

# Requisitos – Aula 15

Professores: Milene Serrano e Maurício Serrano

## Agenda

- › Considerações Iniciais
- › Modelagem de Requisitos – Abordagens Ágeis
  - Scrum
    - › *Product Backlog*;
    - › *Sprint Backlog*, e
    - › *Epics e User Stories*.
  - SAFe
    - › *Epics, Features e User Stories*.
- › Debate
- › Considerações Finais

# Considerações Iniciais

# Considerações Iniciais

<http://agilemethodology.org/>

Sabemos que na atualidade existem muitas abordagens que fazem uso de princípios ágeis para orientar e conduzir o desenvolvimento de software.

Dentre as mais conhecidas, destacam-se:

- OpenUp;
- XP;
- Scrum,
- Lean, e
- SAFe.



<http://www.scaledagileframework.com/#>

<http://www.desenvolvimentoagil.com.br/xp/>



<http://epf.eclipse.org/wikis/openup/>



<http://scrummethodology.com/>

<https://www.lean.org/WhatsLean/>



<http://agilemethodology.org/>

# Considerações Iniciais

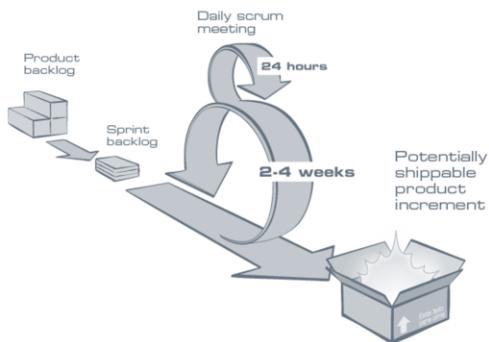
Uma das mais populares é o **Scrum**.

Portanto, iremos conhecer os principais artefatos para modelagem de requisitos nessa metodologia.

**FOCO:** Apenas em artefatos de mais alto nível de abstração, mais próximos à Engenharia de Requisitos. Ok?

O objetivo é gerar um produto de software. Portanto, temos um artefato mais amplo, o qual acorda as necessidades do produto como um todo – **Product Backlog**.

Por ser uma metodologia ágil, iterativa e incremental, baseada em ciclos curtos, gerando entregáveis de valor ao cliente a cada ciclo, temos o conceito de **Sprint**. Portanto, temos outros artefatos de granularidade menor, chamados **Sprint Backlogs**.



# Modelagem de Requisitos

Orientando-se pelo Scrum

# Modelagem de Requisitos

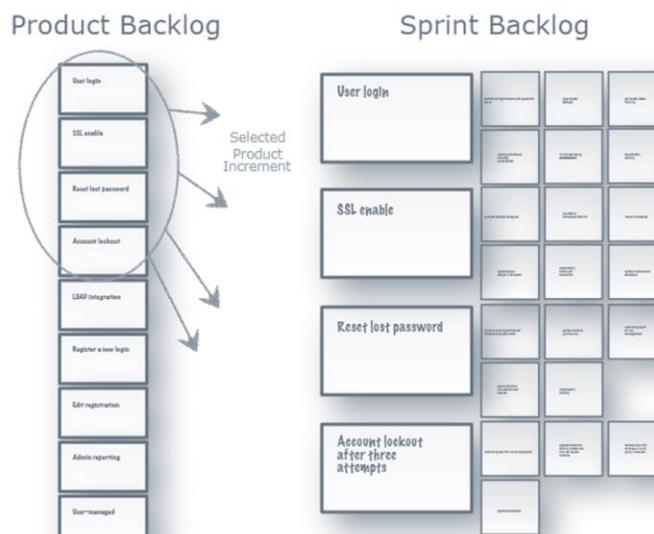
*Product Backlog*

## Product Backlog

O **Product Backlog** é uma lista contendo todas as funcionalidades desejadas para um produto. O conteúdo desta lista é definido pelo **Product Owner**. O **Product Backlog** não precisa estar completo no início de um projeto. Pode-se começar com tudo aquilo que é mais óbvio em um primeiro momento. Com o tempo, o **Product Backlog** cresce e muda à medida que se aprende mais sobre o produto e seus usuários.

Durante a **Sprint Planning Meeting**, o **Product Owner** prioriza os itens do **Product Backlog** e os descreve para a equipe. A equipe então determina que itens será capaz de completar durante a **Sprint** que está por começar. Tais itens são, então, transferidos do **Product Backlog** para o **Sprint Backlog**. Ao fazer isso, a equipe “quebra” cada item do **Product Backlog** em uma ou mais tarefas do **Sprint Backlog**. Isso ajuda a dividir o trabalho entre os membros da equipe. Podem fazer parte do **Product Backlog** tarefas técnicas ou atividades diretamente relacionadas às funcionalidades solicitadas.

