



FIAP

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

FULL STACK, APPS & ARTIFICIAL INTELLIGENCE

SE_&_BM: SOFTWARE ENGINEERING AND BUSINESS MODEL
Prof. Freitas – 2025

10 - Requisitos e Backlog do Produto

Requisitos do Produto

Chamamos de requisito de software a formalização do entendimento do problema e das necessidades do usuário traduzidos sob a forma de referências e especificações para a construção do software.

Quais as características funcionais que o software deve ter? E que funcionalidades (funções)?

Responder essas perguntas caracteriza os REQUISITOS FUNCIONAIS

Requisitos do Produto

Os REQUISITOS FUNCIONAIS delimita o escopo do software, ou seja, o que ele faz e o que ele não faz!

Essas demandas darão origem aos REQUISITOS NÃO FUNCIONAIS, que são características gerais que afetarão a qualidade do sistema, abrangendo aspectos como desempenho, usabilidade, robustez, confiabilidade, entre outros.

Requisitos do Produto

No projeto, é solicitado o REQUISITO DO NEGÓCIO, que são regras que devem ser obedecidas para atender as condições estabelecidas pelo negócio.

Por exemplo, podemos ter regras como: política de preços da empresa, métodos de pagamento, regras de parcelamento, prazos de entrega ou limites de crédito.

Requisitos do Produto

REQUISITOS FUNCIONAIS

Funções que o sistema deve prever para atender às necessidades operacionais dos processos que o sistema irá suportar

REQUISITOS NÃO FUNCIONAIS

Características gerais que afetarão a qualidade do sistema abrangendo aspectos como desempenho, usabilidade, robustez, confiabilidade, entre outros

REQUISITOS DE NEGÓCIO

Regras que devem ser obedecidas pelas funcionalidades para atender as condições estabelecidas pelo negócio

Teremos uma aula para tratar desse tema

Requisitos de Software

Requisitos Funcionais

São **declarações de serviços que o sistema deve fornecer**, de como o sistema deve reagir a entradas específicas e de como o sistema deve se comportar em determinadas situações. Em alguns casos, os requisitos funcionais também podem explicitar o que o sistema não deve fazer.

definem o que o sistema deve fazer.

Requisitos Não Funcionais

São **qualidades ou restrições aos serviços ou funções oferecidos pelo sistema**. Incluem restrições de timing, restrições no processo de desenvolvimento e restrições impostas pelas normas. Ao contrário das características individuais ou serviços do sistema, os requisitos não funcionais, muitas vezes, aplicam-se ao sistema como um todo.

especificam as características que o sistema deve possuir

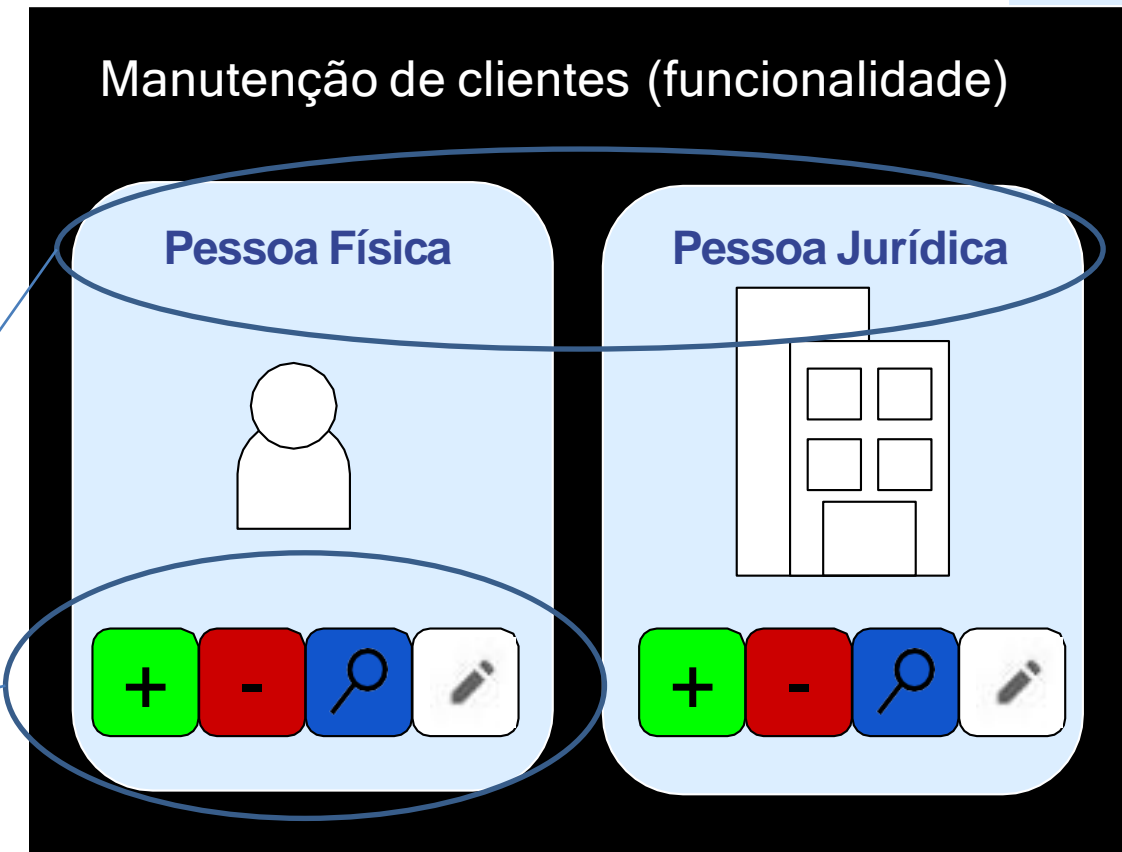
Requisitos de Software: Requisitos Funcionais

Não confundir **Requisito Funcional** com **Funcionalidade**.

Funcionalidade pode ter um ou mais Requisitos Funcionais.

