

**Release Train Engineer (RTE)** - É um líder servo e o chefe *Scrum Master* para ART. O RTE facilita otimizar o fluxo de valor através do programa usando vários mecanismos, como o Programa Kanban, *Inspect & Adapt workshop*, *PI Planning* e mais.

**Business Owners** - são um pequeno grupo de partes interessadas que têm o principal negócio e responsabilidade técnica de aptidão para uso, governança e retorno do investimento para uma solução desenvolvida por um (ART). Eles são os principais interessados na ART e participam ativamente de certos eventos ART.

#### **5.4.1.2.3 Atividades**

**PI Planning** - um evento de planejamento presencial baseado em cadência que serve como batimento cardíaco do ART, alinhando todas as equipes na ART a uma missão comum.

**System Demo** - fornece uma visão integrada de novos recursos para a iteração mais recente entregue por todas as equipes no ART. Cada demo fornece aos atores da ART uma medida objetiva do progresso durante o incremento do programa.

**Inspect & Adapt (I&A)** - é um evento significativo onde o estado atual da Solução é demonstrado e avaliado. As equipes então refletem e identificam os itens de *backlog* de melhoria através de uma oficina estruturada de resolução de problemas.

#### **5.4.1. 3 Itens para checklist Nível de Time**

##### **5.4.1.3.1 Artefatos**

**Story** - As histórias carregam os requisitos do Cliente através do fluxo de valor na implementação. As equipes usam histórias para oferecer valor dentro de uma Iteração, e o Proprietário do Produto possui autoridade de conteúdo sobre sua criação e aceitação. As histórias habilitantes - como qualquer História, devem caber dentro de uma Iteração. Embora os Ativadores não precisem do formato de voz do usuário, seus critérios de aceitação esclarecem os requisitos e os testes de suporte.

**Iteration Goals** - um resultado do evento de planejamento de iteração, são um resumo de alto nível dos objetivos comerciais e técnicos que a equipe ágil concorda em realizar em uma iteração. Eles ajudam a garantir o alinhamento com os objetivos do PI.

**Team Backlog** - consiste em histórias de usuários e facilitadores, a maioria das quais é identificada durante as reuniões de Planejamento de PI e Refinamento de *backlog*.

**Team PI Objectives** - são uma descrição resumida dos objetivos comerciais e técnicos específicos que uma Equipe Ágil pretende alcançar no próximo Incremento do Programa (PI).

#### 5.4.1.3.2 Papéis

**Equipe ágil** - Uma equipe *Scrum*, *XP* ou *Kanban* multifuncional que consiste na equipe Dev, bem como no *Scrum Master* e *Product Owner*. Esta equipe de cinco a dez pessoas tem a capacidade e a autoridade para definir, construir e testar um elemento (*Story* ou *Enabler*) do valor da solução dentro de uma Iteração.

**Equipe de Desenvolvimento (Dev Team)** - Uma equipe pequena e interprofissional de desenvolvedores, testadores e outros especialistas, que trabalham de forma colaborativa para oferecer uma fatia de funcionalidade vertical. O Dev Team é um subconjunto da Equipe Ágil.

**O Proprietário do Produto (PO)** - um membro da Equipe Agile e é a autoridade de conteúdo do *Team Backlog*. O PO é responsável por definir *Stories* e priorizar o *backlog*. O PO é o único membro da equipe com poderes para aceitar *Stories* como feito.

**Scrum Master** - um membro da Equipe Agile, o Scrum Master é um líder e treinador da *Agile Team*. O *Scrum Master* ajuda o time a remover impedimentos, facilita eventos em equipe e promove um ambiente para equipes de alto desempenho.

#### 5.4.1.3.3 Atividades

***Iteration Planning*** - um evento em que uma Equipe Ágil determina os Objetivos de Iteração e quanto da carteira de equipe em que eles podem se comprometer durante uma próxima iteração. O número de histórias e facilitadores selecionados é baseado na capacidade da equipe.

***Iteration Review*** - um evento baseado em cadência onde a equipe inspeciona o incremento no final da iteração e ajusta o backlog da equipe com base no *feedback*. Todo o trabalho realizado durante a Iteração é demonstrado durante a Revisão de Iteração.

***Iteration Execution*** - é como a equipe Agile desenvolve um incremento de sistema efetivo, de alta qualidade, trabalhando e testado dentro do *timebox*. Durante a execução, uma Equipe Ágil se reunirá todos os dias para uma reunião de caixa de 15 minutos chamada *Daily Standup Meeting* (DSU). O objetivo do DSU é sincronizar, analisar o progresso e identificar problemas.

***Iteration Retrospective*** - Um evento realizado no final da Iteração onde a Equipe Agile revisa suas práticas e identifica formas de melhorar. A retrospectiva é baseada em informações qualitativas e quantitativas apresentadas durante a Revisão de Iteração. Refinamento de atraso - um evento realizado uma ou duas vezes durante a iteração para refinar, revisar e estimar Histórias e *Enablers* no *backlog* da equipe. O evento é atendido pela equipe ágil e todos os especialistas em assuntos necessários.

***Innovation and Planning (IP) Iteration*** - Uma iteração que oferece às equipes uma oportunidade de exploração e inovação, tempo dedicado ao planejamento e aprendizado através de canais informais e formais. No caso em que uma Versão está no limite PI, as equipes realizam a verificação, validação e documentação final do sistema.