## *Product Backlog e Sprint Backlog*



## Product Backlog

| ID | Theme          | As a/an               | I want to...   | So that...  | Notes   | Priority | Status        |
|----|----------------|-----------------------|--|---|---|----------|---------------|
| 1  | Functional     | Student               | To have a calendar archive of all activities for different clubs | I can know all the activities I can attend  |   | High     | Not coded yet |
| 1  | Functional     | Student               | Add my activities of interest to my personal calendar            | I can organize my personal events   | Club meetings, internship fairs, appointments, etc          | High     | In progress   |
| 1  | Functional     | Student               | Have reminders of my events                                      | I remember them   |   | Low      | To do         |
| 1  | Non-functional | Student               | Be able to group my events and color-code it                     | I can view my calendar easier   |   | Low      | To do         |
| 2  | Functional     | Student               | Have a platform so I can view other student stories              | I can hear and learn from others and understanding where they are coming from               | Not limited to storyboards, video, audio, and other visuals | High     | Done          |
| 2  | Functional     | Student               | Be able to upload my story and perspective on the website        | I can share my experience with others   |   | High     | In progress   |
| 2  | Non-functional | Advisor               | Be able to like and share stories I see                          | Show other struggling students  |   | Low      | To do         |
| 2  | Non-functional | Student               | Be able to view all stories in one place                         | It's easier for me to view it   |   | Medium   | Done          |
| 2  | Functional     | MyStory contributor   | Be able to upload files larger than 200 MB                       | I can show a longer video/audio   |   | Low      | To do         |
| 2  | Functional     | MyStory contributor   | Be able to edit my MyStory post                                  | I can update new events that occurred   |   | High     | In progress   |
| 2  | Non-functional | Student               | View MyStory in sub categories by majors and/or experiences      | I can view the stories most related to me   | Experiences: how to get an internship, job, etc             | Medium   | Done          |
| 3  | Functional     | Student               | Have a general Q&A forum for any question                        | I can ask mentors questions   |   | Medium   | To do         |
| 3  | Functional     | Student               | Have a Q&A forum for each individual post                        | I can ask the MyStory additional questions if their post does not answer all my questions   |   | Medium   | Not coded yet |
| 4  | Functional     | Senior-in-high school | Be able to contact an advisor                                    | I can have a general overview of the university and create a close connection to my advisor | See their email, number, and office hours                   | Low      | To do         |
| 5  | Non-functional | Bothell Student       | See Seattle/Bothell's campus resources                           | I have extra help if I need it  |   | Low      | Not coded yet |
| 5  | Function       | Bothell Student       | Add resources to resources list if that already listed           | I can share it to everyone of what helped me  |   | Low      | To do         |
| 6  | Functional     | MyStory contributor   | Have an anonymous option   | I can still have my privacy   |   | High     | To do         |
| 7  | Functional     | Student               | Geo-tagged locations for my events                               | I can find them easily  | Across Bothell and Seattle campus                           | Low      | To do         |

Um item por vez é prioridade top\_1

## Product Backlog Items

Top itens são menores,  
menores do que  
 $\frac{1}{4}$  de uma **Sprint**.

Como especificar os itens?



A especificação dos itens é baseada em histórias de usuários (**User Stories**), as quais focam mais em "O que" e não no "como".



Esses itens são **customer-centric features**.



Podem ter um critério de aceitação.



Esforços para cumprimento desses itens podem ser estimados pelos desenvolvedores em **Story Points**.



## Epics e User Stories



Account lockout after three attempts

Acceptance Criteria: ....

**Small**

As a Game Player,

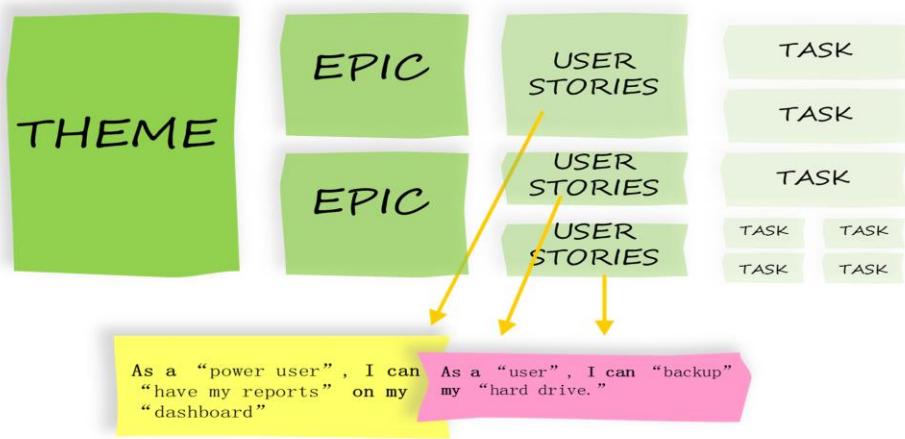
I want my Rocket to move back and forth when I press left and right arrows

so that I can avoid asteroids

## Epics e User Stories

Podem ter outros níveis de granularidade na especificação

### USER STORIES



As a "power user", I can  
"have my reports" on my  
"dashboard"