Exemplo: a empresa tem um sistema com uma tela de “Manutenção de Clientes”. Nesta tela ela pode incluir/excluir/consultar/alterar clientes dos tipos pessoa física e pessoa jurídica.

Requisitos funcionais:
definem o que o sistema deve fazer.



Requisitos de Software: Requisitos Funcionais

Mas quantos requisitos são realizados (atendidos) por esta funcionalidade?

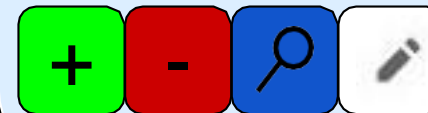
Oito requisitos!

Requisitos Funcionais (Identificador e Nome)
RF001 – Incluir cliente pessoa física
RF002 – Alterar cliente pessoa física
RF003 – Consultar cliente pessoa física
RF004 – Excluir cliente pessoa física
RF005 – Incluir cliente pessoa jurídica
RF006 – Alterar cliente pessoa jurídica
RF007 – Consultar cliente pessoa jurídica
RF008 – Excluir cliente pessoa jurídica

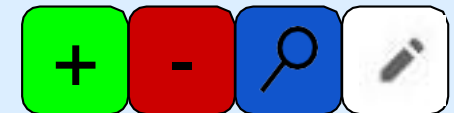
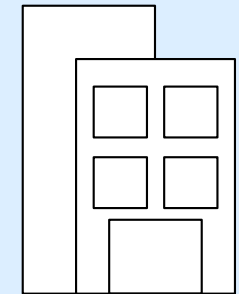
Funcionalidades somente existem para realizar e organizar requisitos funcionais.

Manutenção de clientes

Pessoa Física



Pessoa Jurídica



Requisitos de Software: Requisitos Funcionais

Um requisito funcional descreve o que o sistema deve fazer, enquanto a funcionalidade se refere às capacidades reais ou operações que o sistema possui para atender a esses requisitos.

Exemplos de requisitos funcionais incluem "o sistema deve permitir que os usuários façam login", "o sistema deve calcular a média de uma lista de números" e "o sistema deve permitir que os usuários enviem mensagens entre si".

Exemplos de funcionalidades podem incluir "sistema de autenticação de login", "módulo de cálculo de média" e "sistema de mensagens instantâneas".

Requisitos de Software: Requisitos Funcionais

Um Requisito Funcional de qualidade precisa atender alguns atributos específicos:

1. Unidade
2. Completude
3. Consistência
4. Atomicidade
5. Não-ambiguidade
6. Verificável
7. Rastreável
8. Prioridade

Requisitos de Software: Requisitos Funcionais

Um Requisito Funcional de qualidade precisa atender alguns atributos específicos:

1. Unidade

2. Completude
3. Consistência
4. Atomicidade
5. Não-ambiguidade
6. Verificável
7. Rastreável
8. Prioridade

O Requisito Funcional deve propor uma única coisa apenas. Não deve atender a mais de uma exigência. O Requisito Funcional **“Incluir cliente” não é unitário**, pois se refere a incluir clientes de tipos diferentes (pessoa física e jurídica), assumindo assim várias responsabilidades, quando deveria assumir apenas uma.

Requisitos de Software: Requisitos Funcionais

Um Requisito Funcional de qualidade precisa atender alguns atributos específicos:

1. Unidade
2. Completude
3. Consistência
4. Atomicidade
5. Não-ambiguidade
6. Verificável
7. Rastreável
8. Prioridade

O Requisito Funcional deve ser autocontido, deve ter “início/meio/fim”, ser completo. Por exemplo: o Requisito Funcional **“Pagar fatura” não é completo**, só conta “parte da estória”. Para ser completo deveria ser algo como “Pagar fatura com cartão de crédito para cliente pessoa física”.

Requisitos de Software: Requisitos Funcionais

Um Requisito Funcional de qualidade precisa atender alguns atributos específicos:

1. Unidade
2. Completude
3. **Consistência**
4. Atomicidade
5. Não-ambiguidade
6. Verificável
7. Rastreável
8. Prioridade

O Requisito Funcional **não deve contradizer outro** Requisito Funcional do mesmo escopo do projeto. É como termos dois Requisitos Funcionais se propondo a fazer uma mesma coisa, mas cada Requisito Funcional se propondo a fazer esta coisa de uma forma diferente.

Requisitos de Software: Requisitos Funcionais

Um Requisito Funcional de qualidade precisa atender alguns atributos específicos:

1. Unidade
2. Completude
3. Consistência
4. Atomicidade
5. Não-ambiguidade
6. Verificável
7. Rastreável
8. Prioridade

Um Requisito Funcional para ser atômico precisa também ter unidade, pois atomicidade remete a assumir apenas uma responsabilidade. Mas também **deve ser algo indivisível**, não podendo ser decomposto. Muitos Requisitos Funcionais possuem conjunção, dependem de outros para se realizarem. Onde temos dois Requisitos Funcionais **“Realizar compra de produto”** e **“Realizar pagamento com cartão de crédito”** na realidade, se pensarmos em atomicidade, temos um único RF que é **“Realizar compra de produto com pagamento em cartão de crédito”**.

Requisitos de Software: Requisitos Funcionais

Um Requisito Funcional de qualidade precisa atender alguns atributos específicos:

1. Unidade
2. Completude
3. Consistência
4. Atomicidade
5. Não-ambiguidade
6. Verificável
7. Rastreável
8. Prioridade

Um Requisito Funcional não pode ser ambíguo, não pode propor algo que não fica claro o que é. O Requisito Funcional “**Emitir relatório**” não quer dizer nada. Relatório de que, para que? “Emitir relatório de saldo” já é melhor, mas ainda é ruim. Saldo de que? Seria não ambíguo se não deixasse dúvidas, algo como “Emitir relatório de saldo da conta corrente do cliente pessoa física”.

