



# Certified Professional for Requirements Engineering

RE@Agile  
Syllabus

Stefan Gärtner, Peter Hruschka,  
Markus Meuten, Gareth Rogers,  
Hans-Jörg Steffe



## Termos de Uso:

1. Indivíduos e provedores de treinamento podem utilizar este syllabus como base para cursos, desde que os direitos autorais sejam reconhecidos e incluídos no material do curso. Qualquer pessoa que utilize este syllabus em publicidade precisa do consentimento por escrito do IREB e.V. para esse fim.
2. Qualquer indivíduo ou grupo de indivíduos poderá utilizar este Syllabus como base para artigos, livros ou outras publicações derivadas, desde que tais publicações reconheçam e citem os autores do presente documento e o IREB e.V. como fonte e detentor dos direitos autorais do mesmo.

© IREB e.V.

Todos os direitos reservados. Nenhuma parte desta publicação pode ser reproduzida, armazenada em um sistema de arquivamento ou transmitida de qualquer forma, ou por qualquer meio, seja eletrônico, mecânico, fotocópia, ou gravação ou qualquer outro, sem a autorização prévia e por escrito dos autores ou do IREB e.V.

## Agradecimentos

A versão atual deste syllabus foi elaborada por: Stefan Gärtner, Peter Hruschka, Kim Lauenroth, Markus Meuten, Gareth Rogers e Hans-Jörg Steffe.

Revisão por Rainer Grau. Os comentários de revisão foram fornecidos por Jan Jaap Cannegieter, Andrea Herrmann, Uwe Valentini e Sven van der Zee. Traduzido para a Língua Portuguesa por Ana Moreira, Carlos Silva, George Fialkovitz, Guilherme Siqueira Simões.

Aprovado para liberação em 25 de outubro de 2024 pelo Conselho do IREB por recomendação de Xavier Franch.

Agradecemos a todos pelo seu envolvimento.

Os direitos autorais © 2017–2025 deste syllabus são dos autores listados acima. Os direitos foram transferidos para o IREB International Requirements Engineering Board e.V.

## Objetivo do documento

Este syllabus define os certificados para o RE@Agile Practitioner e para o RE@Agile Specialist do International Requirements Engineering Board (IREB). O syllabus fornece aos provedores de treinamento a base para a criação de seus materiais de curso. Os alunos podem usar o syllabus para ter uma visão geral do conteúdo pretendido e dos objetivos de aprendizagem. Os detalhes do conteúdo para a preparação do material do curso e para o exame podem ser encontrados no "Handbook RE@Agile, Education and Training for the IREB Certified Professional for Requirements Engineering RE@Agile Practitioner | Specialist".

## Conteúdo do Syllabus

O módulo RE@Agile destina-se a pessoas das áreas de Engenharia de Requisitos, Análise de Negócios, Engenharia de Negócios, Desenvolvimento de Software e de Sistemas, bem como a funções existentes na comunidade ágil (Product Owner, Scrum Master, Desenvolvedor) que gostariam de aprofundar seus conhecimentos sobre como lidar com requisitos.

## Nível de detalhe

O nível de detalhamento deste Syllabus possibilita uma consistência de cursos e avaliações em âmbito internacional. Para atingir este objetivo, o syllabus contém o seguinte:

- Objetivos educacionais gerais;
- Conteúdo com uma descrição dos objetivos educacionais, e
- Referências a outras literaturas (quando necessário).

## Objetivos Educacionais/ Níveis de Conhecimento Cognitivo

A todos os módulos e objetivos educacionais deste syllabus é atribuído um nível cognitivo. Os seguintes níveis são usados:

- **L1: Conhecer** (identificar, lembrar, recuperar, recordar, reconhecer) – O candidato reconhecerá, lembrará e recordará um termo ou conceito.
- **L2: Compreender** (resumir, generalizar, abstrair, classificar, comparar, mapear, contrastar, exemplificar, interpretar, traduzir, representar, inferir, concluir, categorizar, construir modelos) – O candidato pode selecionar as razões ou explicações para declarações relacionadas ao tópico e pode resumir, comparar, classificar, categorizar e dar exemplos do conceito.
- **L3: Aplicar** (implementar, executar, usar, seguir um procedimento, aplicar um procedimento) – O candidato pode selecionar a aplicação correta de um conceito ou técnica e aplicá-la a um determinado contexto.
- **L4: Analisar** (analisar, organizar, encontrar coerência, integrar, delinear, analisar, estruturar, atribuir, desconstruir, diferenciar, discriminar, distinguir, focar, selecionar) – O candidato pode separar informações relacionadas a um procedimento ou técnica em suas partes constituintes para melhor compreensão e pode distinguir entre fatos e inferências. A aplicação típica é analisar um documento, software ou situação de projeto e propor ações apropriadas para resolver um problema ou tarefa.
- **L5: Avaliar** (criticar, julgar) – O candidato pode fazer uma crítica bem argumentada de um determinado artefato e fazer um julgamento profundo em um determinado caso.

Observe que um objetivo de aprendizagem no nível cognitivo de conhecimento Ln também contém os elementos de todos os níveis cognitivos abaixo dele (L1 to Ln-1).

Exemplo: Um objetivo de aprendizagem do tipo "Aplicar a técnica de ER xyz" está no nível de conhecimento cognitivo (L3). Entretanto, a capacidade de aplicar requer que os alunos conheçam a técnica da ER xyz (L1) e que entendam para que serve esta técnica (L2).



Todos os termos definidos no glossário devem ser conhecidos (L1), mesmo que não sejam mencionados explicitamente nos objetivos educacionais.

O glossário está disponível para download no site do IREB em <https://www.ireb.org/en/downloads/#cpre-glossary-2-0>

Este syllabus e o respectivo handbook usam a abreviação "ER" para Engenharia de Requisitos.

### Estrutura do syllabus

O syllabus consiste em seis capítulos principais. Um capítulo abrange uma unidade educacional (EU). O título dos capítulos contém o nível cognitivo de seus capítulos, que é o nível mais alto de seus subcapítulos. O tempo de ensino sugerido é o mínimo que um curso deve investir para esse capítulo. As empresas de treinamento têm liberdade para dedicar mais tempo às UEs e aos exercícios. Entretanto, eles devem garantir que o tempo necessário seja mantido em relação às outras UEs. Os termos importantes utilizados em cada capítulo estão listados no seu início.

Exemplo: UE2 Um Início de Projeto Limpo (L2)  
Duração: 120 minutos + 60 minutos de exercícios  
Termos: Visão do produto, Objetivo do produto, Parte interessada, Persona, Escopo do produto, Limite do sistema

Esse exemplo mostra que o capítulo 2 contém objetivos educacionais no nível L2, e 180 minutos são destinados ao ensino do material desse capítulo.

Cada capítulo pode conter subcapítulos. Seus títulos também contêm o nível cognitivo de seu conteúdo.

Os objetivos educacionais (EO) são enumerados. A numeração mostra a que subcapítulo eles pertencem.

Exemplo: EO 3.3.1 Ser capaz de aplicar os critérios INVEST ao redigir requisitos (L3)

Esse exemplo mostra que o objetivo educacional EO 3.3.1 está descrito no subcapítulo 3.3 e que o nível cognitivo L3 é esperado.

## 0 exame

Este syllabus abrange as unidades educacionais e os objetivos educacionais para os exames de certificação do

- CPRE RE@Agile – Practitioner
- CPRE RE@Agile – Specialist

O exame para obter o certificado CPRE RE@Agile – Practitioner – consiste em um **exame de múltipla escolha**.

O exame para obter o certificado CPRE RE@Agile – Specialist – consiste em um **trabalho escrito**.

Ambos os exames incluem perguntas sobre todas as unidades educacionais e todos os objetivos educacionais do syllabus.

Cada pergunta do exame pode incluir material de vários capítulos do syllabus, bem como de vários objetivos de aprendizado ou até mesmo de partes de um objetivo de aprendizado.

### O exame de múltipla escolha para o certificado Practitioner

- testa todos os objetivos de aprendizado do syllabus. No entanto, para os objetivos educacionais nos níveis de conhecimento cognitivo L4 e L5, as questões do exame estão limitadas a itens nos níveis cognitivos L1 a L3.
- pode ser realizado imediatamente após um curso de treinamento, mas também de forma independente dos cursos (p. ex., à distância ou em um centro de exames).

### O trabalho escrito para o certificado de Specialist

- testa todos os objetivos educacionais do syllabus nos níveis de conhecimento cognitivo indicados para cada objetivo de aprendizado.
- segue a descrição da tarefa para RE@Agile – Specialist –, encontrada em <https://cpre.ireb.org/en/downloads-and-resources/downloads#cpre-re-agile-specialist-written-assignment>.
- é individualizado e enviado a um órgão de certificação licenciado.

### Os seguintes objetivos genéricos de aprendizagem também se aplicam ao trabalho escrito para o certificado de Especialista:

EO G1: Analisar e ilustrar problemas do RE@Agile usando um contexto familiar ao candidato ou um que se assemelhe muito a ele (L4).

EO G2: Avaliar e refletir sobre o uso de práticas, métodos, processos e ferramentas RE@Agile em projetos nos quais o candidato esteve envolvido (L5).