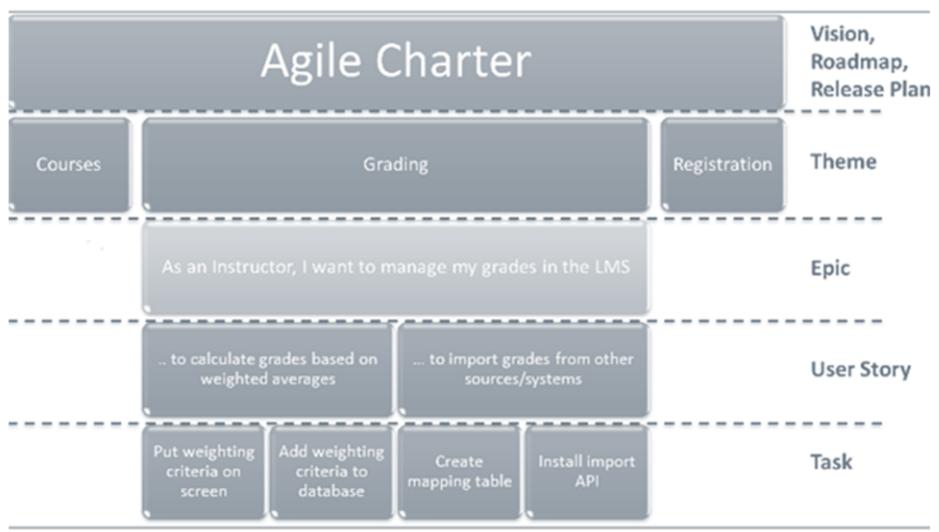
As a "user", I can "backup"  
my "hard drive."

[www.agile-scrum.be](http://www.agile-scrum.be)

## Epics e User Stories – Exemplo I



## Epics e User Stories – Exemplo II



# Modelagem de Requisitos

*Sprint Backlog*

## Sprint Backlog

O **Sprint Backlog** é uma lista de tarefas que o **Scrum Team** se compromete a fazer em uma **Sprint**. Essa lista é comumente organizada em um **Taskboard**.

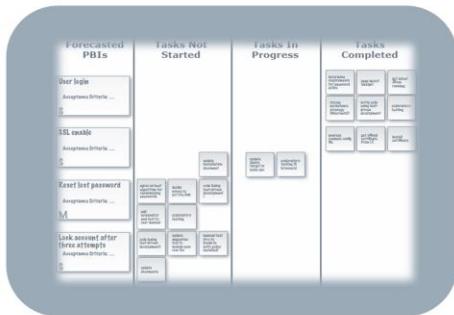
Os itens do **Sprint Backlog** são extraídos do **Product Backlog**, pela equipe, com base nas prioridades definidas pelo **Product Owner** bem como na percepção da equipe sobre o tempo que será necessário para completar as várias funcionalidades.

Cabe a equipe determinar a quantidade de itens do **Product Backlog** que serão trazidos para o **Sprint Backlog**.

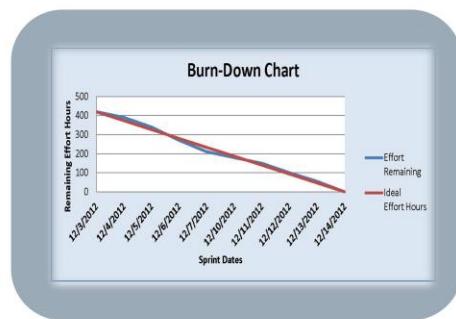
Durante uma **Sprint**, o **Scrum Master** mantém o **Sprint Backlog**, atualizando-o e tornando-o compatível com as tarefas completadas bem como considerando o tempo que a equipe acredita que será necessário para completar aquelas que ainda estão prontas.

As estimativas são atualizadas diariamente, sendo colocadas em um gráfico, resultando em um **Sprint Burndown Chart**.

## Artefatos que surgem...



Taskboard



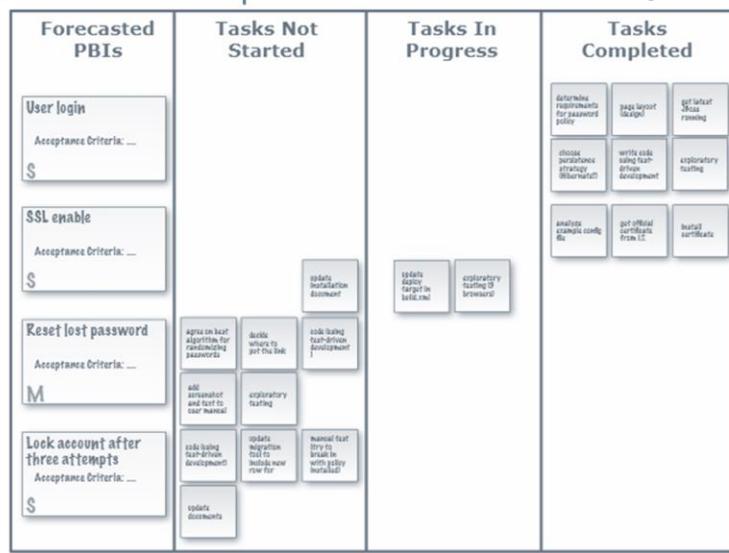
Burndown Chart

Vamos conhecer cada um deles ...