Requisitos de Software: Requisitos Funcionais

Um Requisito Funcional de qualidade precisa atender alguns atributos específicos:

1. Unidade
2. Completude
3. Consistência
4. Atomicidade
5. Não-ambiguidade
6. Verificável
7. Rastreável
8. Prioridade

Não adianta ter um Requisito Funcional se ele não é palpável, possível de associar com um artefato de construção, de testes. Um Requisito Funcional **tem que ser testável**, tem que ser possível atestar que o Requisito Funcional **foi atendido**, foi construído, foi homologado. Para isso tem que ser também rastreável.

Requisitos de Software: Requisitos Funcionais

Um Requisito Funcional de qualidade precisa atender alguns atributos específicos:

1. Unidade
2. Completude
3. Consistência
4. Atomicidade
5. Não-ambiguidade
6. Verificável
7. Rastreável
8. Prioridade

Deve ser possível achar o Requisito Funcional no sistema **pronto, funcional e executável**. Como saber se um Requisito Funcional foi atendido? Para isso é necessário ter rastreabilidade, e isso só é possível ligando as pontas (associar o Requisito Funcional à interface gráfica, que será associada a um caso de uso, que será associado a funcionalidades, que serão implementadas etc.).

Requisitos de Software: Requisitos Funcionais

Um Requisito Funcional de qualidade precisa atender alguns atributos específicos:

1. Unidade
2. Completude
3. Consistência
4. Atomicidade
5. Não-ambiguidade
6. Verificável
7. Rastreável
8. **Prioridade**

sistema de e-commerce:
adicionar itens ao carrinho e pagar

sistema de e-commerce:
criar listas de desejos

Um Requisito Funcional Essencial é algo muito diferente de um Requisito Funcional Desejável, pois possuem valores para o negócio completamente diferentes. O Requisito Funcional **deve possuir sua prioridade**, isso interfere diretamente no projeto do software.

Requisitos de Software: Requisitos Funcionais

Exemplo:

Identificador	RF0001		
Nome	Consultar automaticamente o CEP de clientes através do endereço residencial		
Módulo	Cadastro		
Data de criação	31/01/2016	Autor	Arquitas de Tarento
Data da última alteração	N/A	Autor	N/A
Versão	1	Prioridade	Desejável
Descrição	<p>Para todos os clientes cadastrados deverá ser possível consultar o CEP do endereço residencial de maneira automática através dos dados de logradouro, número, complemento, bairro, cidade e estado.</p> <p>A consulta se dará após os dados citados serem informados, de maneira transparente para o usuário final. Não deve ser necessário clicar em botão ou acionar algum outro comando para que a consulta ocorra.</p>		

Requisitos de Software: Requisitos NÃO Funcionais

Requisitos NÃO Funcionais são aqueles que não estão diretamente relacionados à funcionalidade de um sistema.

Muitas vezes o usuário nem sabe que são necessários

Alguns exemplos de Requisitos NÃO Funcionais são:

- Responsividade de uma página web
- Protocolos de segurança e criptografia de dados
- Velocidade do carregamento do aplicativo



Requisitos de Software: Requisitos NÃO Funcionais

Para uma melhor organização da especificação e semântica do projeto do software, Requisitos NÃO Funcionais são separados por categorias, conforme o propósito de cada requisito. A seguir, a lista das principais categorias existentes:

1. Desempenho
2. Disponibilidade
3. Segurança
4. Interoperabilidade
5. Usabilidade
6. Compatibilidade
7. Confiabilidade
8. Padrões
9. Legais

Requisitos de Software: Requisitos NÃO Funcionais

Para uma melhor organização da especificação e semântica do projeto do software, Requisitos NÃO Funcionais são separados por categorias, conforme o propósito de cada requisito. A seguir, a lista das principais categorias existentes:

1. Desempenho
2. Disponibilidade
3. Segurança
4. Interoperabilidade
5. Usabilidade
6. Compatibilidade
7. Confiabilidade
8. Padrões
9. Legais

Desempenho do sistema, restrições de performance, tempo de resposta em processamentos específicos, cargas, velocidade de resposta de processamentos em telas etc.

Ex.: Tempo de Resposta: As páginas da web devem ser carregadas em menos de 3 segundos em 95% das solicitações.

Ex.: Tempo de Processamento de Tarefas: O sistema deve processar transações de pagamento em menos de 1 segundo em 99% dos casos.

Requisitos de Software: Requisitos NÃO Funcionais

Para uma melhor organização da especificação e semântica do projeto do software, Requisitos NÃO Funcionais são separados por categorias, conforme o propósito de cada requisito. A seguir, a lista das principais categorias existentes:

1. Desempenho
2. Disponibilidade
3. Segurança
4. Interoperabilidade
5. Usabilidade
6. Compatibilidade
7. Confiabilidade
8. Padrões
9. Legais

Disponibilidade do sistema em tempo útil, restrições sobre janelas de manutenção, janelas de produção, soluções de contorno quando houver queda de energia etc.

Ex.: Tempo de Inatividade Programado: O sistema pode ter até 30 minutos de tempo de inatividade programado por semana, agendado durante as horas de menor uso.

Requisitos de Software: Requisitos NÃO Funcionais

Para uma melhor organização da especificação e semântica do projeto do software, Requisitos NÃO Funcionais são separados por categorias, conforme o propósito de cada requisito. A seguir, a lista das principais categorias existentes:

1. Desempenho
2. Disponibilidade
3. Segurança
4. Interoperabilidade
5. Usabilidade
6. Compatibilidade
7. Confiabilidade
8. Padrões
9. Legais

Diretrizes pertinentes à segurança do sistema, como algoritmo de **criptografia** a ser utilizado, regras para criação e manutenção de usuários e senhas, uso de certificados digitais, uso de **protocolos seguros** específicos, uso de captcha etc.

Ex: Auditoria de Acesso: Todas as tentativas de login bem-sucedidas e malsucedidas devem ser registradas em um log de auditoria, incluindo o horário, o usuário e o endereço IP de origem.