A lista de organizações de certificação licenciadas pelo IREB encontra-se no site:

<https://www.ireb.org>

### Histórico de versões

Versão	Data	Comentário
2.0.0	30 de Março de 2023	Versão Inicial baseada no Syllabus original do IREB em inglês
2.1.0	30 de Maio de 2024	Implementação do novo design corporativo, sincronização dos níveis de conhecimento cognitivo, eliminação do termo "Advanced Level"
2.2.0	24 de Maio de 2024	Níveis de conhecimento cógnitivo corrigidos
2.3.0	1º de Outubro de 2025	Objetivos educativos atualizados e complementados Breve descrição de capítulos re-escrita Ilustrações adicionadas aos capítulos

# Conteúdo

Conteúdo .....	7
1 O que é RE@Agile (L2) .....	9
2 Um Início de Projeto Limpo (L3) .....	11
2.1 Visão e objetivos (L3).....	11
2.2 Especificando o limite do sistema (L3) .....	12
2.3 Identificação e Gestão de Stakeholders (L3) .....	12
2.4 Dependências entre visões/objetivos, stakeholders e o limite do sistema (L3).....	13
3 Lidando com Requisitos Funcionais (L4) .....	14
3.1 Diferentes Níveis de Granularidade de Requisitos .....	14
3.2 Comunicação e documentação em diferentes níveis .....	15
3.3 Trabalhando com histórias de usuários e itens de backlog .....	16
3.4 Técnicas de Divisão e Agrupamento .....	17
3.5 Saber quando parar.....	18
3.6 Documentação do Projeto e do Produto de Requisitos .....	18
4 Lidando com Requisitos de Qualidade e Restrições (L3) ...	20
4.1 Entendendo a importância dos requisitos e restrições de qualidade (L2).....	20
4.2 Acrescentar precisão aos requisitos de qualidade (L2) .....	20
4.3 Requisitos de qualidade e backlog (L3) .....	21
4.4 Tornando as Restrições Explícitas (L2) .....	22
5 Priorização e Estimativa dos Requisitos (L3) .....	24
5.1 Determinação do Valor de Negócio .....	24

5.2	Valor de Negócio, Riscos e Dependências (L3) .....	25
5.3	Expressando Prioridades e Organizando o Backlog .....	25
5.4	Estimativa de itens do backlog (L3) .....	26
5.5	Escolha de uma estratégia de desenvolvimento (L2) .....	29
6	Escalando RE@Agile (L2) .....	30
6.1	Escalando Requisitos e Equipes (L2) .....	31
6.2	Critérios para estruturar Requisitos e Equipes em Grande Escala (L2) .....	31
6.3	Roadmaps e Planejamento em Grande Escala (L2) .....	33
6.4	Validação do produto (L2) .....	35
7	Definições de termos, glossário .....	37
8	Referências .....	38

# 1 O que é RE@Agile (L2)

Duração: 45 minutos

Termos: Stakeholders, Product Owner, Cooperativo, Iterativo, Incremental

## Objetivos Educacionais

- EO 1.1 Conhecer a definição de RE@Agile (L1)
- EO 1.2 Entender os objetivos do RE@Agile (L2)
- EO 1.3 Compreender que a responsabilidade por bons requisitos é do product owner (L2)

## Conteúdo

Com base em suas respectivas histórias, as abordagens de ER e ágil geralmente são consideradas separadamente e não em conjunto. Isso geralmente leva ao mal-entendido de que há duas formas de ER: a ER clássica e a ER ágil. Os autores acreditam que só existe ER boa ou ruim – em um mundo não ágil ou ágil. Por isso, chamamos a abordagem de RE@Agile.

Ágil e ER são duas disciplinas com origens diferentes e objetivos distintos que, no entanto, podem aprender muito uma com a outra. No RE@Agile Primer [Prim2017] concluímos: "O valor mais importante é compartilhado pelo ER e pelo ágil, e é fazer com que o usuário do produto fique feliz porque a solução atende às suas necessidades ou cura suas maiores dores."

RE@Agile é uma abordagem cooperativa, iterativa e incremental com quatro objetivos:

1. Conhecer os requisitos relevantes em um nível adequado de detalhes (a qualquer momento durante o desenvolvimento do sistema).
2. Obtenção de acordo suficiente sobre os requisitos entre os stakeholders relevantes.
3. Capturar (e documentar) os requisitos de acordo com as restrições da organização.
4. Realizar todas as atividades relacionadas aos requisitos de acordo com os princípios do Manifesto Ágil.

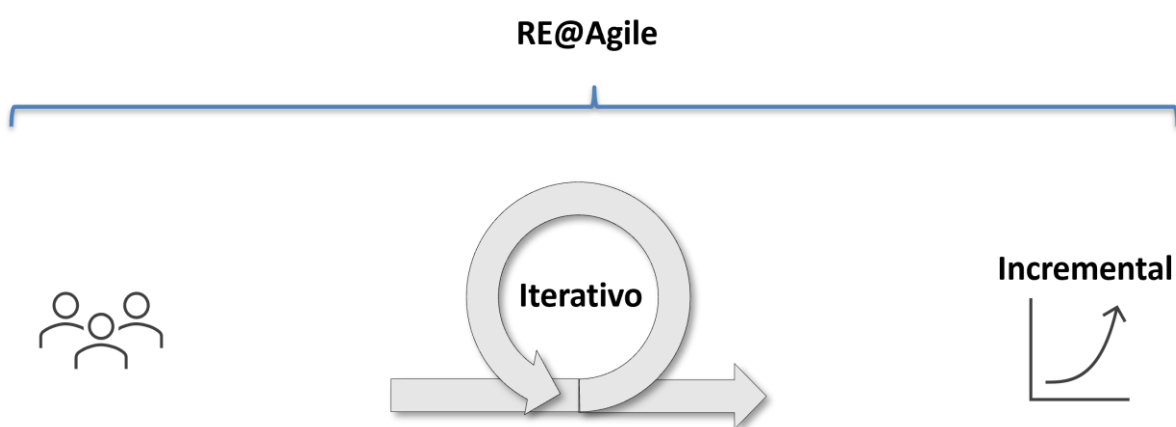


Figura 1: Abordagem cooperativa, iterativa e incremental

Nas abordagens ágeis, diferentes funções fornecem requisitos para o sistema desejado. Independentemente de quem fornece esses requisitos, de quem os estrutura e de quem os

detalha, a pessoa com a função/responsabilidade de product owner continua sendo responsável pelos ER.

## 2 Um Início de Projeto Limpo (L3)

Duração: 120 minutos + 60 minutos de exercícios

Termos: Visão, Objetivo, Parte interessada, Limite do sistema, Diagrama de contexto, Diagrama de caso de uso

### Conteúdo

Mesmo em abordagens ágeis, alguns pré-requisitos importantes devem ser estabelecidos antes que o trabalho de desenvolvimento iterativo e incremental do sistema possa ser iniciado com sucesso.

- Definição da visão e/ou objetivos do sistema
- Identificação do escopo atualmente conhecido do sistema e do limite do sistema
- Identificação dos stakeholders relevantes e outras fontes de requisitos importantes

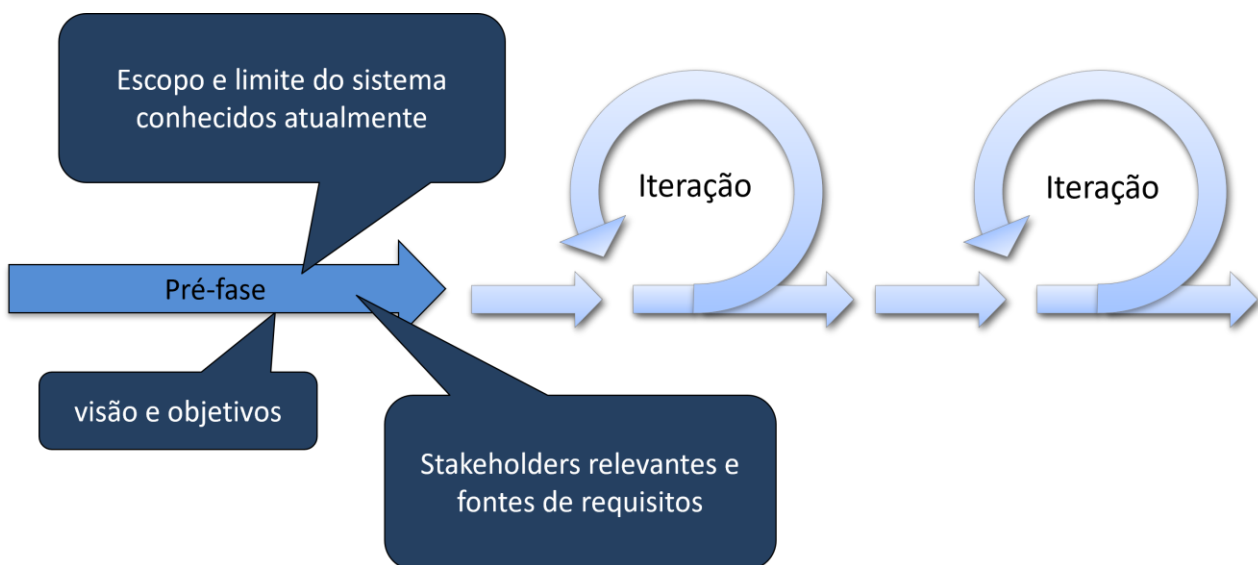


Figura 2: Um início de projeto limpo

### 2.1 Visão e objetivos (L3)

#### Objetivos Educacionais

EO 2.1.1 Aplicar a especificação de objetivos e visões (L3)

#### Conteúdo

A visão do sistema ou visão do produto descreve o objetivo geral que deve ser alcançado com o sistema/produto. A visão é de suma importância para cada atividade de desenvolvimento. Ela define a pedra angular e serve como uma direção geral para todas as atividades de desenvolvimento. Cada requisito deve ajudar a alcançar a visão do sistema.

Abordagens alternativas para a formulação de objetivos ou visões são:

- SMART [Dora1981]
- PAM [Robe2003]

- Product Vision Box [High2001]
- News from the future [HeHe2011]
- Quadros de Visão (Vision Boards)
- Técnicas de Canvas

## 2.2 Especificando o limite do sistema (L3)

### Objetivos Educacionais

EO 2.2.1 Aplicar a especificação do limite do sistema (L3)

### Conteúdo

Um entendimento compartilhado e comum do escopo e do contexto do sistema é um pré-requisito para um esforço de desenvolvimento eficaz e eficiente [Glin2024].

O escopo e o limite do sistema podem ser documentados e esclarecidos com várias técnicas, como:

- Diagramas de contexto
- Linguagem natural
- Diagramas de casos de uso
- Mapas de histórias

## 2.3 Identificação e Gestão de Stakeholders (L3)

### Objetivos Educacionais

EO 2.3.1 Dominar e usar a identificação e o gerenciamento das stakeholders (L3)

### Conteúdo

Da mesma forma que as abordagens tradicionais, os stakeholders mais importantes também devem ser identificadas no início do processo ágil, para que seja definida uma estrutura para a elicitación de requisitos. No entanto, na agilidade, os stakeholders podem e vão mudar constantemente. Portanto, é essencial que a identificação e a definição das stakeholders sejam entendidas como um processo cooperativo, incremental e, acima de tudo, iterativo (consulte também o capítulo 1).