# Taskboard – Exemplo I

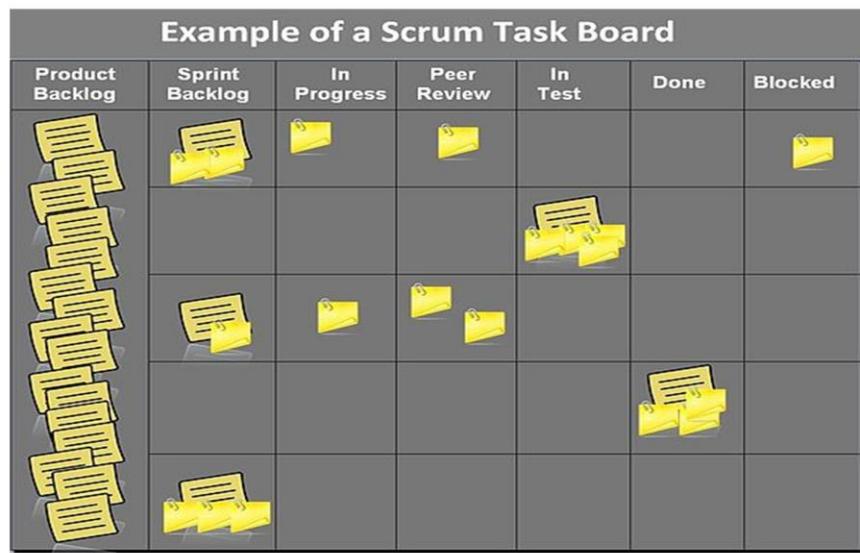
PBIs – Product Backlog Items



PBIs – Product Backlog Items

## Taskboard – Exemplo II

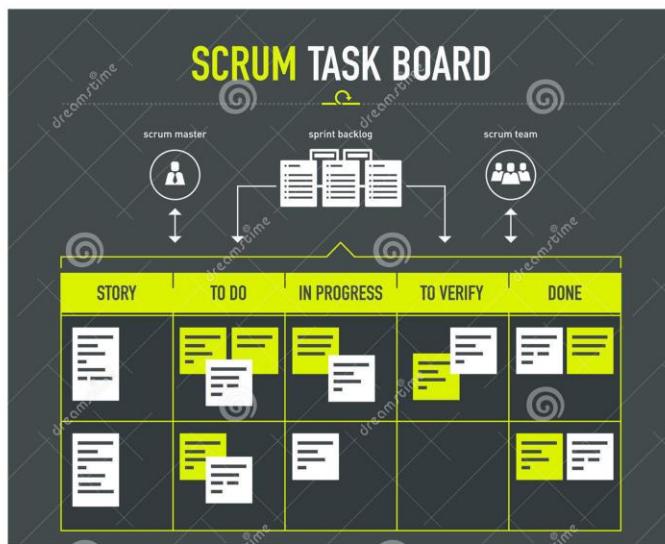
PBIs – Product Backlog Items



PBIs – Product Backlog Items

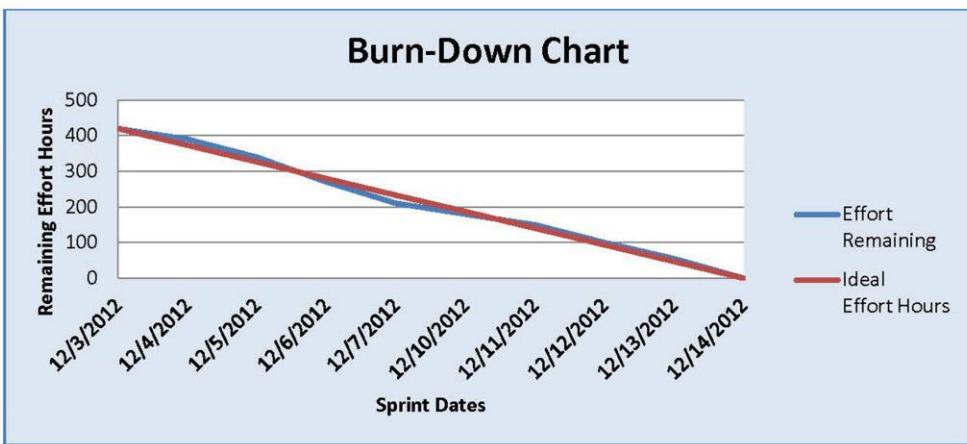
## Taskboard – Exemplo III

PBIs – Product Backlog Items

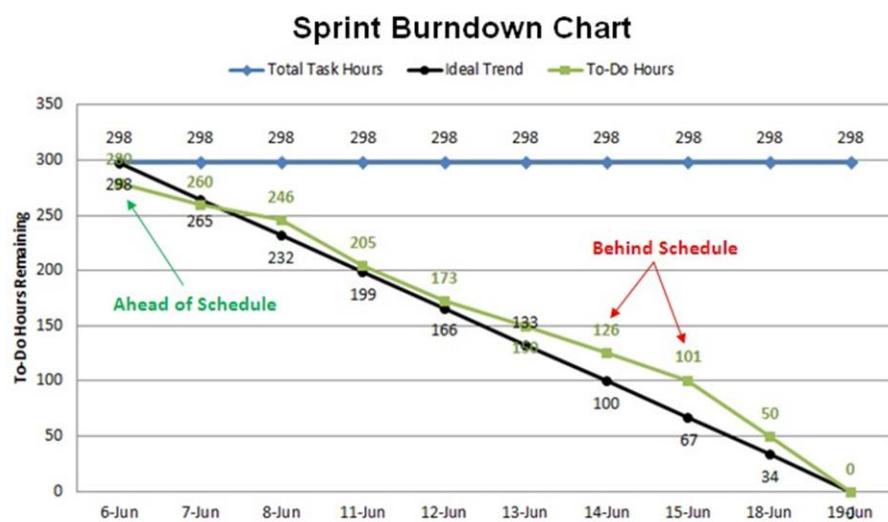


*PBIs – Product Backlog Items*

## Sprint Burndown Chart – Exemplo I



## Sprint Burndown Chart – Exemplo II



# Modelagem de Requisitos

Orientando-se pelo SAFe

# SAFe

Dean Leffingwell



**SAFe** é um acrônimo para **Scaled Agile Framework**.

Criado por Dean Leffingwell e, hoje, mantido pela [Scaled Agile Academy](#).

Orienta-se por princípios Lean e Agile.

É baseado em Scrum, XP (Extreme Programming) e o Lean, mas pensando em escalabilidade, desenvolvimento distribuído em várias equipes.

**Em resumo:** O que o Agile e o Scrum são para as equipes,  
o **SAFe** é para as empresas.

## SAFe

Por que falar de SAFe em Requisitos de Software?

- Basicamente, por se tratar de um framework que permitir escalar práticas ágeis em corporações ou grandes projetos.
- A natureza particular dessa proposta é a ênfase que o framework confere ao **gerenciamento de requisitos**. Este framework vem sendo elaborado desde 2009 por Dean Leffingwell, o mesmo autor de **Managing Software Requirements – A Unified Approach**. Essa é a base teórica para os cursos de gerenciamento de requisitos da IBM Rational.
- Dean Leffingwell foi um dos idealizadores do **IBM Rational RequisitePro**, ferramenta de gestão de requisitos que ficou famosa na década de 90. Hoje, já se tem evolução, em outras ferramentas...

## SAFe

Muitos adotam o SAFe, visando escalar o Scrum, trabalhando com várias equipes...

Dado o sucesso das metodologias ágeis, temos que SAFe tem ganhado popularidade...

Mas, é proprietário...