### 5.4.2 Checklist SAFe

#### [READ ME] Matriz Engenharia de Energia:

##### 1 - Nível de Portfólio:

- |                                     |   |   |
|-------------------------------------|---|---|
| <input checked="" type="checkbox"/> | Foram definidos os Temas Estratégicos ou Temas de Investimento? | > |
| <input checked="" type="checkbox"/> | Foi definido o Papel do Epic Owner?                             | > |
| <input checked="" type="checkbox"/> | Foi definido o Papel do Enterprise Architect?                   | > |
| <input checked="" type="checkbox"/> | Foi definido o Papel do Lean Portfolio Management?              | > |
| <input checked="" type="checkbox"/> | Foram definidos os Épicos Arquiteturais?                        | > |
| <input checked="" type="checkbox"/> | Foram definidos os Épicos Empresariais?                         | > |
| <input checked="" type="checkbox"/> | Foi definido um Backlog para o Nível de Portfólio?              | > |
| <input checked="" type="checkbox"/> | Foram definidas as métricas para o Nível de Portfólio?          | > |

## 2 - Nível de Programa:

- ☒ Foi definido o papel do System Architect? >
- ☒ Foi definido o papel do Product Management? >
- ☒ Foi definido o papel do Release Train? >
- ☒ Foi definido um Backlog para o Nível de Programa? >
- ☒ Foram definidas as métricas para o Nível de Programa? >
- ☒ Foi definido o papel dos Business Owners? >
- ☒ Foram definidos os objetivos de PI (Program Increment)? >
- ☒ Foram definidas as Features? >
- ☒ Foram identificados os requisitos não-funcionais? >
- ☒ Foi definido o papel do Customer (Cliente)? >
- ☒ Foi elaborada a visão do produto (Documento de Visão)? >
- ☒ Foi elaborado o Roadmap? >

## 3 - Nível de Time:

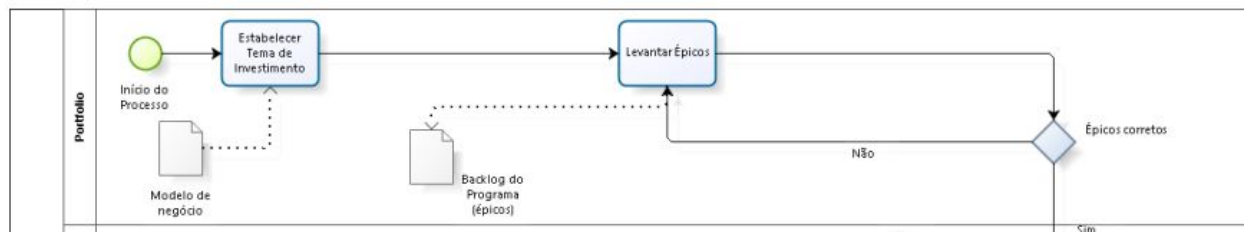
- ☒ Foi definido o Time de desenvolvimento? >
- ☒ Foram escritas as histórias de usuário? >
- ☒ Foi definido um Backlog para o Nível de Time (Kanban)? >
- ☒ Foi definido o papel do Product Owner? >
- ☒ Foi definido o papel do Scrum Master? >
- ☒ Foi definido o Processo Scrum do projeto? >
- ☒ Foram definidas as práticas de qualidade? >
- ☒ Foram realizados os planejamentos de cada iteração? >
- ☒ Foram realizadas as revisões de cada iteração? >
- ☒ Durante as iterações, foram realizadas as reuniões diárias de 15 minutos a cada dia? >
- ☒ Ao final de cada iteração, foi realizada a retrospectiva da iteração? >
- ☒ Foram realizadas iterações específicas de Inovação e Planejamento ao final de cada incremento do nível de >

## 5.5 Realizar Inspeção

Para o projeto foi escolhido um processo de um trabalho de Requisitos de *Software* denominado “Processo de Engenharia de Requisitos - Matriz Engenharia de Energia” elaborado no Primeiro semestre de 2015.

## Nível de Portfólio

No nível de portfólio foram definidos o seguinte *checklist* que foi separado em Artefatos, Papéis e Atividades. Para cada um dos itens contidos nesse *checklist* o projeto será validado. O processo do projeto da Matriz Engenharia de Energia no nível de portfólio pode ser evidenciado pela seguinte imagem:



Algumas atividades do projeto foram definidas com tarefas, vale ressaltar que foram feitas considerações no apêndice B1, relacionado a atividade Levantar Épicos.

No nível de portfólio as atividades foram:

- **Levantar Épicos:** Fazer *Workshop* de Requisitos, Escrever Épicos, Validar Épicos, Manter Rastreabilidade.
- **Estabelecer Tema de Investimento:** Fazer *Workshop* de Requisitos (evidenciado no texto).