Deixar de identificar e envolver uma parte interessada importante no processo de desenvolvimento pode ter um grande impacto. Se os requisitos de tal parte interessada forem identificados tarde demais (ou não forem identificados), isso poderá resultar em mudanças dispendiosas ou até mesmo em um sistema inútil [PoRu2021].

O Onion Model de Ian Alexander [Alex2005] é uma ferramenta simples para identificação e classificação de stakeholders. O modelo consiste em três tipos de stakeholders (camadas de uma cebola) que podem ser pesquisadas sistematicamente:

- Stakeholders do sistema
- Stakeholders do contexto circundante
- Stakeholders do ambiente mais amplo

Como regra geral, a identificação dos stakeholders deve basear-se em uma ampla gama de fontes.

Dependendo do sistema e do domínio, a documentação existente, os sistemas vizinhos com interfaces para o sistema desenvolvido, os sistemas legados ou até mesmo os sistemas concorrentes também podem ser fontes importantes (além dos stakeholders) de requisitos.

## 2.4 Dependências entre visões/objetivos, stakeholders e o limite do sistema (L3)

### Objetivos Educacionais

EO 2.4.1 Aplicar a mudança dinâmica de visão e objetivos, stakeholders e escopo do sistema (L3)

### Conteúdo

As definições de visão e objetivos, bem como os stakeholders e os limites do sistema são aqui interdependentes [PoRu2021]:

- Os stakeholders relevantes formulam a visão e os objetivos. Portanto, a identificação de um novo stakeholder relevante pode ter um impacto sobre a visão e os objetivos.
- A visão e os objetivos podem ser usados para orientar a identificação de novos stakeholders por meio de perguntas: Qual stakeholder pode estar interessado em alcançar a visão e os objetivos ou é afetado por eles?
- A visão e os objetivos podem ser usados para definir um escopo inicial por meio de perguntas: Quais elementos são necessários para alcançar a visão e os objetivos?
- Alterar o limite do sistema (e, portanto, o escopo) pode ter um impacto sobre a visão e os objetivos. Se um aspecto for removido do escopo, será necessário verificar se o sistema ainda tem meios suficientes para atingir a visão e os objetivos.
- Os stakeholders definem os limites do sistema. Portanto, a identificação de um novo stakeholder relevante pode ter um impacto sobre o escopo.
- Uma mudança no escopo (p. ex., para cumprir uma meta) exige a concordância dos stakeholders relevantes.

Essas interdependências devem ser usadas para equilibrar todos os três elementos (visão e objetivos, stakeholders, escopo) para examinar o impacto da alteração de um dos três elementos sobre o outro.

Devido a essas dependências entre a visão e os objetivos, os stakeholders e o escopo, recomendamos tratar todos esses elementos juntos e de forma coerente.

## 3 Lidando com Requisitos Funcionais (L4)

Duração: 195 minutos + 120 minutos de exercícios

Termos: Requisito funcional, Épico, Recurso, História, História do usuário, Definição de Pronto (DoR), INVEST, Níveis de granularidade

### Conteúdo

Esse núcleo da UE tem uma visão estática dos requisitos funcionais, ou seja, estrutura um grande conjunto de requisitos em hierarquias de abstração.

Assim que se aceita a ideia de que os requisitos existem em diferentes níveis de granularidade, surgem naturalmente algumas perguntas:

- Como lidamos com múltiplos níveis de granularidade?
- Quais critérios podem e devem ser aplicados para dividir tópicos grandes e abstratos em partes menores?
- Às vezes é necessário agrupar muitos requisitos pequenos em partes maiores para que tenhamos um "quadro geral" para orientação?
- Quão precisos temos que ser antes que os desenvolvedores possam começar com a implementação?
- É necessário ou aconselhável manter vários níveis de requisitos, ou podemos jogar fora declarações abstratas assim que tivermos requisitos mais concretos?
- Preferimos estruturar o backlog de acordo com as relações/processos funcionais ou de acordo com outras relações, como contextos técnicos?
- Temos que capturar tudo isso por escrito ou podemos simplesmente falar sobre isso?

### 3.1 Diferentes Níveis de Granularidade de Requisitos

#### Objetivos Educacionais

EO 3.1.1 Conhecer a existência de requisitos funcionais em diferentes níveis de granularidade (L1)

#### Conteúdo

Os stakeholders geralmente comunicam os requisitos em diferentes níveis de granularidade. Algumas vezes eles pedem grandes pedaços de funcionalidade, outras vezes eles pedem detalhes menores para serem acrescentados ou alterados.

Épicos (às vezes também chamados de temas) ou histórias grandes (que representam processos de negócio potencialmente complexos) são uma boa maneira de obter uma visão geral de todos os requisitos conhecidos atualmente no backlog, ou seja, uma visão geral de tudo o que os stakeholders esperam de um sistema ou produto. Os épicos são decompostos em elementos mais refinados, geralmente recursos e histórias. As histórias fornecem o nível mais refinado de granularidade e podem comunicar requisitos de diferentes perspectivas, por exemplo, perspectiva do usuário, perspectiva técnica.

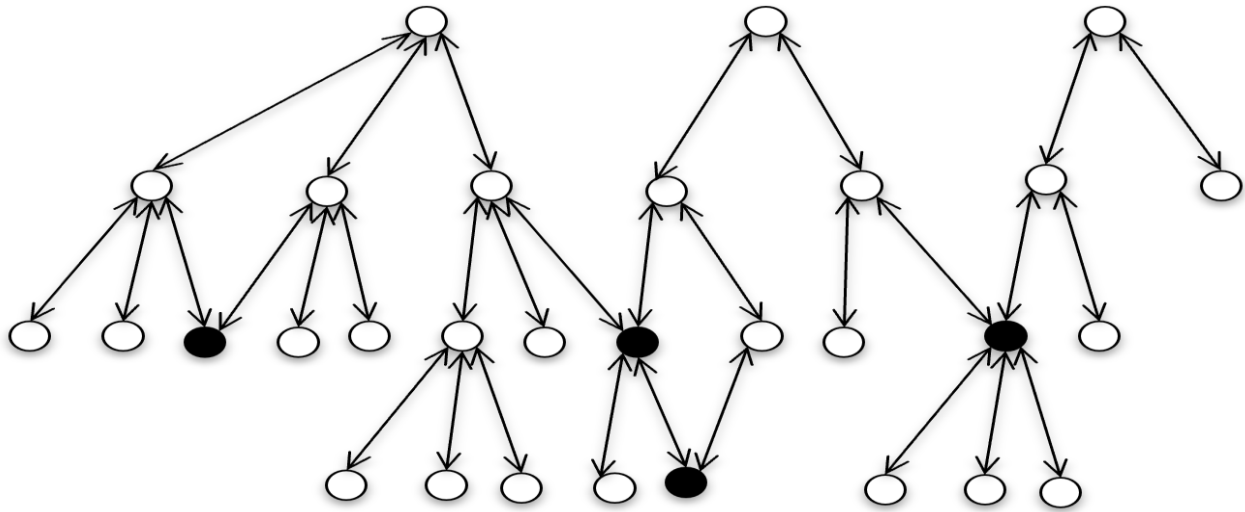


Figura 3: Granularidade dos requisitos

## 3.2 Comunicação e documentação em diferentes níveis

### Objetivos Educacionais

- EO 3.2.1 Analisar e aplicar a decomposição de requisitos no nível mais alto (L4)
- EO 3.2.2 Analisar e aplicar diferentes estratégias de decomposição em grande escala (L4)
- EO 3.2.3 Analisar e aplicar a identificação, documentação e comunicação de requisitos funcionais em diferentes níveis de granularidade (L4)

### Conteúdo

Com base no princípio de "dividir para conquistar", precisamos decompor um sistema ou produto grande em partes menores.

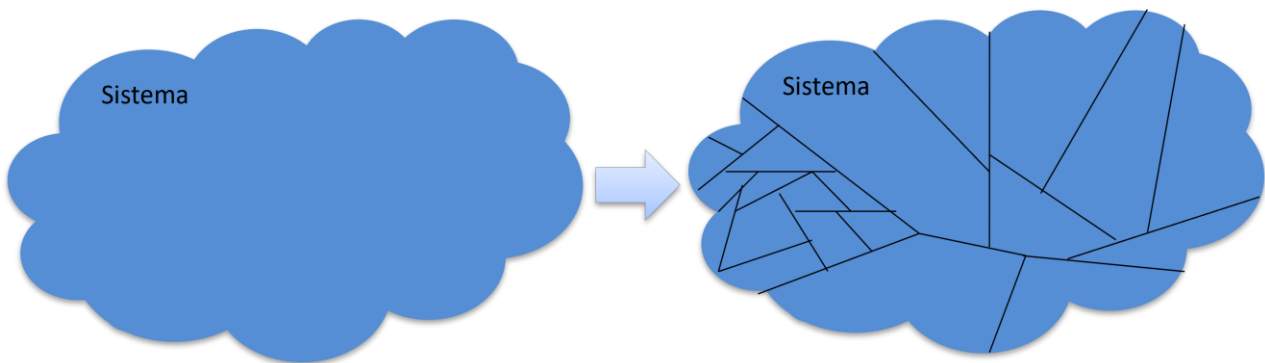


Figura 4: Decomposição do sistema em partes menores

Há diferentes abordagens para atingir esse objetivo (p. ex., divisão em funções lógicas, divisão de acordo com aspectos organizacionais, divisão de acordo com o hardware, divisão de acordo com os dados, etc.). Ao fazer isso, analisamos a abordagem para dividir os requisitos em processos de criação de valor iniciados externamente.

Além de decompor os requisitos em partes menores, também precisamos nos preocupar com a comunicação e a documentação dos requisitos funcionais. A opção básica é entre

desenhar (p. ex., com modelos) e escrever (p. ex., em linguagem natural). Acima de tudo, cabe a toda a equipe decidir qual é o formato preferido. Por exemplo, a visão geral pode ser fornecida por meio de um diagrama de caso de uso, enquanto as etapas do processo dos casos de uso são descritas em palavras simples (p. ex., como uma história de usuário).