Requisitos de Software: Requisitos NÃO Funcionais

Para uma melhor organização da especificação e semântica do projeto do software, Requisitos NÃO Funcionais são separados por categorias, conforme o propósito de cada requisito. A seguir, a lista das principais categorias existentes:

1. Desempenho
2. Disponibilidade
3. Segurança
4. Interoperabilidade
5. Usabilidade
6. Compatibilidade
7. Confiabilidade
8. Padrões
9. Legais

Necessidades de integração do sistema com outros sistemas, **integração com APIs**, componentes, banco de dados externos
Ex.: Taxa de Sucesso de Integração: O sistema deve ser capaz de integrar-se com outros sistemas externos com uma taxa de sucesso de pelo menos 99% nas transações de dados.

Requisitos de Software: Requisitos NÃO Funcionais

Para uma melhor organização da especificação e semântica do projeto do software, Requisitos NÃO Funcionais são separados por categorias, conforme o propósito de cada requisito. A seguir, a lista das principais categorias existentes:

1. Desempenho
2. Disponibilidade
3. Segurança
4. Interoperabilidade
5. Usabilidade
6. Compatibilidade
7. Confiabilidade
8. Padrões
9. Legais

Facilidade de uso, correção fácil de erros em formulários, quantidade máxima de cliques por tipo de funcionalidade, uso de componentes e lógicas de telas específicas, restrição/premissas para uso de componentes gráficos (grids, barras de rolagem, menus), recursos de acessibilidade para deficientes, compatibilidade com idiomas etc.

EX. : Taxa de Erros de Usuário: A taxa de erros de entrada de dados por parte dos usuários não deve exceder 2% durante a interação com o sistema.

Requisitos de Software: Requisitos NÃO Funcionais

Para uma melhor organização da especificação e semântica do projeto do software, Requisitos NÃO Funcionais são separados por categorias, conforme o propósito de cada requisito. A seguir, a lista das principais categorias existentes:

1. Desempenho
2. Disponibilidade
3. Segurança
4. Interoperabilidade
5. Usabilidade
6. Compatibilidade
7. Confiabilidade
8. Padrões
9. Legais

Browser e sistemas operacionais nos quais o software deverá rodar, versões de browser e sistemas operacionais, protocolos compatíveis, versões de linguagens de programação e banco de dados para retrocompatibilidade etc.

Ex.: Compatibilidade de Navegador: O sistema deve ser compatível com os seguintes navegadores da web: Google Chrome (versão 80 ou superior), Mozilla Firefox (versão 75 ou superior), Microsoft Edge (versão 80 ou superior), Safari (versão 13 ou superior).

Requisitos de Software: Requisitos NÃO Funcionais

Para uma melhor organização da especificação e semântica do projeto do software, Requisitos NÃO Funcionais são separados por categorias, conforme o propósito de cada requisito. A seguir, a lista das principais categorias existentes:

1. Desempenho
2. Disponibilidade
3. Segurança
4. Interoperabilidade
5. Usabilidade
6. Compatibilidade
7. **Confiabilidade**
8. Padrões
9. Legais

Políticas para **backup** do sistema e seus dados, quantidade limite de erros em cálculos e processamentos com erro, regras para **rollback** quando houver alguma falha, recursos para restauração automática do sistema em caso de queda de energia etc.

Ex.: Backup e Recuperação de Dados: O sistema deve realizar backups automáticos dos dados a cada 24 horas e ser capaz de restaurar dados de backup em menos de 1 hora, em caso de falha ou perda de dados.

Requisitos de Software: Requisitos NÃO Funcionais

Para uma melhor organização da especificação e semântica do projeto do software, Requisitos NÃO Funcionais são separados por categorias, conforme o propósito de cada requisito. A seguir, a lista das principais categorias existentes:

1. Desempenho
2. Disponibilidade
3. Segurança
4. Interoperabilidade
5. Usabilidade
6. Compatibilidade
7. Confiabilidade
8. Padrões
9. Legais

Padrões em geral aplicáveis ao software e ao projeto: padrão de log de erro, de log de informação, padrão de mensagens, metodologia para desenvolvimento do sistema, padrões de projeto (design patterns) a serem aplicados, padrões arquiteturais etc.

Ex.: O sistema deve ser compatível com padrões de interoperabilidade reconhecidos internacionalmente, como XML, JSON, MQTT, etc., e ser capaz de trocar dados com outros sistemas que seguem os mesmos padrões.

Requisitos de Software: Requisitos NÃO Funcionais

Para uma melhor organização da especificação e semântica do projeto do software, Requisitos NÃO Funcionais são separados por categorias, conforme o propósito de cada requisito. A seguir, a lista das principais categorias existentes:

1. Desempenho
2. Disponibilidade
3. Segurança
4. Interoperabilidade
5. Usabilidade
6. Compatibilidade
7. Confiabilidade
8. Padrões
9. Legais

Exigências de conformidade do software com **alguma legislação pertinente ao projeto**, por exemplo, atendimento a alguma norma da Agência Nacional de Saúde para software de hospital, a norma do Banco Central para sistemas financeiros etc.

O sistema deve respeitar os direitos de propriedade intelectual de terceiros e não infringir patentes, direitos autorais ou marcas registradas. A conformidade pode ser medida através de revisões legais e relatórios de conformidade.

Requisitos de Software: Requisitos NÃO Funcionais

Exemplo:

Identificador	RNF05	Categoria	Usabilidade
Nome	Uso de design responsivo nas interfaces gráficas		
Data de criação	30/01/2016	Autor	Diógenes de Apolônia
Data da última alteração	N/A	Autor	N/A
Versão	1	Prioridade	Importante
Descrição	<p>O sistema de Atendimento a Clientes será construído para rodar em ambiente web. Deverá possuir um design responsivo (https://en.wikipedia.org/wiki/Responsive_web_design).</p> <p>A interface do sistema deverá se comportar adequadamente independente do front-end que será utilizado para acesso – Browser, Smartphone ou Tablet.</p> <p>Obs.: durante o processo de homologação do sistema serão realizados testes de usabilidade validando este requisito. O não atendimento a este requisito gerará o não pagamento relativo</p>		

Backlog do Produto

O Backlog do Produto é uma lista priorizada de itens sobre os quais o Time de Desenvolvimento trabalhará no decorrer do projeto. Trata-se da lista de funcionalidades e requisitos que deverão ser entregues ao cliente ao longo das Sprints.