Os Artefatos foram:

- Modelo de Negócios
- *Backlog* do Programa (Épicos)

Os Papéis Envolvidos foram:

- *Product Manager, Scrum Master, Time*

### **Nível de Programa**

Após a aplicação do checklist desenvolvido para a inspeção, os artefatos desenvolvidos presentes consistiam em:

- *PI Objectives*
- *Features*
- *Program Backlog*

Das atividades necessárias a serem desenvolvidas a nível de programa, estavam presentes:

- *PI Planning*

Dos papéis necessários a serem desenvolvidos a nível de programa, estavam presentes:

- *Product Management*

### **Nível de Time**

Após a aplicação do checklist desenvolvido para a inspeção, os artefatos desenvolvidos presentes consistiam em:

- Histórias de usuário
- *Team Backlog (Kanban)*

Dos papéis necessários a serem desenvolvidas a nível de time, estavam presentes:

- *Scrum Master*

- Time de desenvolvimento

Das atividades necessárias a serem desenvolvidos a nível de time, estavam presentes:

- *Iteration Planning*
- *Iteration Review*
- *Iteration Execution*
- *Iteration Retrospective*

## 5.6 Relatorio de Resultados

### 5.6.1 Relatório da Inspeção do Nível de Portfólio

Como resultado do checklist no nível de Portfólio foram cumpridos um total de 4 itens de um total de 8 definidos como atividades essenciais no nível. Algumas atividades do projeto foram definidas com tarefas, vale ressaltar que foram feitas considerações no apêndice, relacionado a atividades.

#### 5.6.1.1 Omissões

As omissões abaixo são as atividades, papéis ou artefatos que foram omitidos no processo avaliado pelo checklist no nível de Portfólio.

#### Papel do *Epic Owner*

**Justificativa:** No contexto a qual o processo foi planejado, a equipe era muito pequena, não havendo necessidade da definição do *Epic Owner*.

#### *Enterprise Architect*

**Justificativa:** No contexto a qual o processo foi planejado, a equipe era muito pequena, não havendo necessidade da definição do *Enterprise Architect*.

---

***Portfolio Management***

**Justificativa:** Não houve.

**Métricas para o Nível de Portfólio**

**Justificativa:** Não houve.

**Resultado:** 4/8 itens cumpridos.

**5.6.2 Relatório da Inspeção do Nível de Programa**

Como resultado do checklist no nível de Programa foram cumpridos 8 itens de um total de 12 definidos como atividades essenciais no nível. Algumas atividades do projeto foram definidas com tarefas, vale ressaltar que foram feitas considerações no apêndice, relacionado a atividades.

**5.6.2.1 Omissões**

As omissões abaixo são as atividades, papéis ou artefatos que foram omitidos no processo avaliado pelo checklist no nível de Program.

**papel dos *Business Owners***

Justificativa: No contexto a qual o processo foi planejado, a equipe era muito pequena, não havendo necessidade da definição de um *Business Owner*.

**métricas para o Nível de Programa**

Justificativa: Não houve

**papel do *Release Train***

Justificativa: No contexto a qual o processo foi planejado, a equipe era muito pequena, não havendo necessidade da definição do *Release Train Engineer*.

**papel do *System Architect***

Justificativa: No contexto a qual o processo foi planejado, a equipe era muito pequena, não havendo necessidade da definição do *System Architect*.

**Resultado:** 8/12 itens cumpridos.

### 5.6.3 Relatório da Inspeção do Nível de Time

Como resultado do checklist no nível de Time foram cumpridos 9 itens de um total de 12 definidos como atividades essenciais no nível. Algumas atividades do projeto foram definidas com tarefas, vale ressaltar que foram feitas considerações no apêndice, relacionado a atividades.

Podemos observar a falta do item relacionada às práticas de qualidade. Sabe-se da importância da qualidade de um software, pois um software sem qualidade é algo instável, difícil de manter e inseguro.

#### 5.6.3.1 Omissões

As omissões abaixo são as atividades, papéis ou artefatos que foram omitidos no processo avaliado pelo checklist no nível de time.

##### ***Program Increment -( PI)***

Justificativa: Não houve

##### **práticas de qualidade**

Justificativa: Não houve

##### **papel do *Product Owner***

Justificativa: No contexto a qual o processo foi planejado, a equipe era muito pequena, não havendo necessidade da definição do *Product Owner*.

Resultado: 9/12 itens cumpridos.

## 5.7 Propostas de Melhoria

As propostas de melhorias abaixo, tem por finalidade adequar o processo avaliado, para o SAFe, afim de garantir a sua consistência em todos os níveis

### 5.7.1 Proposta de melhoria no Nível de Portfólio

Com os resultados obtidos como proposta de melhoria, seria necessário melhorar alguns aspectos:

- A atividade de Levantar requisitos no processo apresentado possui muitas tarefas, em que, poderia ser modularizado para atividades diferentes. Em específico poderia ser utilizado no processo uma atividade separada de Validar Épicos.
- No nível de portfólio o papel do do **Portfolio Management**, é obrigatório esta presente, ou ter um outro papel que o desempenho quesito que não consta no processo, desta forma o processo deverá passar por uma melhoria para adição deste papel.
- Especificar melhor na descrição de épicos e temas de investimento papéis envolvidos, estes papéis devem ser no nível de portfólio.
- Deve ser definida uma atividade de Gerenciamento de Mudanças no Nível de Portfólio. Essa atividade teria como função ter um controle sobre novos temas de investimentos e épicos principalmente. No processo atual não é possível ter um controle completo das atividades estabelecidas.