### 3.3 Trabalhando com histórias de usuários e itens de backlog

#### Objetivos Educacionais

- EO 3.3.1 Aplicar a ideia de Pensamento Centrado no Usuário nos requisitos com Histórias de Usuário (L3)
- EO 3.3.2 Criar requisitos com base na ideia da história do usuário e usando o modelo de história do usuário (L3)
- EO 3.3.3 Aplicar os critérios INVEST ao redigir requisitos (L3)

#### Conteúdo

As histórias de usuários se tornaram muito importantes no ambiente ágil porque apontam para a importância de estruturar e desenvolver requisitos e os produtos resultantes de uma forma centrada no usuário. Em outras palavras, colocar o usuário em primeiro lugar. Essa perspectiva do usuário é importante porque desenvolvemos principalmente de forma ágil para receber feedback regular dos usuários.

É importante observar que, no ambiente ágil, o termo "história do usuário" é frequentemente usado com três finalidades diferentes:

- Como uma forma orientada ao usuário para discutir e formular requisitos.
- Como o nível mais baixo (mais detalhado) de estruturação de requisitos (Épicos -> Recursos -> Histórias (de usuário)). Na verdade, o termo "história do usuário" é frequentemente usado em vez de "história" na estrutura de requisitos.
- Como um modelo para descrever os requisitos em diferentes níveis hierárquicos a partir da perspectiva do usuário.

### Como uma estrutura formal (modelo)

(Possível em todos os níveis hierárquicos)

"Como <função/pessoa>, quero  
<objetivo/desejo> para que  
<benefício>"

Figura 5: Modelo de história de usuário

Neste syllabus, para atenuar essa ambiguidade, usamos o termo "história" para os elementos mais refinados da estrutura de requisitos; portanto, uma "história de usuário" é um tipo específico de história, ou seja, uma história centrada no usuário. Quando nos referirmos ao modelo apresentado na Fig. 5, usaremos o termo "modelo de história de usuário". Observe que o modelo de história de usuário pode ser usado para descrever não apenas histórias, mas também épicas e recursos, se for considerado apropriado.

Em geral, deve-se observar que as histórias de usuários não são requisitos completos em si; elas são mais uma promessa de comunicação. Para criar requisitos completos e os itens correspondentes do backlog, será necessário um detalhamento adicional (documentado e/ou em discussão).

Ao apresentar as histórias de usuários [Cohn2010], Bill Wake [Wake2003] definiu que as histórias de usuários devem estar em conformidade com os princípios INVEST (Independent, Negotiable, Valuable, Estimated, Small, Testable).

Os itens do backlog geralmente são escritos em cartões de índice ou notas adesivas, além de ferramentas de TI (p. ex., Jira, MS-Devops) e organizados em paredes ou mesas para facilitar o planejamento e a discussão. Usando o modelo 3C (cartão, conversa, confirmação), esses itens do backlog podem ser discutidos em mais detalhes e complementados com outros artefatos de requisitos (p. ex., modelos, documentação).

### 3.4 Técnicas de Divisão e Agrupamento

#### Objetivos Educacionais

- EO 3.4.1 Analisar e aplicar técnicas de divisão para requisitos funcionais de granulação maior (L4)
- EO 3.4.2 Analisar e aplicar o agrupamento e a abstração de requisitos funcionais detalhados em requisitos mais abrangentes (L4)

#### Conteúdo

Para gerar itens do backlog que sejam pequenos o suficiente para caber em uma única iteração, os itens maiores do backlog podem ser divididos em itens mais refinados (histórias). Vários autores sugeriram padrões que podem ser aplicados para este fim, desde a redução da lista de funcionalidades até a redução das variações de negócio ou canais de entrada [Leffl2010]. Observe que até mesmo as histórias mais refinadas devem ser definidas de forma que forneçam algum valor para pelo menos um stakeholder.

Itens menores do backlog podem ser agrupados em blocos ou processos maiores (p. ex., casos de uso) para maior clareza e exibidos graficamente por meio de um mapa de histórias. Isso ajuda a manter uma visão geral dos requisitos e, ao mesmo tempo, a apresentar a estratégia de médio prazo, atribuindo os requisitos a sprints e versões.

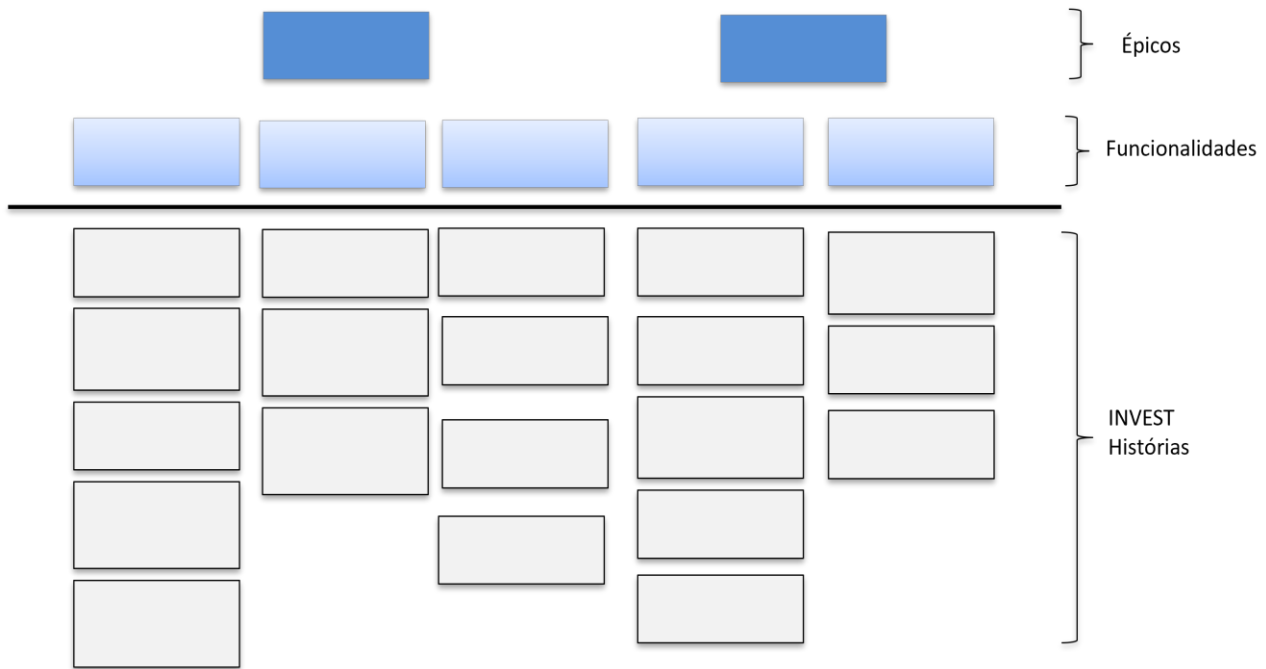


Figura 6: Divisão e agrupamento

### 3.5 Saber quando parar

#### Objetivos Educacionais

- EO 3.5.1 Aplicar o refinamento de requisitos (L4)
- EO 3.5.2 Aplicar a garantia de qualidade de histórias em um contexto ágil (L4)

#### Conteúdo

O product owner é responsável por continuar as discussões com os desenvolvedores até que ambas as partes tenham um entendimento comum dos requisitos [Meyer2014].

Para esse nível de entendimento comum, é definida a Definição de Pronto (DoR), que pode ser usada para a garantia de qualidade dos itens da lista de pendências, a fim de garantir que um entendimento comum e um nível suficiente de detalhes tenham sido alcançados.

### 3.6 Documentação do Projeto e do Produto de Requisitos

#### Objetivos Educacionais

- EO 3.6.1 Entenda como distinguir entre informação/documentação do projeto e do produto (L2)
- EO 3.6.2 Métodos e técnicas para preservar a informação para uso futuro (L1)

#### Conteúdo

O backlog do produto pode ser considerado como um substituto para o documento de requisitos de um projeto tradicional. No entanto, é importante lembrar que a parte escrita de um item do backlog (p. ex., usando o modelo de história de usuário: "Como usuário, eu quero...") está incompleto até que as discussões sobre esse item do backlog tenham ocorrido.

Geralmente, é melhor pensar na parte escrita como uma referência a uma representação mais precisa desse requisito. Os itens do backlog (Épicos, Recursos, Histórias, ...) podem apontar para um diagrama que represente um fluxo de trabalho, uma planilha que ilustre como realizar um cálculo ou qualquer outro artefato que o product owner ou a equipe deseje.

Além do backlog do produto, pode haver diferentes motivos para uma documentação detalhada dos requisitos, como para fins de comunicação, para fins de reflexão, para fins legais, para fins de arquivamento.

A definição de um nível adequado de documentação depende de vários fatores, como o escopo do projeto, o número de stakeholders envolvidos, as restrições legais e/ou os aspectos críticos de segurança do projeto. Com base nesses fatores, as equipes em um ambiente ágil tentam evitar um excesso de documentação e encontrar um denominador mínimo de documentação útil.

Trabalhar com um backlog de produto "vivo" é uma maneira eficiente de lidar com a documentação, mas nem sempre é suficiente. A documentação estruturada de todos os requisitos implementados em um produto que esteja atualizada não é apenas legalmente obrigatória em alguns projetos, mas também serve como ponto de partida ideal para identificar mais rapidamente as solicitações de alteração com base na documentação existente.

## 4 Lidando com Requisitos de Qualidade e Restrições (L3)

Duração: 90 minutos + 30 minutos de exercícios

Termos: Requisitos de qualidade, restrições, árvore de qualidade, Definição de Feito (DoD), critérios de aceite

### Conteúdo

Esta UE analisa os requisitos e restrições de qualidade em projetos ágeis. Embora o termo "requisitos não-funcionais" ainda seja usado com frequência na prática como um termo abrangente, usamos as categorias mais concretas e precisas "requisitos de qualidade" e "restrições" de acordo com [Glin2024].

### 4.1 Entendendo a importância dos requisitos e restrições de qualidade (L2)

#### Objetivos Educacionais

- EO 4.1.1 Compreender a importância dos requisitos de qualidade em um contexto ágil (L2)
- EO 4.1.2 Esquemas de categorização de acordo com os requisitos e restrições de qualidade (L2)

### Conteúdo

Muitos métodos ágeis concentram-se apenas nos requisitos funcionais e não dão ênfase suficiente às qualidades e restrições [Meyer2014].

