

Processos de software

Elton Morais



Objetivos

Introduzir o conceito de processo de software como um conjunto coerente de atividades para produção de software.



Objetivos

- **Compreender** os conceitos e os modelos de processo de software;
- **Apresentar** três modelos genéricos de processo de software e às situações nas quais eles podem ser utilizados;
- **Conhecer** as atividades de processo fundamentais de engenharia de requisitos, do desenvolvimento, dos testes e da evolução de software;



Objetivos

- **Compreender** por que os processos devem ser organizados para lidar com as mudanças nos requisitos e no projeto de software;
- **Compreender** o conceito de melhoria do processo de software e os fatores que afetam a qualidade do processo;



Introdução

Um processo de software é um **conjunto de atividades relacionadas** que levam à produção de um sistema de software.



Introdução

- Assim como **não** há um **método universal de engenharia de software** que seja aplicável a todos os tipos diferentes de sistemas, também **não** existem **processos de software universalmente aplicáveis**;
- O processo utilizado depende do **tipo de software** que está sendo desenvolvido, dos **requisitos** e das habilidades das **pessoas** que o desenvolvem;



Introdução

- Embora existam muitos processos, eles devem incluir:
 - Especificação;
 - Desenvolvimento;
 - Validação;
 - Evolução;
- Essas atividades se desdobram em **subatividades** como validação dos requisitos, gerenciamento de configuração, teste de unidades, planejamento etc.;



Introdução

- Quando se trata de processos de software, é importante descrever **quem** está envolvido, **o que** está sendo produzido e **quais** condições influenciam a sequência de atividades;
- Uma atividade de processo resulta em produtos ou em entregas;
- Os papéis refletem as responsabilidades das pessoas envolvidas no processo;
- Há condições que devem ser mantidas antes ou depois de uma atividade do processo ter sido aprovada ou um produto ter sido produzido.



Modelos de Processo de Software

Um modelo de processo de software, ou ciclo de vida do desenvolvimento de software – SDLC (*Software Development Life Cycle*) é uma representação simplificada de um processo de software.



Modelos de Processo de Software

- Modelos de processos:
 - Modelo em cascata;
 - Desenvolvimento incremental;
 - Integração e configuração;



Modelos de Processo de Software

- **Modelo em cascata** – representa as atividades fundamentais do processo, como especificação, desenvolvimento, validação e evolução, na forma de fases de processo distintas, como especificação de requisitos, projeto de software, implementação e testes;
- **Desenvolvimento incremental** – intercala as atividades de especificação, desenvolvimento e validação. O sistema é desenvolvido como uma série de versões (incrementos), com cada uma delas acrescentando funcionalidade à versão anterior;



Modelos de Processo de Software

- **Integração e configuração** – baseia-se na disponibilidade de componentes ou sistemas reusáveis. O processo de desenvolvimento de sistemas se concentra na configuração desses componentes, para que sejam utilizados em um novo contexto, e na integração deles em um sistema;



Modelos de Processo de Software

- Como **não** existe modelo de processo universal aplicável, o processo correto depende do cliente e dos requisitos do software, do ambiente em que esse software será utilizado e do tipo de software que está sendo desenvolvido;
- Exemplo:
 - Sistemas críticos em segurança – normalmente são desenvolvidos a partir de um processo em cascata;
 - Produtos de software – atualmente são desenvolvidos a partir de um modelo incremental;
 - Sistemas de negócio – cada vez mais, são desenvolvidos por meio de configuração dos sistemas preexistentes e da integração entre eles, criando um novo sistema;