### 5.7.2 Proposta de melhoria no Nível de Programa

De acordo com os dados levantados após o *checklist* percebe-se as possíveis melhorias a nível de program:

- O nível de programa por ser o nível onde os requisitos não estão muitos abstratos já é possível definir métricas para auxiliar na qualidade do código podendo afetar diretamente os requisitos, uma boa solução seria desenvolver as **métricas para o Nível de Programa** onde essas métricas poderiam auxiliar a criar o plano de qualidade do nível de time .
- As atividades do nível de programa estão sendo desempenhadas por papéis do nível de programa, deve-se ressaltar que cada nível tem os papéis responsáveis por realizar as atividades, que no processo em questão estão nomeadas como tarefas. Uma boa solução seria realizar um estudo do nível do programa no SAFe e alterar as atribuições das atividades para os papéis correspondentes.

### 5.7.3 Proposta de melhoria no Nível de Time



De acordo com os dados levantados após o *checklist* percebe-se as possíveis melhorias a nível de time:

- A equipe justificou o número da equipe pequeno não necessitar de *Product Owner*, mas uma sugestão seria deixar alguém responsável pelos critérios de aceitação, por escrever os casos de uso, etc. É uma prática que ajuda a entender o cliente e seus objetivos, facilitando inclusive outras etapas do processo, como o *Product Increment* etc.
- A equipe não define as práticas de qualidade, o que é um erro grave no desenvolvimento de um *software*. São inúmeros os motivos de ter um plano que define as práticas de qualidade como garantir uma camada a mais de proteção, legibilidade do código, complexidade, gerar mais confiança do cliente ao produto uma vez que evita falhas etc. Recomenda-se fortemente a inclusão de um plano de prática de qualidade.

## 6. PROCESSO ANTERIOR

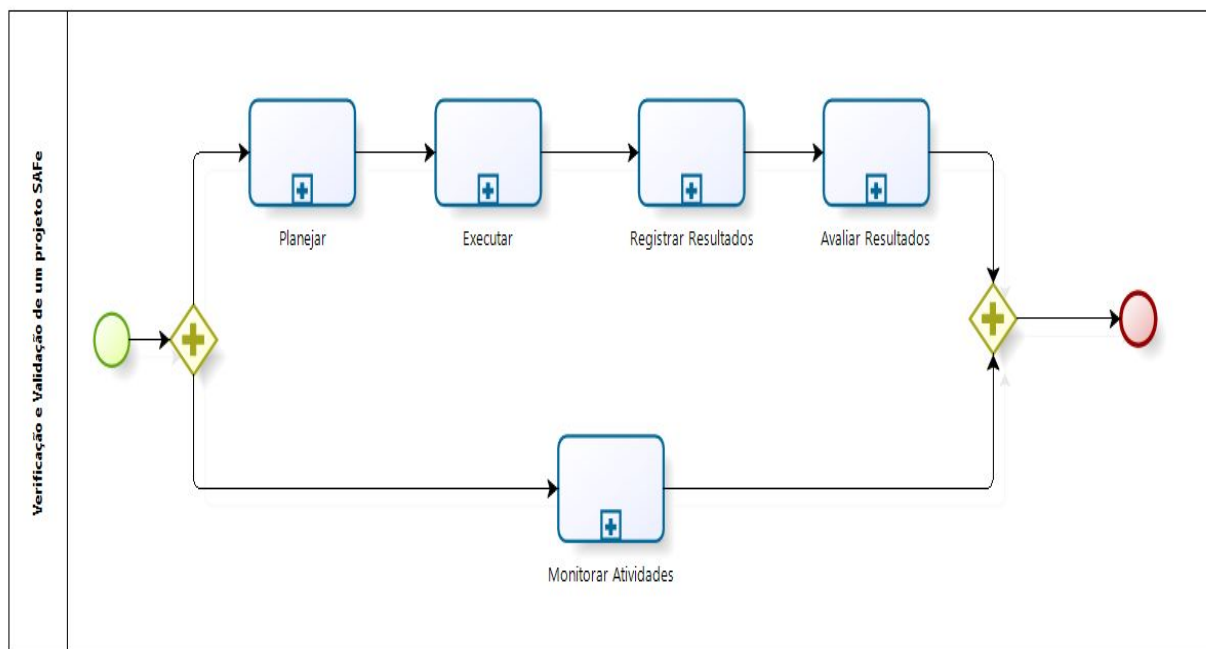


Figura 1 - Processo de Verificação e Validação

A imagem acima é uma visão resumida do processo adotado pelo grupo para a realização de Verificação e Validação. Como o projeto baseia-se na técnica de inspeção, o processo apresentado foi definido baseando-se nas atividades comumente utilizadas por ela. Ele consiste de basicamente 5 sub-atividades que serão detalhadas a seguir. De forma genérica, as atividades são planejamento, execução, registro dos resultados, avaliação dos resultados e em paralelo e essas, é executado o monitoramento das atividades.