As principais restrições e qualidades previstas para o sistema devem ser explicitadas no início do ciclo de vida de um produto, uma vez que determinam as escolhas arquitetônicas fundamentais (infraestrutura, arquitetura de software e projeto de software). Ignorá-los ou aprender muito tarde no projeto pode colocar em perigo todo o esforço de desenvolvimento. Outras qualidades podem ser capturadas iterativamente, sob demanda, como com os requisitos funcionais [Meyer2014].

Esquemas de categorização para requisitos e restrições de qualidade (p. ex., [RoRo2013], [ISO25010]) podem ser usados como listas de verificação para não esquecer categorias importantes.

### 4.2 Acrescentar precisão aos requisitos de qualidade (L2)

#### Objetivos Educacionais

- EO 4.2.1 Entenda os detalhes ou a decomposição dos requisitos e restrições de qualidade (L2)
- EO 4.2.2 Entenda a derivação dos requisitos funcionais a partir dos requisitos de qualidade (L2)
- EO 4.2.3 Entenda a especificação dos critérios de aceite dos requisitos de qualidade (L2)
- EO 4.2.4 Entenda o valor agregado de árvores de qualidade (L2)

## Conteúdo

Os requisitos iniciais de qualidade são muitas vezes deliberadamente vagos. Eles têm que ser capturados em seu formato vago como um ponto de partida. Requisitos de qualidade e restrições vagos podem ser refinadas em requisitos mais precisos. Algumas vezes, requisitos funcionais concretos serão derivados deles.

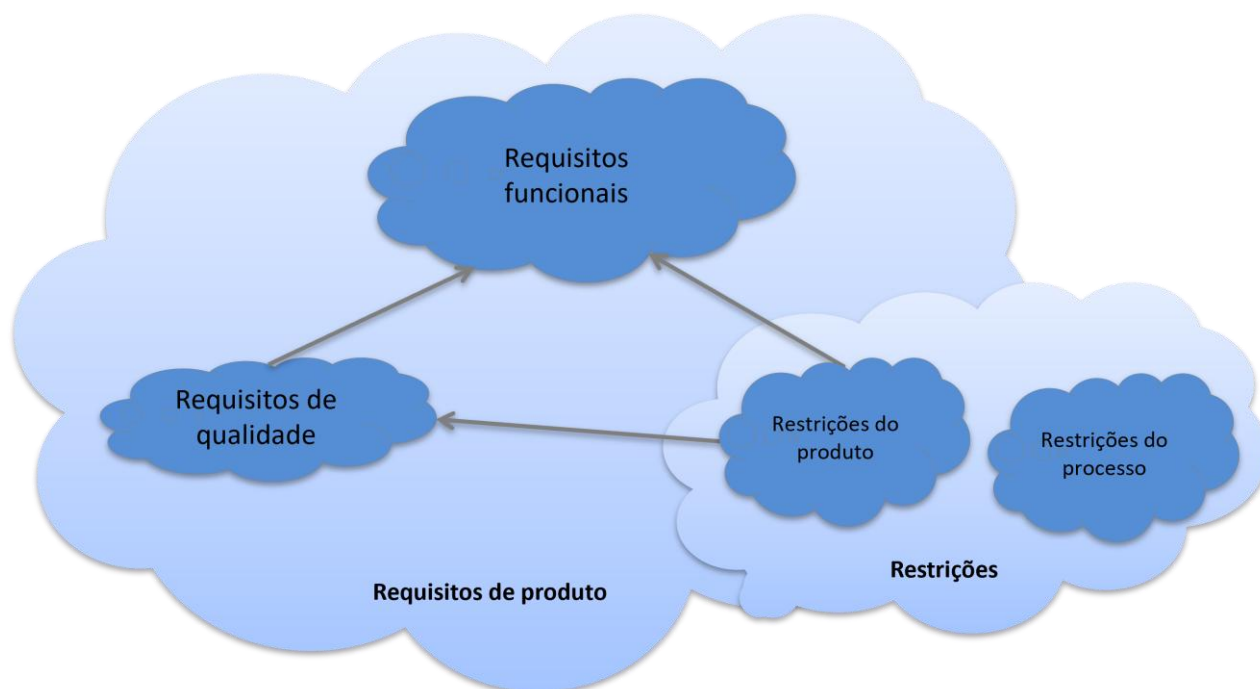


Figura 7: Decomposição de requisitos e restrições de qualidade

Decompor (ou detalhar) um requisito de qualidade significa especificar os requisitos de qualidade em um nível mais baixo de detalhe, por exemplo, usando as generalizações nos esquemas de categorização como "usabilidade" e tornando-as mais precisas ao encontrar requisitos para "facilidade de uso" e "facilidade de aprendizado".

Derivar significa que os requisitos de qualidade podem ser alcançados definindo requisitos funcionais, ou seja, sugerindo funções que atinjam a qualidade desejada ou restrições. Um exemplo para refinar um requisito de segurança é a introdução de um conceito de papel e senhas.

Árvores de qualidade ([BOSS2022], [CleA2001]) também são uma forma comprovada de estruturar os requisitos de qualidade.

Os critérios de aceite também devem ser definidos para os requisitos de qualidade a fim de torná-los testáveis em um estágio posterior, assim como outros tipos de requisitos [PoRu2021]. O tipo de critério de aceite utilizado dependerá da categoria da qualidade.

## 4.3 Requisitos de qualidade e backlog (L3)

### Objetivos Educacionais

EO 4.3.1 Aplicar a vinculação de requisitos de qualidade a requisitos funcionais (L3)

- EO 4.3.2 Aplicar criando itens de backlog separados para os requisitos de qualidade (L3)
- EO 4.3.3 Compreender os requisitos de qualidade como parte do DoD (L2)
- EO 4.3.4 Compreender a diferença entre requisitos de qualidade e critérios de aceite (K2)

### Conteúdo

Os requisitos de qualidade generalizados precisam estar vinculados a requisitos funcionais mais específicos [PoRu2021], por exemplo, alguma taxa de transferência quantificável associada a uma história ou recursos de segurança específicos associados a um épico.

Outras qualidades, por exemplo, escalabilidade, capacidade de manutenção ou robustez, devem ser informadas ao desenvolvimento e verificadas em cada iteração. Uma maneira comum de conseguir isso é incluí-los na Definição de Pronto. Isto é frequentemente apoiado por testes automatizados [Leffl2010].

Outra abordagem é ter um registro separado (fora do backlog do produto) de tais qualidades para mantê-las visíveis para as equipes, por exemplo, como uma lista comum ou na forma de listas de verificação. Esses requisitos são todos de igual importância (pois todos precisam ser atendidos) [Leffl2010].

Também é uma boa prática tornar visíveis os relacionamentos entre requisitos funcionais e de requisitos de qualidade afetados através da instalação de uma matriz em uma parede, indicando a relação "afetada por" com marcas nas respectivas células.

Ao estruturar um backlog, os product owners geralmente se deparam com a questão de saber se um requisito de qualidade reconhecido/levantado é realmente um requisito de qualidade, um critério de aceitação ou talvez também um critério de aceitação para um requisito de qualidade.

- Os requisitos de qualidade referem-se a questões de qualidade que não são cobertas pelos requisitos funcionais. Como performance, disponibilidade, capacidade de manutenção, segurança ou confiabilidade.
- Os critérios de aceite são critérios que um requisito (que pode ser um requisito funcional ou de qualidade) deve atender para ser aceito pelas stakeholders.

Podemos ver que tanto os requisitos funcionais quanto os requisitos de qualidade podem e devem ter critérios de aceite.

## 4.4 Tornando as Restrições Explícitas (L2)

### Objetivos Educacionais

- EO 4.4.1 Compreender os diferentes tipos de restrições em um contexto ágil (L2)
- EO 4.4.2 Saber como caracterizar as restrições (L1)

### Conteúdo

As restrições são um tipo importante de requisitos que limitam as escolhas de projeto dos desenvolvedores [Glin2024]. As restrições podem ser categorizadas como restrições de produto ou restrições de processo. As restrições do produto incluem o uso necessário de tecnologias, a reutilização de componentes existentes, decisões de compra ou venda ou recursos na forma de material, conhecimento e competências. As restrições do processo,

por outro lado, são definidas por processos organizacionais ou de desenvolvimento. Estes incluem políticas e regulamentos organizacionais, limites financeiros, normas e padrões, regulamentos e auditorias de conformidade, restrições legais e governamentais.

É importante explicitar tais restrições para que todos na equipe estejam cientes delas. As mais limitantes devem ser conhecidas no início do projeto. Outras devem ser capturadas assim que forem descobertas. Em geral, as restrições costumam afetar vários requisitos funcionais. Isso levanta a questão de como as restrições (bem como os requisitos de qualidade) devem ser documentadas. Já explicamos as possibilidades para isso no capítulo 4.3. O mesmo se aplica às restrições: no mínimo, uma verificação deve ser incluída na Definição de Feito para garantir que elas foram atendidas.

Observe que você não precisa necessariamente escrever todas as restrições como um item da lista de pendências. Pode ser suficiente informar à equipe que, por exemplo, C# e ORACLE são restrições inegociáveis.

Tais restrições são normalmente aplicáveis a uma gama mais ampla de projetos. Depois de registrados, eles podem ser facilmente reutilizados em outros projetos ou produtos.

# 5 Priorização e Estimativa dos Requisitos (L3)

Duração: 120 minutos + 90 minutos de exercícios

Termos: Valor do negócio, MVP, MMP, Pôquer de planejamento, Cone de incerteza, Velocidade, Dimensionamento, Histórias de referência, Tamanhos de camisetas, Sequência de Fibonacci

## Conteúdo

Mesmo em um mundo ágil perfeito, as previsões são necessárias para determinar quanto trabalho pode ser "feito" em uma iteração previamente especificada (timebox). Além disso, as organizações de desenvolvimento que excedem uma equipe geralmente precisam de previsões para priorizar e planejar adequadamente o trabalho.