Portanto, é interessante conhecer....

Mas, não esquecer que existem outras propostas abertas... Essas podem ser tão interessantes quanto...

## SAFe

SAFe propõe um mecanismo definindo diferentes níveis de times, com **epic owners**, **business owners**, **product owners** e muitos outros envolvidos.

Para tanto, são utilizados os tradicionais **times Scrum**, para os níveis operacionais que tenham prazo de entrega; e **equipes Kanban**, para times de decisão focados em questões de prioridade.

A dinâmica de times varia considerando: a importância da prioridade versus previsão de entrega. Prioridade alta e prazo curto, equipes mais capacitadas, por exemplo.

# SAFe

Como o SAFe materializa essa variação na dinâmica da equipe?

Simples, definindo três níveis com dinâmicas diferentes, sendo: dinâmica das equipes (**teams**), dinâmica do programa ou projeto (**program**) e dinâmica do portfólio.

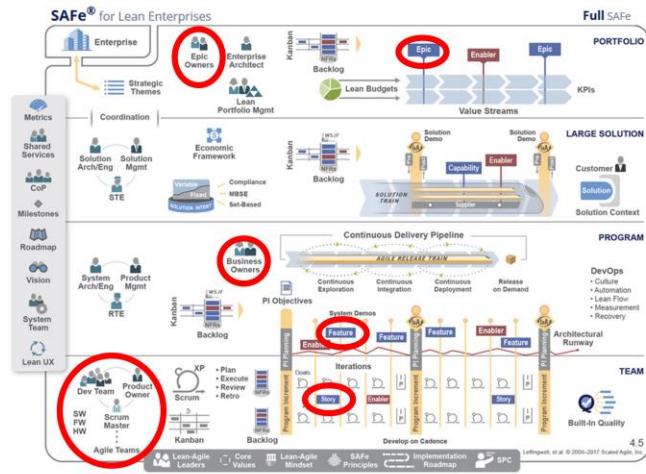
Esses três níveis de abstração organizacional possuem também três níveis de narrativa de requisitos.

Esquematicamente:

|                                 |                            |
|---------------------------------|----------------------------|
| Dinâmica de portfólio           | Temas ou Épicos            |
| Dinâmica de programa ou projeto | Características (Features) |
| Dinâmica de equipe              | Histórias de usuários      |

Existem várias versões do SAFe, as quais variam bastante...

## SAFe - Última Versão



Nível extra, para lidar com grandes desafios! Estudo de viabilidade, propostas de soluções (demo), e outros recursos são focos desse nível.

SAFe 4.5 apoia-se nas últimas vertentes de Lean-Agile thinking - <http://www.scaledagileframework.com/>  
Copyright © 2010-2017 Scaled Agile, Inc.: <http://www.scaledagile.com/permissions-form/>

Existem várias versões do SAFe, as quais variam bastante...

Existem várias versões do SAFe, as quais variam bastante...