## 6.1 Subprocesso Planejar

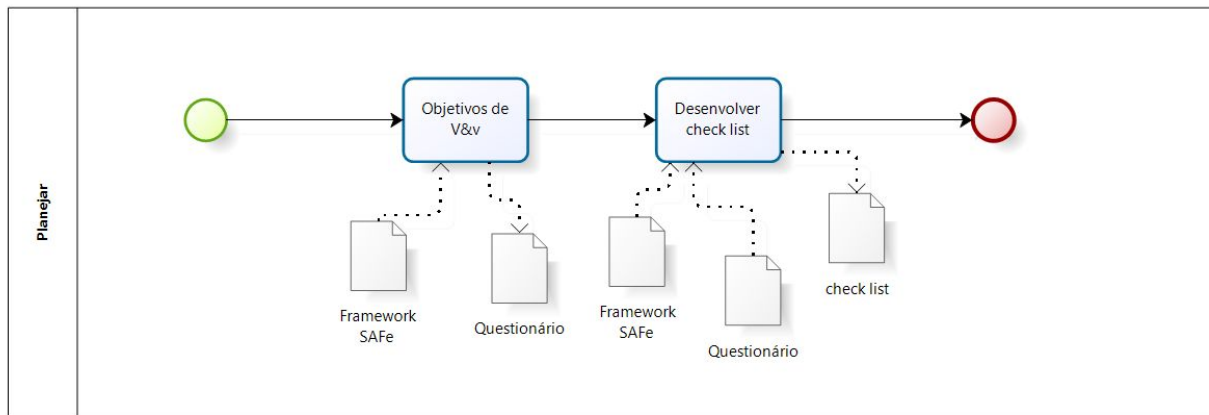


Figura 2 - Planejar Inspeção

O subprocesso de Planejamento da inspeção é bastante importante para o restante do processo. Neste subprocesso, é importante identificar quais serão os objetivos de Verificação e Validação e como ele ocorrerá, assim como o que do produto deve ser avaliado.

Para esse projeto, o principal artefato a ser produzido é o checklist, que será criado utilizando um questionário baseado no framework SAFe. O restante dos documentos que detalham como processo ocorrerá foram definidos anteriormente. Dessa forma, durante o planejamento, seria suficiente apenas atualizá-los.

## 6.2 Subprocesso Executar

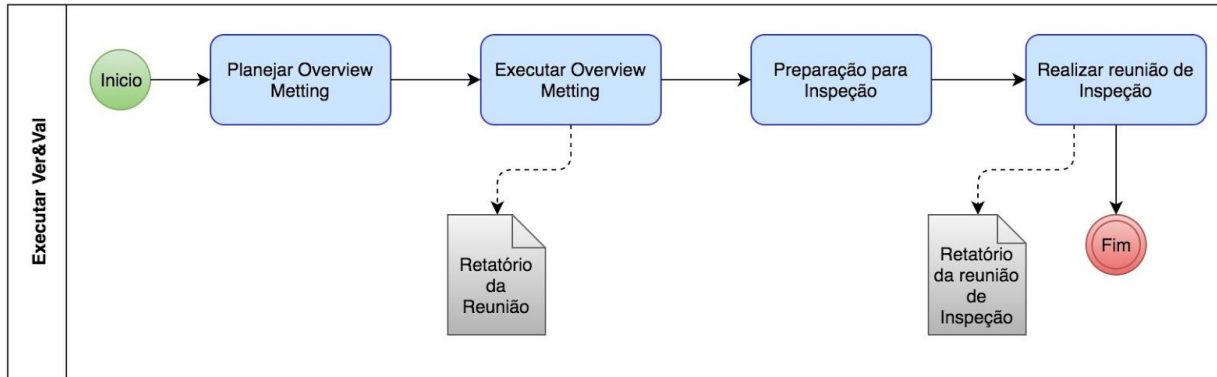


Figura 3 - processo de Executar Inspeção

Antes de executar a reunião de inspeção, algumas atividades são normalmente realizadas. Estas são:

- **Overview meeting:**

Essa é uma reunião informal em que o autor tem a oportunidade de descrever as *features* importantes do produto para o time de inspeção, trata-se também do contexto. Essa atividade pode ser omitida caso essa informação já seja conhecida pelos outros participantes.

- **Preparação:**

Cada participante é responsável, nesta atividade, por examinar os artefatos de trabalho antes a reunião de inspeção. Nesta etapa, normalmente, são identificados 75% dos erros encontrados durante inspeções. Também são levantadas questões, classificação e outros aspectos que não englobam erros mas deseja-se discuti-los durante a reunião.

Além da análise dos documentos para encontrar erros e questões, também é verificado se o que foi desenvolvido está de acordo com as especificações e padrões. A fim de atingir os objetivos dessa atividade, o uso de *checklist* é recomendado e bastante utilizado.

- **Reunião de Inspeção:**

A reunião de Inspeção é guiada pelo moderador e pelo leitor. É interessante comentar que o moderador pode determinar que a data da reunião pode ser remarcada, caso ele perceba que não houve tempo suficiente para os participantes se prepararem.

O leitor descreve cada segmento que deve ser analisado no momento e os inspetores respondem, levantando defeitos, questões e sugestões. Então, o autor durante esse momento, deve garantir que entendeu qual foi o problema apontado, para que possa realizar a sua réplica ou corrigir o defeito futuramente.

Enquanto tudo isso ocorre, o escritor anota os pontos levantados, gerando um documento que é a principal entrega dessa atividade. Ao mesmo tempo, o moderador resolve problemas, falta de participação ou comportamento inapropriado dos participantes da reunião.