Podemos dizer que Backlog do produto é uma lista que contém os requisitos necessários para a construção de um produto de alto valor. Ou seja, trata-se de uma lista de entregas do projeto organizada em ordem de prioridade.

Como estruturar o Backlog do Produto

1) Levantar requisitos

A primeira etapa para a elaboração de um backlog de produto é o levantamento de requisitos. Uma boa maneira de fazer isso é utilizando histórias dos usuários.

As histórias dos usuários mostram as principais necessidades do produto, para quem, por que e o que deve ser desenvolvido. Essa etapa é importante porque ajuda agregar mais valor ao produto e diminuir as chances de inserir tarefas inúteis ou dispensáveis.

Como estruturar o Backlog do Produto

2) Defina critérios de priorização

Antes de tudo, é essencial definir a ordem de prioridade dos itens no backlog, ou seja, o grau de importância de cada um.

Não existe uma fórmula ou ordem correta para priorizar os itens do backlog. Você pode sempre reordená-los de acordo com o feedback do cliente. Porém, existem alguns conceitos iniciais que podem te ajudar nessa função, como:

Como estruturar o Backlog do Produto

- Entender quais itens terão mais impacto no projeto, ou seja, maior valor;
- Analisar quais itens são mais fáceis de concretizar;
- Analisar quais itens geram mais receita de imediato;
- Ter em mente que é o cliente que diz o que é sucesso ou não. É ele que determina as prioridades dos itens restantes no backlog junto com o Product Owner;
- Criar funcionalidades apenas quando elas acrescentarem valor ao produto.
- Lembre-se que as prioridades mudam de acordo com as necessidades e o momento. Por isso, a lista do backlog é dinâmica e nunca está totalmente completa.

Como estruturar o Backlog do Produto

3) Dê um nome/título aos itens

Nomear os itens é muito importante para que a equipe consiga identificar do que se trata. É preferível dar um nome curto e claro aos itens do backlog. Isso ajuda o time a compreender melhor o assunto em questão e diferenciar uma tarefa das outras.

Alguns exemplos a seguir:

- “Testar a funcionalidade x”
- “Distribuir pesquisa”
- “Implantar a funcionalidade x”.

Como estruturar o Backlog do Produto

4) Escreva uma descrição

Nesse momento é hora de detalhar o item criado. É muito importante que a descrição contenha todas as informações da tarefa, explicadas de uma forma clara, objetiva e compreensível.

Na descrição pode conter:

- Referências para a realização da atividade;
- Informações necessárias para o desenvolvimento da tarefa;
- Sugestões;
- Explicações.

Como estruturar o Backlog do Produto

5) Defina um prazo

Agora você precisa definir qual o prazo para cada item ser entregue. Para isso, pode-se analisar o contexto do produto em geral e as metas que foram discutidas nas reuniões de Sprint.

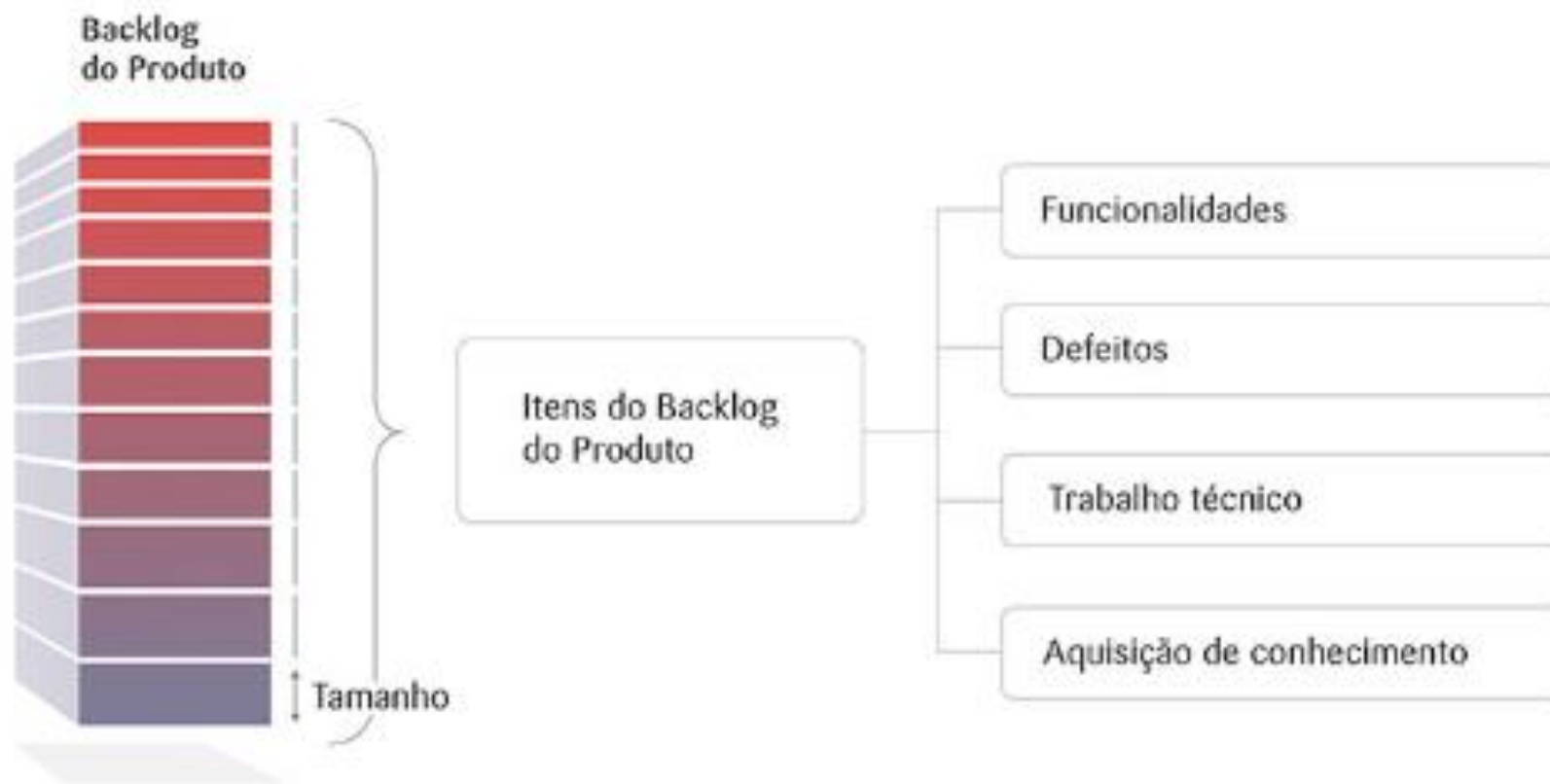
Defina informações como início estimado, término estimado e esforço (em horas) estimado.