## SAFe – Exemplo de Especificação I...

Fabrikam Stories

|            |              |  | Mapping | Off | Parents       | Show |
|------------|--------------|--|---------|-----|---------------|------|
| New        | Work Item... | Title  |         |     |               |      |
| Epic       | Epic         | > Improve User Experience                            |         | New | Business      |      |
| Epic       | Epic         | > Framework to port applications to all devices      |         | New | Architectural |      |
| Epic       | Epic         | △ Counter the Heartbleed web security bug            |         | New | Architectural |      |
| Feature    | Feature      | △ Scan all software for the OpenSSL cryptographic... |         | New | Architectural |      |
| User Story |              | Scan all code base and identify the affected c...    |         | New | Business      |      |
| User Story |              | Scan all code base and identify the affected c...    |         | New | Business      |      |
| Feature    | Feature      | △ Update software to resolve the OpenSSL cryptog...  |         | New | Architectural |      |
| User Story |              | Update and re-test suite code base affected b...     |         | New | Business      |      |
| User Story |              | Update and re-test service code base affected...     |         | New | Business      |      |
| Epic       | Epic         | > Shared Personalization and State                   |         | New | Business      |      |
| Epic       | Epic         | △ Service interfaces to support Rest API             |         | New | Architectural |      |
| Feature    | Feature      | △ Convert legacy OData service interfaces to Rest... |         | New | Architectural |      |
| User Story |              | Convert all services from using experimental...      |         | New | Business      |      |
| User Story |              | Convert all client service calls from using expe...  |         | New | Business      |      |

Existem várias versões do SAFe, as quais variam bastante...

Conter um bug de segurança chamado Heartbleed

Procurar a biblioteca defasada do OpenSSL

Procurar por todo o código e identificar ...

Atualizar o software para resolver os problemas da biblioteca defasada do OpenSSL.

Existem várias versões do SAFe, as quais variam bastante...

## SAFe – Exemplo de Especificação II...

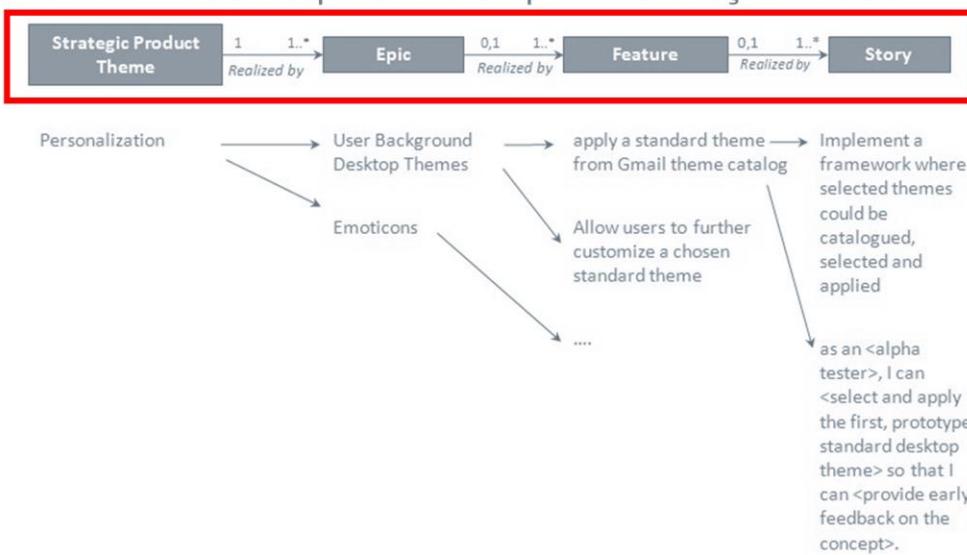
The screenshot shows a backlog management interface for a Customer Service project. The left sidebar has tabs for Backlogs, Queries, Epics, Features, and Backlog items, with Backlog items selected. The main area displays a list of backlog items under the heading "Customer Service Backlog items". The items are organized by type: Epic, Feature, Product Backlog Item, Bug, Product Backlog Item, Feature, Product Backlog Item, and Product Backlog Item. A single "Bug" item is highlighted with a red rectangular box. The columns show the Work Item Type, Title, and Area Path for each item. The "Bug" item has the title "Switch context issues" and the area path "Scrum\Customer Service".

| Work Item Type       | Title                                     | Area Path                     |
|----------------------|---|-------------------------------|
| Epic                 | Customer service - improve UI performance | Scrum                         |
| Feature              | Customer Service - Phone                  | Scrum\Customer Service        |
| Product Backlog Item | Check service status                      | Scrum\Customer Service        |
| <b>Bug</b>           | <b>Switch context issues</b>              | <b>Scrum\Customer Service</b> |
| Product Backlog Item | GPS locator interface                     | Scrum\Customer Service        |
| Feature              | Customer Service - Web                    | Scrum\Customer Service        |
| Product Backlog Item | Technician dashboard improvements         | Scrum\Customer Service        |
| Product Backlog Item | Scheduler                                 | Scrum\Customer Service        |

Existem várias versões do SAFe, as quais variam bastante...