Por último, um laudo é gerado pelo moderador, definindo-se, entre outras coisas, o estado do produto. Esse estado pode ser definido como aceito como está, aceito com pequenas revisões ou é necessário grandes alterações, no último caso, é comum uma inspeção futura.

Outra possível saída dessa atividade, que faremos no projeto, é um documento indicando o que pode ser melhorado para futuros processos de inspeção.

### 6.3 Subprocesso Registrar Resultados

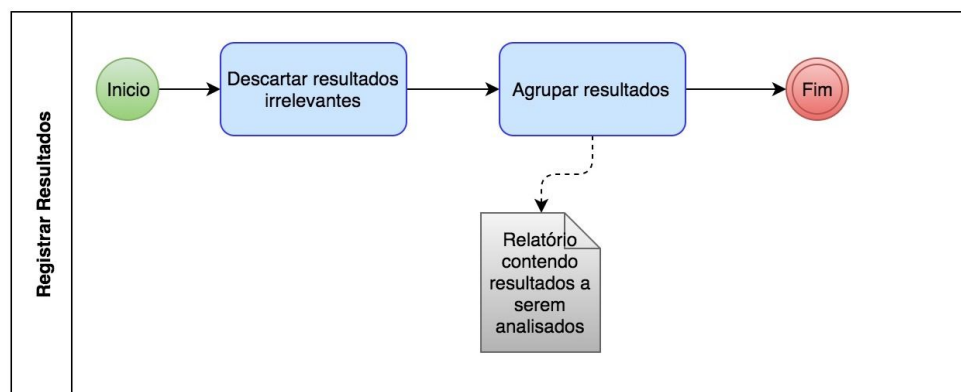


Figura 4 - processo de Registrar Resultados

Essa Atividade recebe como insumo, o documento redigido na atividade de “Executar Verificação e Validação” que contém os pontos levantados durante a reunião. Os dados registrados são melhor organizados e detalhados nessa atividade. Os dados similares são agrupados e os desnecessários descartados.

O objetivo é gerar um documento mais formal e compreensível, para que facilite futuras consultas, por parte da equipe encarregada em realizar as modificações detalhadas, e análises com o propósito de avaliação dos dados e das causas de problemas por parte da equipe de qualidade.