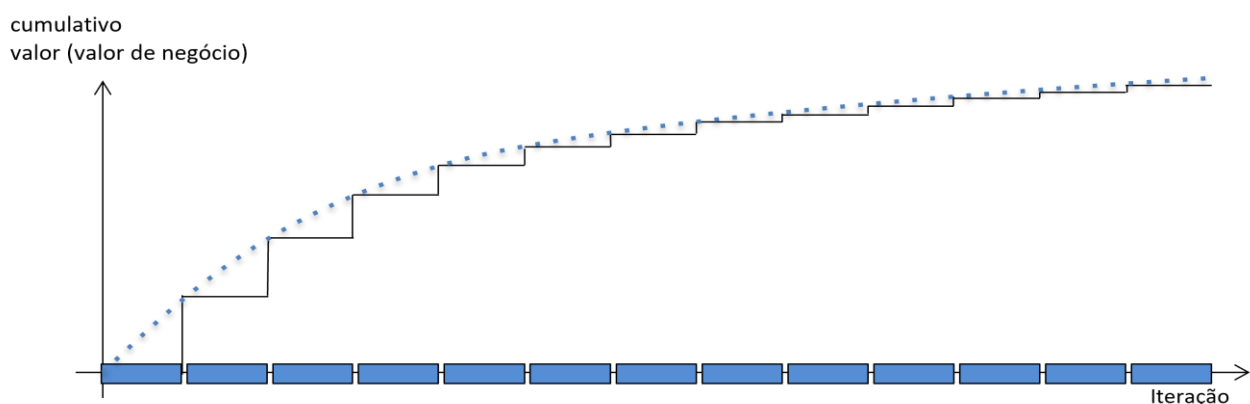


Figura 8: Aumento do valor comercial

## 5.1 Determinação do Valor de Negócio

### Objetivos Educacionais

- OE 5.1.1 Entender a determinação do valor de negócio (L2)
- OE 5.1.2 Entender como usar o valor de negócio para priorizar itens de backlog (L2)
- EO 5.1.3 Aplicar métodos alternativos de cálculo do valor comercial (L3)
- OE 5.1.4 Entender como alinhar a medição do valor de negócio às metas estratégicas da organização (L2)

### Conteúdo

As abordagens ágeis visam maximizar o valor geral do negócio e otimizar permanentemente o processo geral de criação de valor comercial [Leffl2010]. Em um backlog do produto, todos os requisitos (grosseiros ou finos) devem ser ordenados principalmente pelo valor que oferecem à organização. Um pré-requisito para fazer isso é uma definição acordada do valor comercial para este produto/empresa.

O valor comercial não é definido apenas pelo lucro: os cálculos alternativos incluem Retorno do Investimento, Período de Retorno do Investimento, Valor Presente Líquido, Emprego Mais

Curto Ponderado Primeiro (WSJF), Custo do Atraso e Balanced Scorecard. Valor de mercado, tempo para comercializar e redução de riscos potenciais, todos representam potencialmente tipos de valor comercial, assim como a excelência operacional e organizacional [Rein2009].

De fato, a definição de valor comercial pode ser diferente em cada organização, cada projeto, e da perspectiva de diferentes stakeholders. Os profissionais devem entender como alinhar a medição do valor comercial aos objetivos estratégicos da organização, e ser capazes de adaptar este alinhamento à medida que estes objetivos mudam.

## 5.2 Valor de Negócio, Riscos e Dependências (L3)

### Objetivos Educacionais

EO 5.2.1 Compreender as dependências entre o valor comercial potencial e os riscos relacionados (L2)

### Conteúdo

Muitas vezes o valor comercial potencial e os riscos são interdependentes. O foco em um valor comercial específico pode aumentar riscos específicos, mudando o foco do valor comercial pode mudar os riscos também [Rein2009].

Em cada caso, a priorização dos requisitos deve ser ajustada de acordo com a estratégia selecionada, levando em conta as dependências entre os requisitos.

## 5.3 Expressando Prioridades e Organizando o Backlog

### Objetivos Educacionais

EO 5.3.1 Aplicar a priorização de pedidos em atraso (L3)

EO 5.3.2 Aplicar diferentes estratégias básicas de priorização (L3)

EO 5.3.3 Aplicar a determinação de dependências entre requisitos (L3)

EO 5.3.4 Compreender a sequência de itens do backlog com base em suas dependências (L2)

### Conteúdo

Uma vez determinado o valor que significa para você, você tem que expressar estas prioridades e ordenar o backlog de acordo com as prioridades dadas aos itens de backlog. Há muitos métodos diferentes para atribuir valor aos itens do backlog. Alguns deles são muito simples, outros são altamente complexos. Exemplos de estratégias de priorização incluem:

- MoSCoW.
- Alto/Médio/Baixo
- Com base em uma série de números que expressam o valor comercial
- Com base na avaliação do negócio usando vários critérios

Itens de Backlog	Volume de negócios no segundo trimestre		Minimização do risco técnico		Aumentar a usabilidade		Resultado geral
	Peso = 5	Valor	Peso=4	Valor do Risco	Peso=2	Valor da Usabilidade	
Requisito 1	3	15	0	0	0	0	<b>15</b>
Requisito 2	0	0	3	12	1	2	<b>14</b>
Requisito 3	4	20	2	8	2	4	<b>32</b>
Requisito 4	2	10	2	8	3	6	<b>24</b>
Requisito 5	0	0	2	8	5	10	<b>18</b>
...							
...							

Figura 9: Prioridades e ordenação do backlog

Ao priorizar os itens da lista de pendências, geralmente deve-se considerar que quanto mais próximo estiver o momento da implementação planejada de um item da lista de pendências, mais claras deverão ser as prioridades dos itens selecionados. Uma priorização com os critérios "alto", "médio", "baixo" geralmente resulta em um número muito grande de itens do backlog que recebem o valor "alto". Por exemplo, se 30% de todos os itens do backlog forem atribuídos a uma alta prioridade, o resultado é que os desenvolvedores não sabem o que é mais importante para o product owner. Isso também indica que o product owner não tem uma estratégia clara para a implementação de curto e médio prazo. O objetivo da priorização deve ser sempre fazer uma declaração clara sobre o que os stakeholders podem esperar como valor do produto em um futuro próximo.

## 5.4 Estimativa de itens do backlog (L3)

### Objetivos Educacionais

- EO 5.4.1 Aplicar previsões e estimativas (L3)
- EO 5.4.2 Compreender como obter uma previsão de médio prazo (L2)
- EO 5.4.3 Compreender a vantagem das estimativas relativas, de categorização e de grupo (L2)
- EO 5.4.4 Compreender as técnicas de estimativa (L2)

### Conteúdo

As estimativas iniciais do projeto geralmente são imprecisas. Eles se tornam cada vez mais precisos à medida que a atividade é repetida (um princípio conhecido como Cone da Incerteza). Ao analisar o que foi entregue em iterações anteriores, a velocidade da equipe pode ser calculada. Isso permite que a capacidade de iterações futuras seja mais bem estimada.

Para obter melhores estimativas de médio prazo, os requisitos grandes, como os épicos, são divididos em recursos. Isso permite que as estimativas sejam feitas em nível de recurso e depois adicionadas ao épico. Embora essas estimativas ainda não sejam muito precisas, elas

são úteis para a estimativa épica. Além disso, eles servem como uma espécie de teste de conhecimento sobre a Epic (suposições etc.).

Os métodos ágeis definem regras que ajudam a fazer estimativas melhores e mais precisas:

- Todos os envolvidos na estimativa devem ter a mesma compreensão do trabalho que precisa ser "feito".
- As estimativas devem ser feitas por aqueles que fazem o trabalho, a equipe multifuncional (Desenvolvedores no Scrum). Isto ajuda a colocar todas as pessoas envolvidas no mesmo nível de conhecimento, trocando conhecimentos e suposições sobre o trabalho a ser feito.
- As estimativas devem ser feitas relativamente / relativamente ao trabalho já realizado (Estimativa por analogia), uma vez que é mais provável que essas estimativas sejam precisas do que estimativas absolutas.
- As estimativas devem ser feitas em uma unidade artificial que represente esforço, complexidade e risco em uma só.

Várias técnicas suportam a estimativa relativa. Os mais conhecidos são o pôquer de planejamento [Cohn2006] e o Magic Stimulation.

## 5.5 Escolha de uma estratégia de desenvolvimento (L2)

### Objetivos Educacionais

- EO 5.5.1 Compreender o conceito de MVP (produto mínimo viável) (L2)
- EO 5.5.2 Compreender o conceito de MMP (produto mínimo comercializável) (L2)

### Conteúdo

Diferentes estratégias podem ser aplicadas ao selecionar o que deve ser escolhido para lançamentos antecipados, com base no valor conhecido, risco e esforço necessários para desenvolver um item em atraso. Dois conceitos são típicos para um desenvolvimento Ágil: desenvolver um produto mínimo viável (MVP) e desenvolver um produto mínimo comercializável (MMP).

Um produto mínimo viável é a versão de um novo produto que permite a uma equipe coletar a quantidade máxima de aprendizagem validada sobre os clientes com o menor esforço. O MVP é a ideia central da metodologia Lean Startup desenvolvida por Eric Ries, que se baseia no ciclo Construir-Medir-Aprender. O MVP não é necessariamente um produto de software implementável.

O MMP descreve o produto com o menor conjunto de recursos possível que atende às necessidades dos primeiros usuários (inovadores e adotantes iniciais) e, portanto, pode ser comercializado.