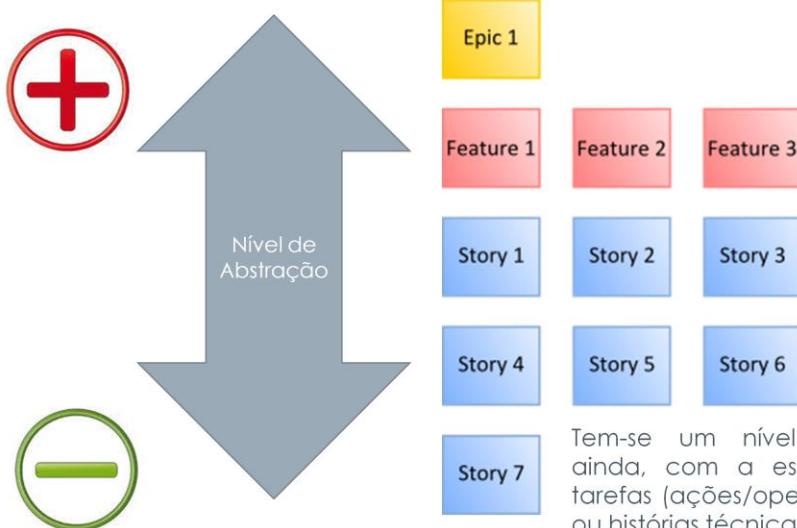
Existem várias versões do SAFe, as quais variam bastante...

## SAFe – Exemplo de Especificação III...



Existem várias versões do SAFe, as quais variam bastante...

## SAFe – Em resumo...

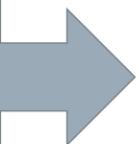


# Debate

## Debate

Por isso, existe a necessidade de complementar essa especificação para especificar mais adequadamente os requisitos não funcionais...

A preocupação chave é operacionalizar logo, para que a equipe de desenvolvimento gere os entregáveis...  
Pensar em Requisitos Não Funcionais é de SUMA RELEVÂNCIA, mas nessas metodologias, fica pouco enfatizada essa necessidade.



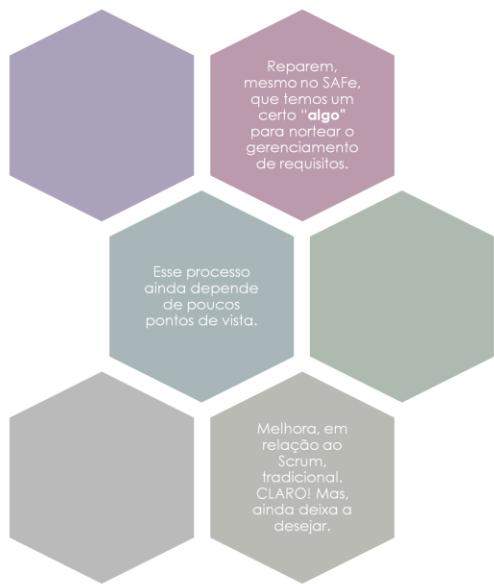
Requisitos não Funcionais...

RNF

## Debate

Novamente, por isso,  
existe a necessidade  
de complementar  
essa especificação ...

Investigar pontos de vista diferentes  
é de SUMA RELEVÂNCIA!!!  
Lembram do caso da ambulância  
de Londres? : (



Pontos de Vista diferentes...

Pontos de Vista diferentes...

## Debate

Novamente, por isso,  
existe a necessidade  
de complementar  
essa especificação ...

Uma matriz de rastreabilidade  
poderia ajudar. Iremos falar sobre  
pós-rastreabilidade mais adiante.  
Mas, a pré-rastreabilidade não é  
atendida apenas utilizando o SAFe.  
Ok?

Reparam, que o  
SAFe, orientado  
ao  
gerenciamento  
de requisitos, ...

até permite  
manter um elo  
entre os *epics*,  
*features*, e  
*stories*.

Mas, não tem a  
preocupação  
em manter a  
rastreabilidade.

Rastreabilidade...

Rastreabilidade...

## Mas...



As metodologias ágeis sabem que não são feitas para todo e qualquer tipo de desenvolvimento...



Nas documentações das metodologias ágeis, têm sempre observações sobre qual é o perfil de projeto para o qual a metodologia é adequada...



Requisitos não totalmente desconhecidos...

Requisitos em evolução...

Sabe-se, mais ou menos, onde se quer chegar...

O domínio cognitivo não é de todo desconhecido...

Não são utilizadas para tudo...

Têm suas aplicabilidades... bem como estabelecem o perfil de projeto ótimo para aplicá-las...