## 6.4 Subprocesso Avaliar Resultados

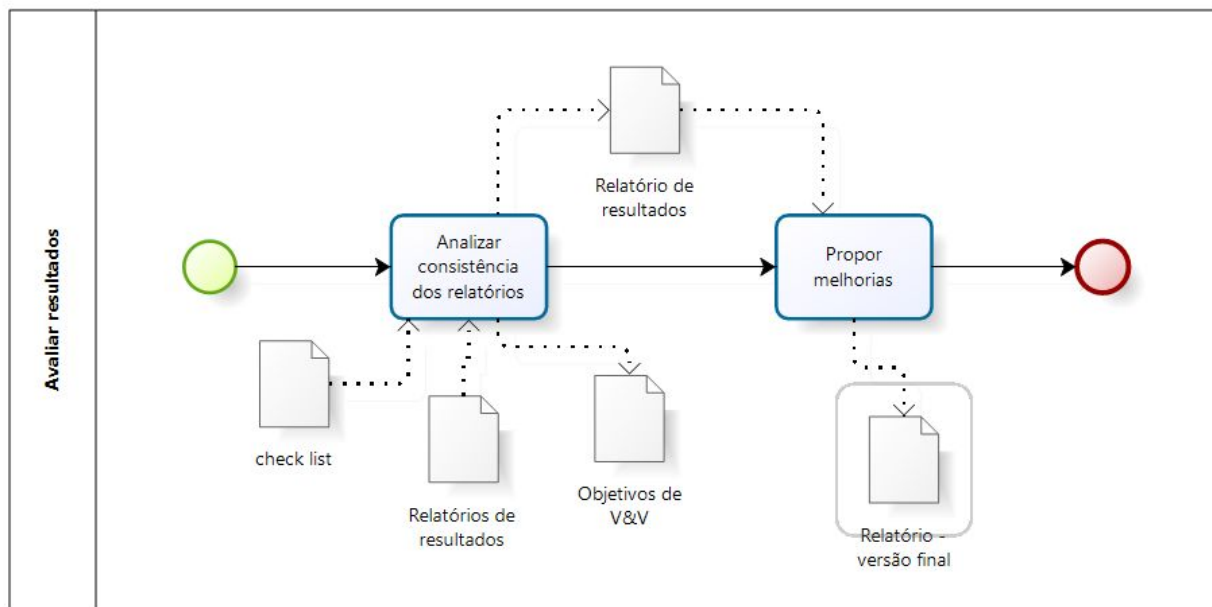


Figura 5 - Avaliar Resultados

## 6.5 Subprocesso Monitorar Atividades

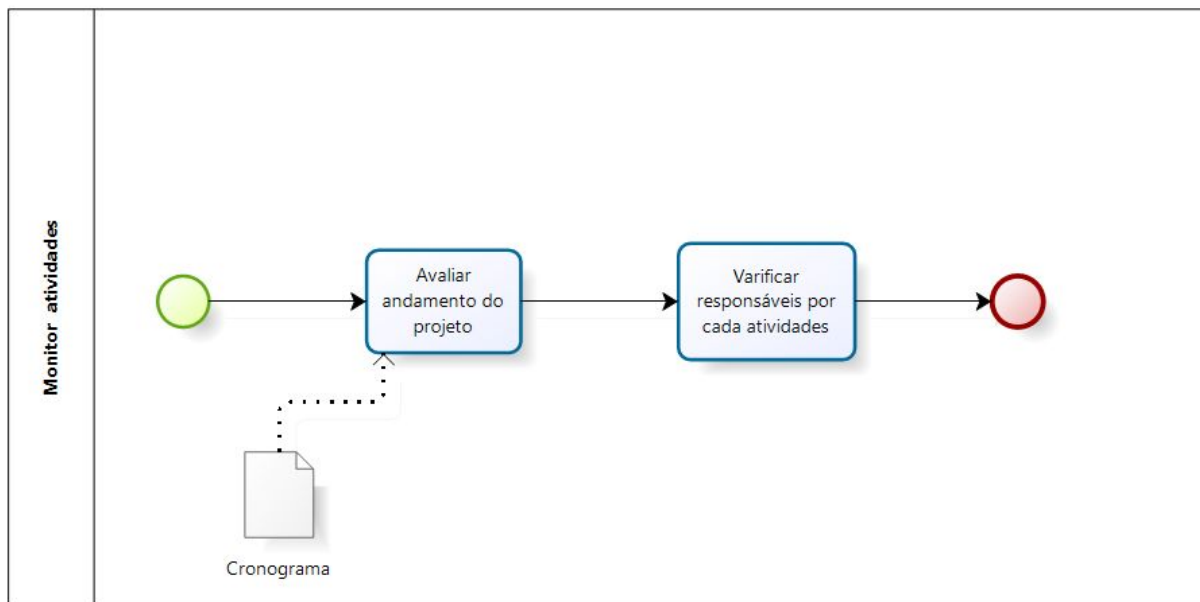


Figura 6 - Monitorar atividades

## 7. FOCO DE QUALIDADE

O *Scaled Agile Framework* são definidas atividades em diferentes níveis de abstração que tornam viável a execução de um projeto maior utilizando métodos ágeis.

Para a análise da qualidade de um processo SAFe na disciplina de Requisitos os processos são avaliados de acordo com aspectos relativos a cada contexto, por outro lado, são avaliados também a partir de atividades chave em cada nível.

Para a avaliação do trabalho, são definidas atividades importantes para em cada nível do SAFe, são elas:

**Atividades gerais:** Gerenciar Requisitos (ocorre em paralelo com processo todo), Definição de Requisitos não funcionais.

**Atividades no nível de Portfólio:** Definir Temas de Investimento, Definir Épicos, Priorizar e Validar Épicos.

**Atividades no nível de Programa:** Estabelecer a visão do produto, Definir *Features*, Priorizar e Validar *Features*, *Roadmap*, Elaborar *Program Increment*.

**Atividades no nível de Time:** Definir Histórias de Usuário, Priorizar Histórias, Implementação e abordagem do *Scrum* adotada pela equipe (verificar se atividades são realmente do *Scrum*).

## 8. CONCLUSÃO

Percebe-se que este projeto, provavelmente, trará contribuições relevantes às disciplinas e alunos de universidades que utilizam o *framework* SAFe. A partir desse *checklist*, o aprendizado das disciplinas acadêmicas que ensinam sobre engenharia de requisitos de software ágeis pode não se limitar apenas a teoria, mas partir também para a prática, no âmbito da verificação e validação. Agregar este *checklist* aos projetos, pode trazer experiências relacionadas ao aspecto de melhoria da qualidade dos requisitos elicitados e, conseqüentemente, melhoria da qualidade na maneira de como a disciplina é ministrada.

Além dessa contribuição para o ambiente acadêmico, este projeto pode se estender para o ambiente corporativo. Possibilitando que empresas identifiquem aspectos frágeis do processo que estão utilizando e, a partir daí, avaliem possíveis melhorias.

No panorama atual de desenvolvimento de software, muito se tem discutido sobre metodologias de desenvolvimento, como métodos tradicionais e ágeis. Um *framework* que vem se destacando no mercado é o SAFe, *Scaled Agile Framework*, que permite escalar prática ágeis em projetos de software com foco em requisitos. Embora muitas empresas façam adaptação deste *framework* para suas necessidades muitas delas pecam pela falta de conhecimento sobre o que pode ser omitido e acrescentado ao *framework*, uma vez que atividades chaves, artefatos ou papéis que são essenciais para um bom processo seguindo esse *framework* deixam de estar presentes acarretam fortes perdas do potencial proporcionado pelo mesmo.

Tendo em vista o panorama descrito acima, e a importância de validar os processos adaptados seguindo este *framework*, destaca-se a importância deste trabalho não só para projetos acadêmicos, mas também para empresas de pequeno porte que precisam adaptar o SAFe para seu contexto específico.

Uma vez que alinhar as diretrizes sobre os papéis, artefatos e atividades e o quanto os mesmos podem agregar de valor para um bom modelo de desenvolvimento, pode deixar nítido o que é adaptável e o que não é.