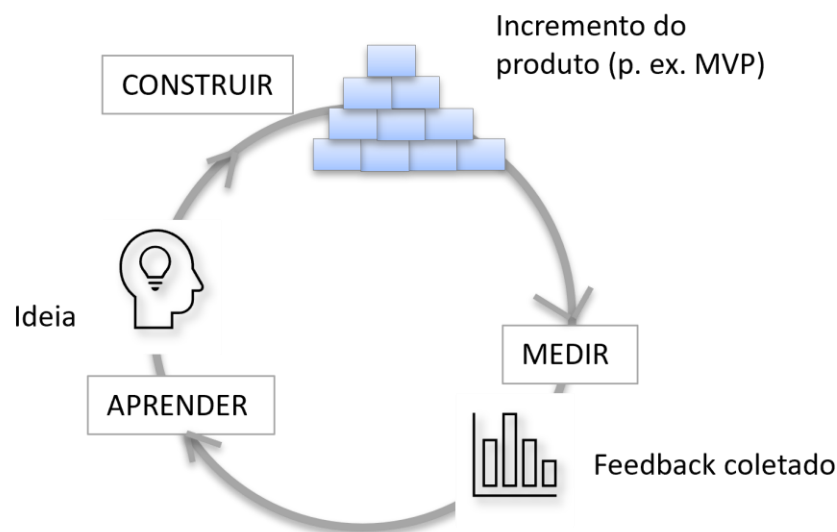


Figura 10: O ciclo "Build-Measure-Learn" de desenvolvimento enxuto

## 6 Escalando RE@Agile (L2)

Duração: 105 minutos

Termos: Estruturas de dimensionamento, dimensionamento de requisitos e equipes, estruturação de requisitos e equipes em grande escala, roteiros e planejamento em grande escala, validação de produtos

### Conteúdo

A ER é mais fácil para produtos que são pequenos o suficiente para serem manuseados por uma única equipe em um único local.

Nesta unidade educacional, discutimos porque o desenvolvimento de produtos às vezes precisa ser dimensionado e por que os produtos precisam ser desenvolvidos por mais de uma equipe, seja no mesmo local ou distribuída geograficamente. Ao fazer o dimensionamento, o product owner geral (como a função responsável pelo gerenciamento de requisitos) provavelmente terá mais dificuldades com os aspectos de gerenciamento do que com os aspectos de requisitos. Discutiremos como os dois fatores, time-to-market e complexidade (seja complexidade funcional ou requisitos de qualidade desafiadores), justificam e impulsionam o processo de escalonamento. Mas as restrições organizacionais e técnicas também influenciam a forma como o dimensionamento é realizado.

As perguntas a seguir são importantes:

- O que significa dimensionamento e como ele afeta os requisitos e as equipes?
- Como (re)organizamos os requisitos e as equipes em grande escala?
- Como as versões e os roteiros são definidos e usados no planejamento de longo prazo?
- Como os requisitos são validados em ambientes de escala?

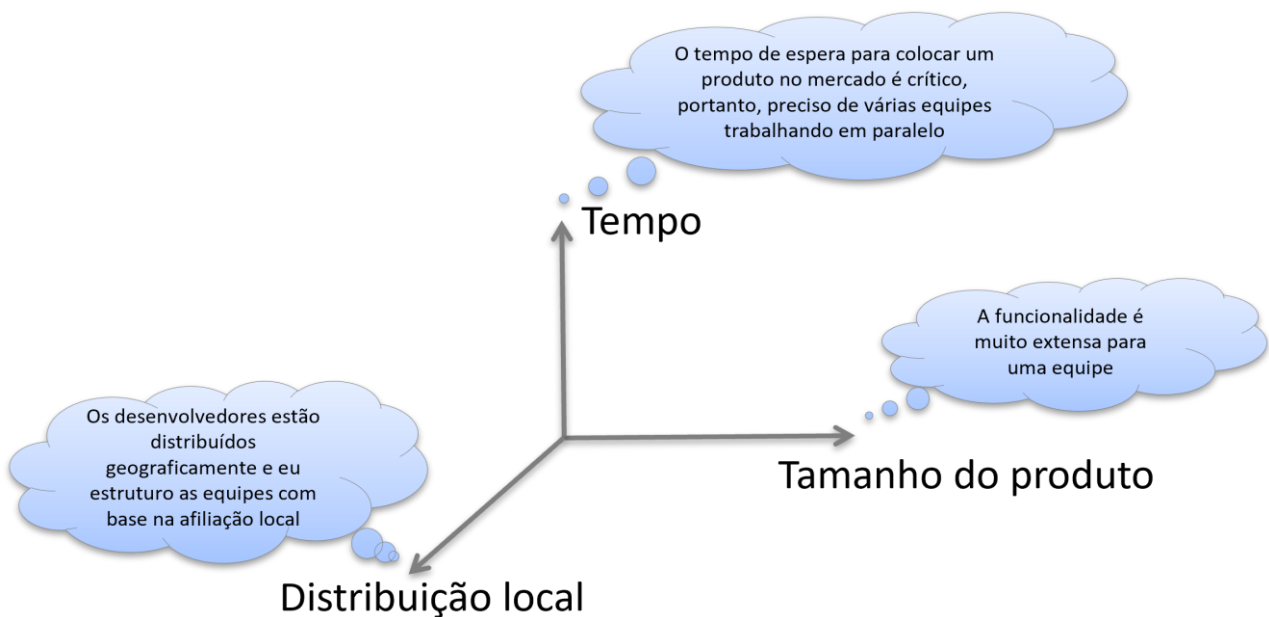


Figura 11: Motivos para escalar

## 6.1 Escalando Requisitos e Equipes (L2)

### Objetivos Educacionais

- EO 6.1.1 Conhecer exemplos comuns de estruturas de dimensionamento (L1)
- EO 6.1.2 Compreender os desafios e o mecanismo para dimensionar os requisitos no contexto ágil (L2)

### Conteúdo

Usamos o termo "escalonamento" para descrever uma alteração no tamanho, seja do sistema ou do produto, seja do número de pessoas envolvidas.

Desde 2010, várias estruturas de escalonamento Ágil foram desenvolvidas para resolver esses problemas. Entre eles estão Nexus [Nexu2021], SAFe [SAFe2021a] [SAFe2021b], LeSS [LeSS], Scrum@Scale [S@SG2021], BOSSA Nova [BOSS2022], Scrum of Scrums [KnlvS2012], e Spotify [Spot2012], embora existam mais. As estruturas de escalonamento variam em seu nível de maturidade, o número de boas práticas, diretrizes e regras, e o grau de adaptabilidade às necessidades específicas de uma organização.

Quando você faz escalonamento, duas coisas sempre serão verdadeiras: você será forçado a adicionar uma hierarquia aos requisitos e uma hierarquia à organização. São necessários requisitos de granularidade grosseira ao discutir o produto como um todo; serão necessários requisitos de granularidade refinada nas equipes que implementam algum aspecto do produto. E as próprias equipes precisarão organizar sua cooperação para funcionar com sucesso dentro de uma equipe maior.

## 6.2 Critérios para estruturar Requisitos e Equipes em Grande Escala (L2)

### Objetivos Educacionais

- EO 6.2.1 Conhecer os desafios de organizar uma lista de pendências e de se comunicar sobre os requisitos em uma rede de equipes (L1)
- EO 6.2.2 Compreender a divisão de requisitos em diferentes configurações (de projeto) do contexto ágil (L2)

### Conteúdo

No desenvolvimento de produtos em larga escala, a maioria das equipes tem que trabalhar juntas no mesmo produto. Na prática, cada equipe desenvolve uma parte específica do produto que deve ser integrada a outras partes do produto geral para criar uma solução funcional. Somente o produto integrado tem valor para os stakeholders.

Ao dimensionar o desenvolvimento de produtos para várias equipes, não é suficiente que todos os product owners simplesmente se reúnam e, de alguma forma, discutam quais equipes devem desenvolver cada parte do produto, e depois esperem pelo melhor! São necessárias estruturas e práticas sofisticadas para apoiar a colaboração da equipe, gerenciar as mudanças de requisitos e permitir a entrega rápida do produto. Caso contrário, os desenvolvedores podem desperdiçar esforços coordenando com equipes que não são relevantes para seu trabalho.



Do ponto de vista dos requisitos, temos que fechar o ciclo: desde o requisito inicial (negócio) exigido pelos stakeholders, passando pela divisão de requisitos complexos em partes menores gerenciáveis pelos desenvolvedores e, em seguida, garantindo que os resultados reunidos se combinem para formar uma solução que possa ser liberada para a empresa.

Para trabalhar nos requisitos de forma colaborativa e tomar decisões razoáveis de forma autônoma, as equipes precisam de um panorama geral dos requisitos das outras equipes com as quais precisam colaborar, sem, no entanto, ficarem sobrecarregadas com todos os detalhes. Os product owners devem, portanto, encontrar um nível adequado de detalhes, suficiente para que as equipes entendam o impacto de suas decisões em outras equipes.

Para fornecer incrementos de produtos que possam ser enviados com o mínimo de dependência de outras equipes, as equipes em um ambiente Ágil devem trabalhar em recursos de ponta a ponta com fraco acoplamento. Em nosso contexto, o termo "funcionalidades de ponta a ponta" refere-se a um conjunto de funções coerentes que executam uma tarefa específica que fornece valor de negócio aos stakeholders. Os casos de uso são uma abordagem para estruturar os requisitos, nem sempre associada ao Agile, mas recomendada por vários autores (por exemplo [Jaco2011], Cockburn, [Leffl2010]).

Infelizmente, em muitos casos, não é tão fácil decompor os requisitos com base em unidades fracamente acopladas de funcionalidade ponta a ponta. Devido ao projeto arquitetônico (p. ex., tecnologia, infraestrutura, componentes do sistema, plataforma comum, camadas arquitetônicas como front e backend), bem como considerações organizacionais (habilidades especializadas, localização da equipe, subcontratados), as unidades de funcionalidade podem se sobrepor. Isso significa que diferentes equipes em um ambiente ágil devem trabalhar juntas para implementar recursos específicos e seus respectivos product owners precisam colaborar mais estreitamente com os requisitos. Alternativamente, uma equipe dedicada pode ser estabelecida para trabalhar especificamente na sobreposição, e colaborar com cada uma das equipes originais focadas em uma unidade de funcionalidade.

## 6.3 Roadmaps e Planejamento em Grande Escala (L2)

### Objetivos Educacionais

- EO 6.3.1 Saber a diferença entre um roteiro e uma lista de pendências em um contexto ágil (L1)
- EO 6.3.2 Compreender a criação e o gerenciamento de um roteiro no contexto ágil (L2)

### Conteúdo

No desenvolvimento de produtos em larga escala, os donos de produtos gerenciam os requisitos no backlog focado no produto. Em contraste com o backlog, um roadmap é usado para planejar o desenvolvimento do produto incrementalmente. Um roteiro é uma previsão de como o produto crescerá [Pich2016]. Os roteiros não alteram o conteúdo dos itens do backlog, mas os organizam em uma linha do tempo. Ele responde à pergunta sobre quando podemos esperar, grosso modo, quais funcionalidades.