Como estruturar o Backlog do Produto

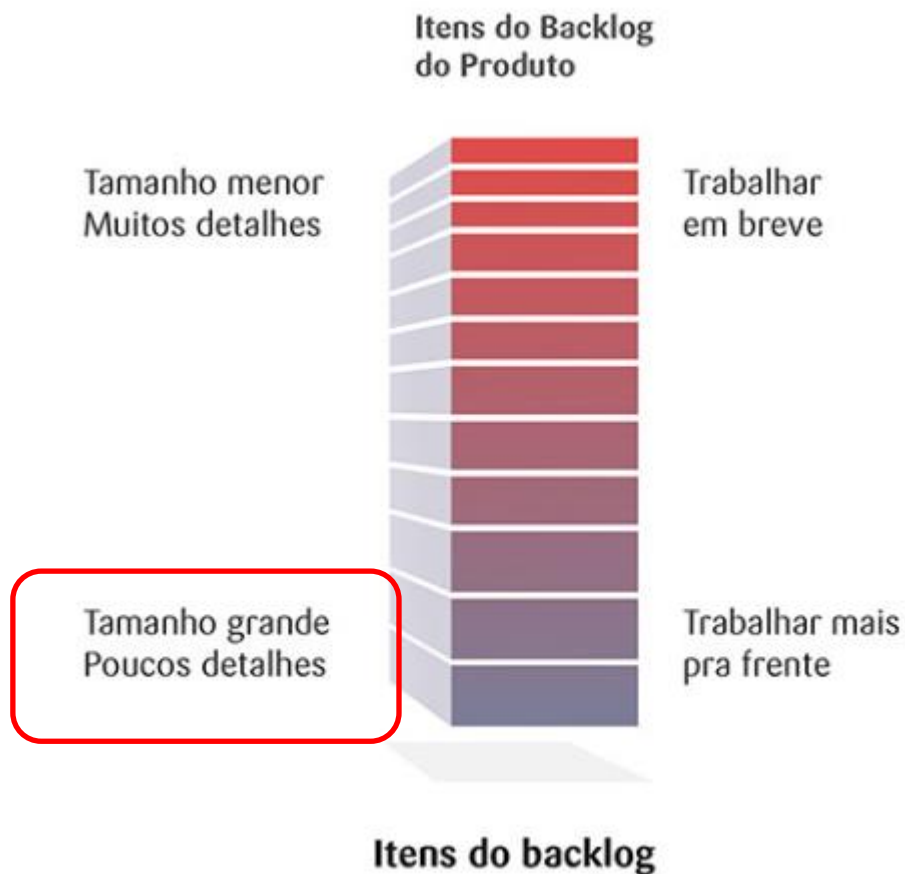
Atributos que um bom backlog do produto deve conter

- Visibilidade clara e compreensível;
- Organização;
- Possuir ordem bem definida;
- Possuir tarefas bem categorizadas;
- Ser detalhado adequadamente.