## 9. TRABALHOS FUTUROS

Como a disciplina possui um tempo de duração curta, o escopo não podia englobar outros métodos de Verificação e Validação para medir a qualidade dos processos. Devido a isso, o tópico de trabalhos futuros foi criado para que sejam registradas melhorias e acréscimos que serão feitos no trabalho.

Ao final da disciplina, foram identificadas melhorias que poderiam ser feitas ao *checklist* desenvolvido, tal como a determinação de diferentes pesos para os itens do *checklist* com o intuito de enfatizar os itens mais importantes do *framework* SAFe. Ou seja, enfatizar os papéis e atividades do SAFe que são imprescindíveis para qualquer processo de engenharia de requisitos que utiliza o *framework*. Essa ação permitiria uma análise qualitativa aprimorada do processo que está sendo avaliado. Por exemplo: a atividade da construção do Documento de Visão é importante para o processo, logo é uma atividade que não deve faltar. Já a atividade da realização da Iteração de Inovação e Planejamento é interessante para a melhoria da qualidade do produto, mas não tem uma relevância comparável ao da existência do Documento de Visão, logo a atividade da construção do Documento de Visão terá um peso maior na aplicação do *checklist*.

A fim de determinar o peso de cada um dos itens do *checklist* diversas estratégias podem ser adotadas. Uma possibilidade seria consultar especialistas no *framework* SAFe e especialistas em níveis de maturidade e, a partir da opinião de cada um deles, determinar os pesos de cada um dos itens. Outra possibilidade seria um estudo estatístico baseado em projetos que utilizam o SAFe. Neste estudo, seriam analisados tanto projetos que deram certo, como os mal sucedidos, identificando em cada um deles quais itens de avaliação foram utilizados e quais foram descartados (ex. papéis, atividades, etc). A partir deste estudo seria possível identificar a correlação desses itens de avaliação e o sucesso do projeto. Dessa forma, seria possível determinar pesos de cada item do checklist utilizando como base sua correlação com o sucesso dos projetos.



Foi possível constatar ainda, uma lacuna de estudos acadêmicos relacionados à aplicação de *frameworks* ágeis, como o SAFe, em grandes organizações. Quando se observa especificamente a área de verificação e validação de métodos ágeis, no que diz respeito ao processo e a abordagens de verificação e validação, a lacuna é maior ainda. Desse modo, este projeto pode abrir o horizonte para novas pesquisas voltadas para a verificação e validação de processos ágeis em geral, não limitando-se somente ao SAFe.

Outro aspecto que deve ser considerado para trabalhos futuros é que o *framework* estudado é escalável, customizável e sempre tem recebido novas atualizações. Devido a isso, o trabalho de aferição da qualidade dos processos de engenharia de requisitos que utilizam o SAFe deve permitir também mudanças para que o mesmo esteja sempre em conformidade com a última versão do *framework* ou com a versão escolhida.

Por fim, além da aplicação deste estudo para melhoria das próprias disciplinas da universidade, ele poderia servir de inspiração para empresas da área de engenharia de software. Futuramente, por exemplo, ele pode ser adaptado para auxiliar empresas de desenvolvimento de software na utilização do SAFe para melhorar a qualidade de seus produtos e processos.

## 10. REFERÊNCIAS

1. SANTOS C. M. C, PIMENTA C. A. M., NOBRE M. R. C. 2007. “A ESTRATÉGIA PICO PARA A CONSTRUÇÃO DA PERGUNTA DE PESQUISA E BUSCA DE EVIDÊNCIAS”. Disponível em <[http://www.scielo.br/pdf/rlae/v15n3/pt\\_v15n3a23.pdf](http://www.scielo.br/pdf/rlae/v15n3/pt_v15n3a23.pdf)>, acesso em 23 de Abril de 2017.
2. GALVÃO, M. R.; SAWADA, N. O., TREVIZAN, M. A. 2004. “REVISÃO SISTEMÁTICA: RECURSO QUE PROPORCIONA A INCORPORAÇÃO DAS EVIDÊNCIAS NA PRÁTICA DA ENFERMAGEM”. Disponível em <<http://www.scielo.br/pdf/rlae/v12n3/v12n3a14>>, acesso em 23 de Abril de 2017.
3. SOMMERVILLE, I., Engenharia de Software. 8ª ed. São Paulo: Addison Wesley, 2007.  
ALVES, Amann da L. M., GOUVÊA, André L. C. *Técnica de inspeção de software baseada em técnica de leitura e verificação de modelos UML de alto nível*. UNIRIO. Rio de Janeiro, 2013. Disponível em: <<http://bsi.uniriotec.br/tcc/201308AlvesGouvea.pdf>>, acesso em 3 abril de 2017.
4. Asana. Disponível em: <<https://app.asana.com/>>, acesso em 28 abril de 2017.
5. SAFe, Scaled Agile Framework . Disponível em: <<http://www.scaledagileframework.com/#>> acesso em 22 de junho de 2017.
6. SAFe, Portfolio Level. Disponível em: <<http://www.scaledagileframework.com/portfolio-level/>>, acessado em 22 de junho de 2017.
7. SAFe, Program Level. Disponível em: <<http://www.scaledagileframework.com/program-level/>>, acessado em 22 de junho de 2017.
8. SAFe, Team Level. Disponível em: <<http://www.scaledagileframework.com/team-level/>>, acessado em 22 de junho de 2017.