Um roteiro é um meio útil para comunicar metas e decisões (estratégicas) aos desenvolvedores e outros stakeholders. Ele divide uma meta de longo prazo em iterações gerenciáveis, representa as dependências entre as equipes e fornece orientação e transparência aos stakeholders.

Um roadmap mostra objetivos estratégicos, marcos e requisitos de granularidade grosseira. Marcos importantes podem ser internos ou determinados por eventos externos, como uma feira ou a introdução de nova regulamentação no mercado. A representação de um roteiro depende de seu objetivo, grupo alvo e horizonte de planejamento.

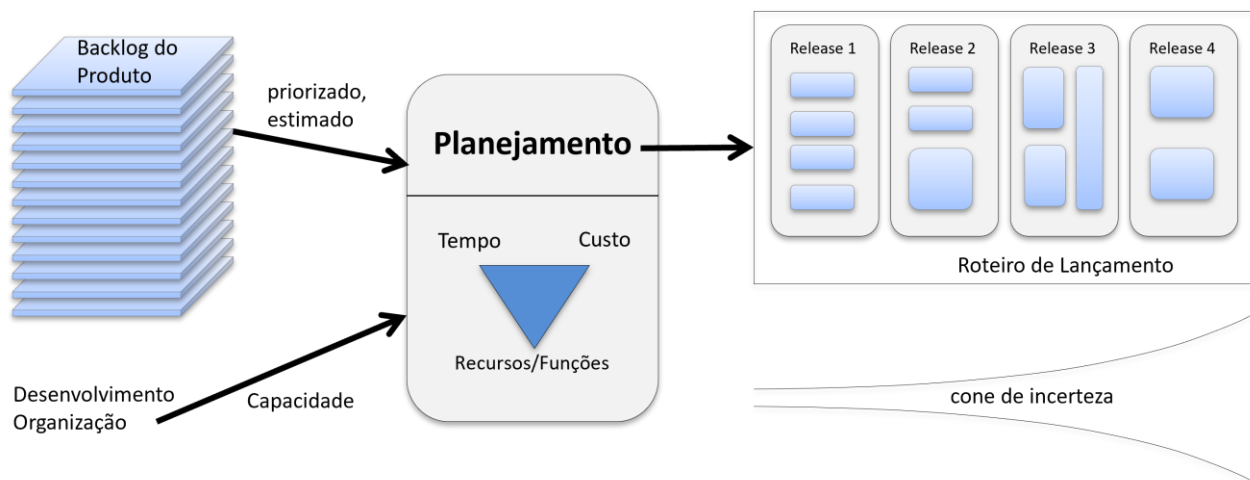


Figura 12: Roadmaps e planejamento em larga escala

Para desenvolver um roadmap de produtos a longo prazo, um dono de produto deve primeiro definir uma visão e estratégia do produto. Posteriormente, os donos de produtos devem então elucidar os requisitos de granularidade grosseira, engajando-se com os stakeholders necessários. Neste momento, não há necessidade de investir tempo em requisitos detalhados. Embora os requisitos estejam sujeitos a um alto nível de incerteza nesta fase, o roadmap do produto como um plano de iteração rudimentar e inicial é suficientemente bom para apoiar o planejamento e a sincronização.

## 6.4 Validação do produto (L2)

### Objetivos Educacionais

EO 6.4.1 Compreender métodos concretos para validar os requisitos do produto no contexto ágil (L2)

### Conteúdo

Uma das principais ideias do desenvolvimento Ágil é desenvolver uma pequena parte do produto, gerar feedback envolvendo os stakeholders e adaptar o desenvolvimento do produto de acordo com as descobertas e os insights obtidos. Assim, seguindo o princípio do ciclo Build-Measure-Learn [Ries2011], a validação do produto torna-se um passo importante para obter feedback rápido. Sempre que um novo incremento de produto é lançado, os product owners usam esse incremento para verificar seu valor comercial e examinar se os requisitos do produto foram compreendidos corretamente.

A validação no nível do produto (p. ex., revisão de sprint no Scrum) é um método importante no desenvolvimento de produtos em larga escala, pois garante que os product owners compartilhem a responsabilidade total, desde os requisitos de negócio até a integração do

produto. É o produto inteiro que tem valor para os stakeholders, não apenas pequenas partes de produto.

Outra abordagem para a validação de produtos no desenvolvimento em larga escala é aquela baseada na análise de dados [MaeA2016]. O incremento integrado do produto é entregue aos usuários e, com base em seu comportamento, são feitas medições se as funcionalidades do produto têm um impacto positivo, neutro ou negativo.

## 7 Definições de termos, glossário

Para as definições dos termos, recomendamos que o leitor consulte o glossário de engenharia de requisitos do IREB CPRE [Glin2024], que não é apenas um glossário abrangente da terminologia de engenharia de requisitos, mas também define muitos termos do campo da agilidade. Para termos específicos de agilidade, o leitor pode consultar o Guia do Scrum atual [ScSu2020].

## 8 Referências

- [Alex2005] Alexander, I. F.: A Taxonomy of Stakeholders – Human Roles in System Development. International Journal of Technology and Human Interaction, Vol 1, 1, 2005, páginas 23–59.
- [BOSS2022] <https://www.agilebossanova.com/#bossanova>. Última revisão em março de 2025.
- [CleA2001] P. Clements et al.: Evaluating Software Architectures, SEI Series in Software Engineering, 2001
- [Cohn2010] Cohn, M.: User Storys für die agile Software-Entwicklung mit Scrum, XP u.a., mitp, 2010
- [Cohn2006] Cohn, M.: Agile Estimation and Planning, Addison Wesley, 2006
- [Dora1981] Doran, G. T: There’s a S.M.A.R.T. way to write management’s goals and objectives, Management Review. AMA FORUM. AMA FORUM. 70 (11): 35–36 1981.
- [Glin2024] Glinz, M.: A Glossary of Requirements Engineering Terminology. Standard Glossary for the Certified Professional for Requirements Engineering (CPRE) Studies and Exam, Version 2.1.1, 2024. <https://cpre.ireb.org/en/knowledge-and-resources/downloads#cpre-glossary>. Última revisão em março de 2025.
- [HeHe2011] Heath, C., Heath, D.: Switch: Veränderungen wagen und dadurch gewinnen. Crown Business, 2010
- [High2001] Highsmith, J.: Design the Box. Agile Project Management E-Mail Advisor 2001, <http://www.joelonsoftware.com/articles/JimHighsmithonProductVisi.html>. Última revisão em março de 2025.
- [Hrus2017] [https://b-agile.de/downloads/articles/story\\_splitting.pdf](https://b-agile.de/downloads/articles/story_splitting.pdf). Última revisão em março de 2025.
- [ISO25010] ISO/IEC 25010:2023: Systems and software engineering – Systems and Software Quality Requirements and Evaluation (SQuaRE) – System and software quality models: <https://www.iso.org/standard/78176.html>. Última revisão em março de 2025.
- [Jaco2011] <https://www.ivarjacobson.com/publications/white-papers/use-case-ebook>. Última revisão em março de 2025.
- [Leffl2010] Leffingwell, D.: Agile Software Requirements – Lean Requirements Practices for Teams, Programs, and the Enterprise. Addison Wesley, 2010.
- [LeSS] Large-Scale Scrum: <https://less.works>. Última revisão em março de 2025.
- [MaeA2016] Maalej, W., Nayebi, M., Johann T., Ruhe, G.: Toward Data-Driven Requirements Engineering. IEEE Software (Volume 33, Issue 1), 2016.
- Meye2014] Meyer, B.: Agile! The Good, the Hype and the Ugly, Springer, 2014.

- [Nexu2021] <https://www.Scrum.org/resources/nexus-guide>. Última revisão em março de 2025.
- [Pich2016] Pichler, R.: Strategize – Product Strategy and Product Roadmap Practices for the Digital Age, Pichler Consulting 2016.
- [PoRu2021] Pohl, K., Rupp, C.: Basiswissen Requirements Engineering: Aus- und Weiterbildung nach IREB-Standard zum Certified Professional for Requirements Engineering Foundation Level, dpunkt.verlag, 5. Auflage, 2021.
- [Prim2017] CPRE RE@Agile Primer <https://cpre.ireb.org/en/downloads-and-resources/downloads#cpre-agile-primer-syllabus-and-study-guide>. Última revisão em março de 2025.
- [Rein2009] Reinertsen, D.G.: The Principles of Product Development Flow – Second Generation Lean Product Development. Celeritas Publishing, 2009.
- [Ries2011] Ries, E.: The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses, Crown Business, New York, NY 2011.
- [RoRo2013] Robertson S. Robertson J.: Mastering the Requirements Process – Getting Requirements Right, 3rd edition, Addison Wesley, 2013.
- [Robe2003] Robertson, S.: Stakeholders, Goals, Scope: The Foundation for Requirements and Business Models, 2003, <https://www.volere.org/wp-content/uploads/2018/12/StkGoalsScope.pdf>. Última revisão em março de 2025.
- [SAFe2021a] <https://www.scaledagileframework.com/roadmap/>. Última revisão em março de 2025.
- [SAFe2021b] <https://www.scaledagileframework.com/pi-planning/>. Última revisão em março de 2025.
- [SAFe2021] <https://www.scaledagileframework.com/safe-requirements-model/>. Última revisão em março de 2025.
- [S@SG2021] Sutherland, J. and Scrum, Inc: Scrum@Scale Guide: <https://www.Scrumatscale.com/Scrum-at-scale-guide/>. Última revisão em março de 2025.
- [KnIvS2012] Kniberg, H.; Ivarsson, A.: Scaling Agile @ Spotify with Tribes, Squads, Chapters & Guilds. <https://blog.crisp.se/wp-content/uploads/2012/11/SpotifyScaling.pdf>. Última revisão em março de 2025.
- [Spot2012] <https://blog.crisp.se/wp-content/uploads/2012/11/SpotifyScaling.pdf>. Última revisão em março de 2025.
- [ScSu2020] Schwaber, K., Sutherland, J.: The Scrum Guide. 2020 <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>. Última revisão em março de 2025.

[Wake2003] Wake, B: INVEST in Good Stories, and SMART Tasks, 2003,  
<https://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>. Última  
revisão em março de 2025.