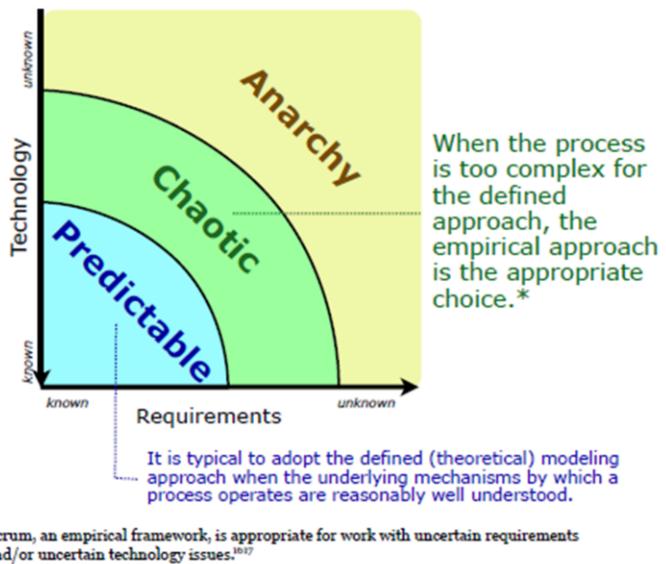
Não são utilizadas para tudo... Têm suas aplicabilidades... bem como estabelecem o perfil de projeto ótimo para aplicá-las...

Reparem...

Incerto... Caótico...  
Mas, não anárquico e  
desconhecido por  
completo...  
Portanto, tem-se  
conhecimento, mesmo  
que pequeno e sujeito a  
mudanças...

FONTE: Scrum Reference Card  
by Michael James and Luke  
Walter for CollabNet, Inc.

## When is Scrum Appropriate?



Incerto... Caótico... Mas, não anárquico e desconhecido por completo...  
Tem-se conhecimento, mesmo que pequeno e sujeito a mudanças...

# Considerações Finais

## Considerações Finais

- › Nessa aula, foi apresentada a atividade de modelagem de requisitos com base em artefatos de abordagens ágeis, mais especificamente na metodologia **Scrum** e no framework **SAFe**.
- › No caso, focou-se em:
  - *Product Backlog* e suas particularidades;
  - *Sprint Backlog* e suas particularidades, e
  - no trio: *Epics*, *Features* e *User Stories*.
- › Continuem os estudos!



# Referências

# Referências

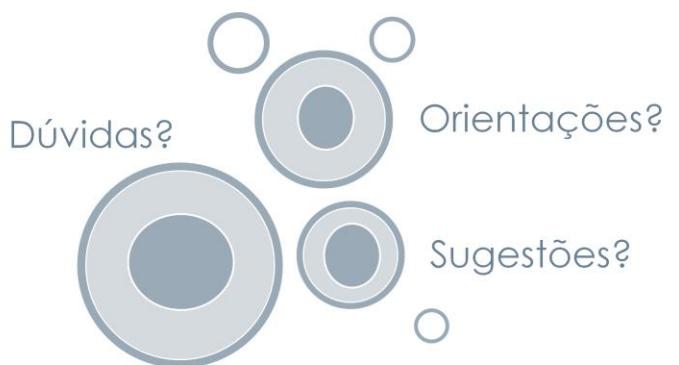
## Bibliografia Básica

1. [Elibrary] Young, Ralph. Requirements Engineering Handbook. Norwood, US: Artech House Books, 2003.
2. [Open Access] Leite, Julio Cesar Sampaio do Prado. Livro Vivo - Engenharia de Requisitos. <http://livrodeengenhariaderequisitos.blogspot.com.br/> (último acesso: 2017)
3. [Elibrary] Chemuturi, Murali. Mastering Software Quality Assurance : Best Practices, Tools and Technique for Software Developers. Ft. Lauderdale, US: J. Ross Publishing Inc., 2010.
4. Software & Systems Requirements Engineering: In Practice - Brian Berenbach, Daniel Paulish, Juergen Kazmeier, Arnold Rudorfer (Livro bem completo mas, não tem exemplar físico na biblioteca, nem mesmo consta na Elibrary)
5. Requirements Engineering and Management for Software Development Projects - Murali Chemuturi (Livro bem completo mas, não tem exemplar físico na biblioteca, nem mesmo consta na Elibrary)

# Referências

## Bibliografia Complementar

1. [BIBLIOTECA – 15 exemplares] Pfleeger, Shari Lawrence. Engenharia de Software: Teoria e Prática. 2º. Edição. São Paulo: Prentice Hall, c2004. xix, 535 p. ISBN 978858791831
2. [BIBLIOTECA – 3 exemplares] Withall, Stephen. Software Requirement Patterns. Redmond: Microsoft Press, c2007. xvi, 366 p. ISBN 978735623989.
3. [BIBLIOTECA - vários exemplares] Leffingwell, 2011, Agile Software Requirements, <http://www.scaledagileframework.com/> (último acesso: 2017)
4. [Ebrary] Evans, Isabel. Achieving Software Quality Through Teamwork. Norwood, US: Artech House Books, 2004.
5. [Ebrary] Yu, Eric, Giorgini, Paolo, and Maiden, Neil, eds. Cooperative Information Systems: Social Modeling for Requirements Engineering. Cambridge, US: MIT Press, 2010.
6. [Open Access] Slides disponíveis <https://www.wou.edu/~eltonm/Marketing/PP%20Slides/> (último acesso: 2017) em:



FIM

[mileneserrano@unb.br](mailto:mileneserrano@unb.br) ou [mileneserrano@gmail.com](mailto:mileneserrano@gmail.com)  
[serrano@unb.br](mailto:serrano@unb.br) ou [serr.mau@gmail.com](mailto:serr.mau@gmail.com)