Como estruturar o Backlog do Produto

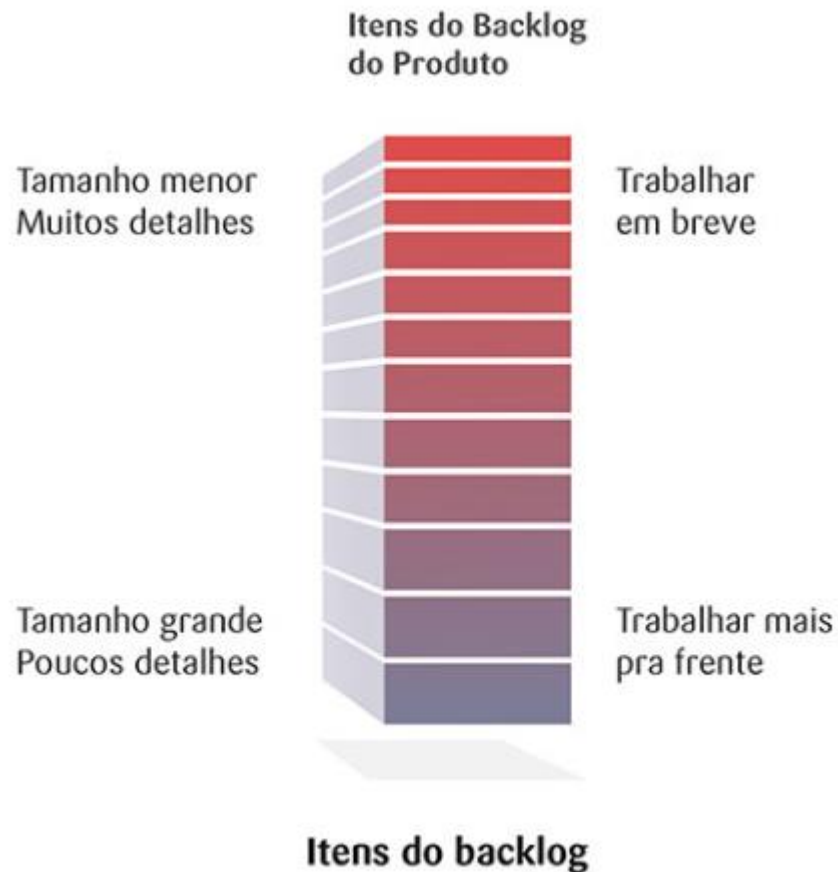


Como estruturar o Backlog do Produto



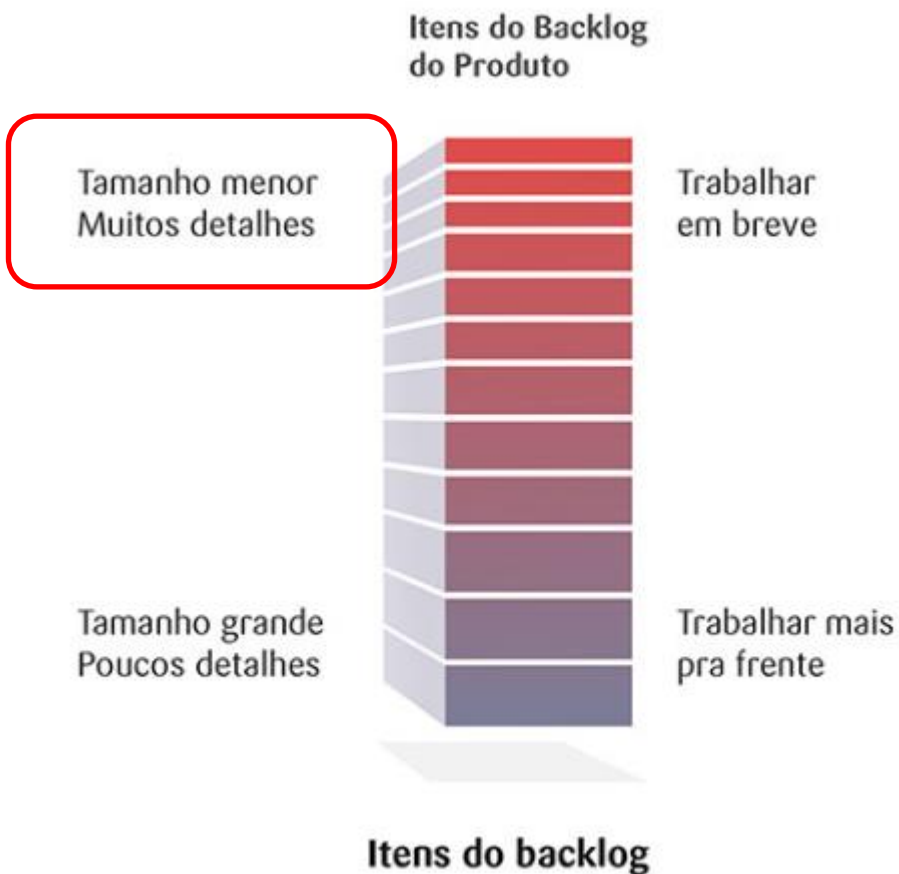
Um **item grande** é comumente chamado de épico ou tema. Possui uma entrega de valor clara, ainda que pouco detalhada, e algumas hipóteses quanto à solução que deve gerar. Nessa etapa, não se sabe "como" ele será construído nem do que ele é composto;

Como estruturar o Backlog do Produto



Um **item médio** é chamado de funcionalidade ou recurso. Nessa fase já há um maior conhecimento sobre a entrega de valor e de "como" ela será construída. Também constroem-se protótipos para validar as ideias e depois desdobrá-las em pedaços menores;

Como estruturar o Backlog do Produto

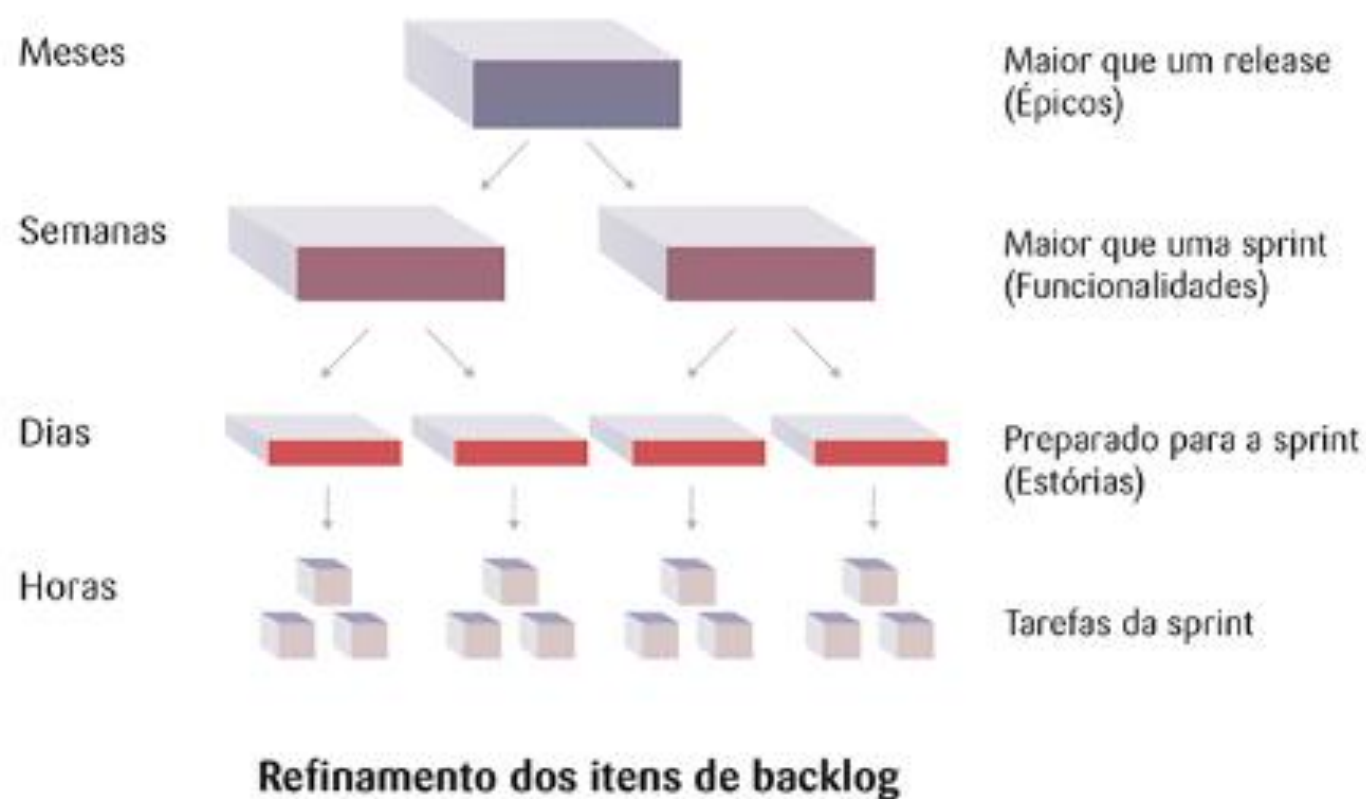


Os **itens pequenos** são chamados de estórias de usuário – são itens que estão prontos para entrar em desenvolvimento. Toda a equipe técnica participa da preparação desses itens, pois é o time que compreende a complexidade e o tempo de trabalho que as tarefas de desenvolvimento exigirão.

Como estruturar o Backlog do Produto

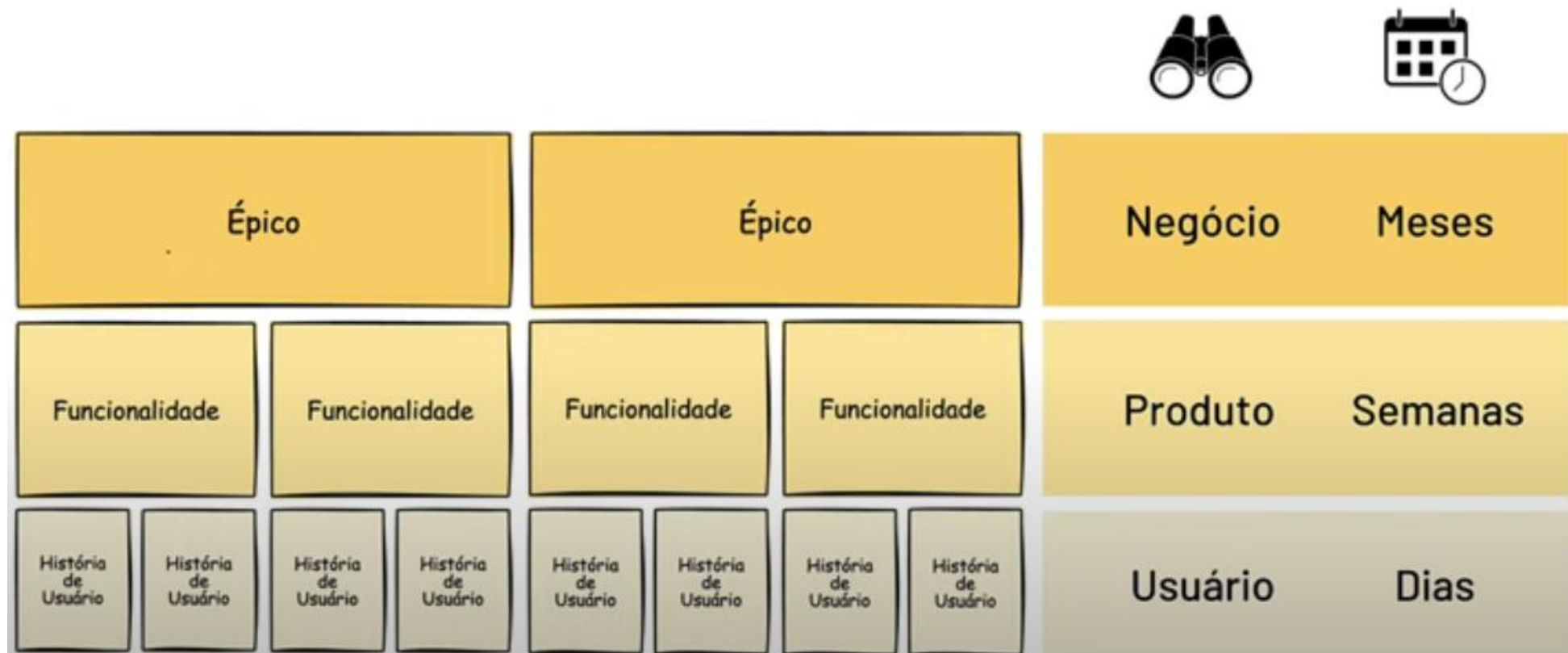
REFINAMENTO DO BACKLOG

Ao lado uma representação do que é chamado de refinamento do backlog:



Como estruturar o Backlog do Produto

Épicos representam os objetivos de alto nível, funcionalidades detalham partes significativas do produto e histórias de usuários descrevem requisitos específicos do usuário em termos de funcionalidades desejadas. A seguir, representação do que é chamado de refinamento do backlog:



Exemplo de Épico, funcionalidade e história de usuário

Produto:
Shopping



Épicos do Shopping:

- Praça de Alimentação



- Praça de compras



- Estacionamento



Funcionalidades da Praça de Alimentação:

- Padaria



- Cafeteria



- Doceria



Histórias de Usuários da Padaria:

- Visualizar produto



- Consultar preço



- Pagar produto