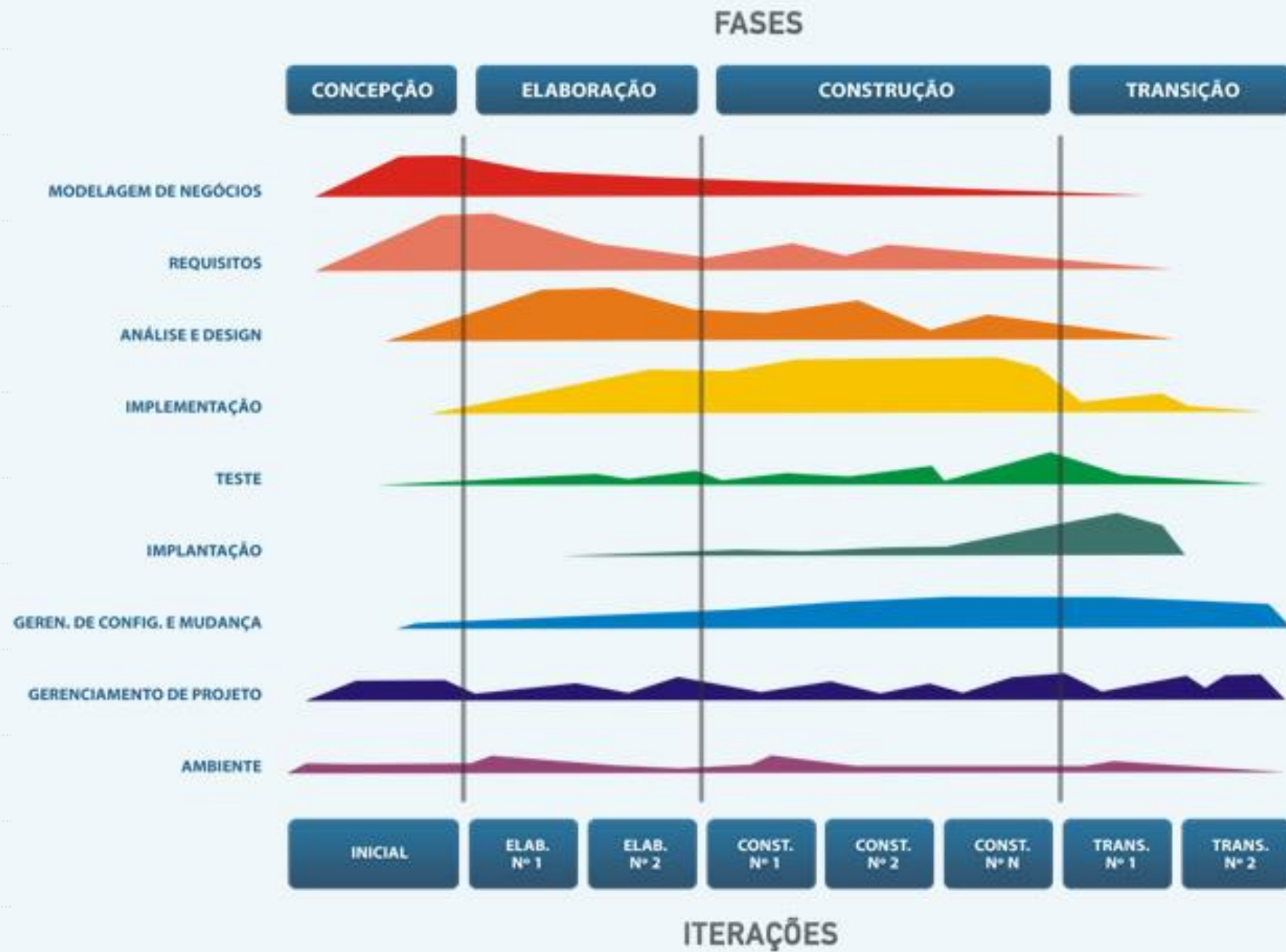
Modelos de Processo de Software

- Há tentativas de desenvolver modelos de processo universais baseados em todos os modelos genéricos;
- O mais conhecido deles é o *Rational Unified Process* – RUP;
- RUP possui três perspectivas:
 - Dinâmica – mostra as fases do modelo no tempo;
 - Estática – mostra as atividades do processo;
 - Prática – sugere práticas a serem utilizadas no processo;



Modelos de Processo de Software

- RUP possui quatro fases:
 - Concepção – elaboração do planejamento do projeto com os stakeholders;
 - Elaboração – busca modelos para orientar o projeto, como documentação, casos etc.;
 - Construção – construção do projeto;
 - Transição – fase do projeto em que passa do ponto de testes para a implementação;





0 modelo em cascata

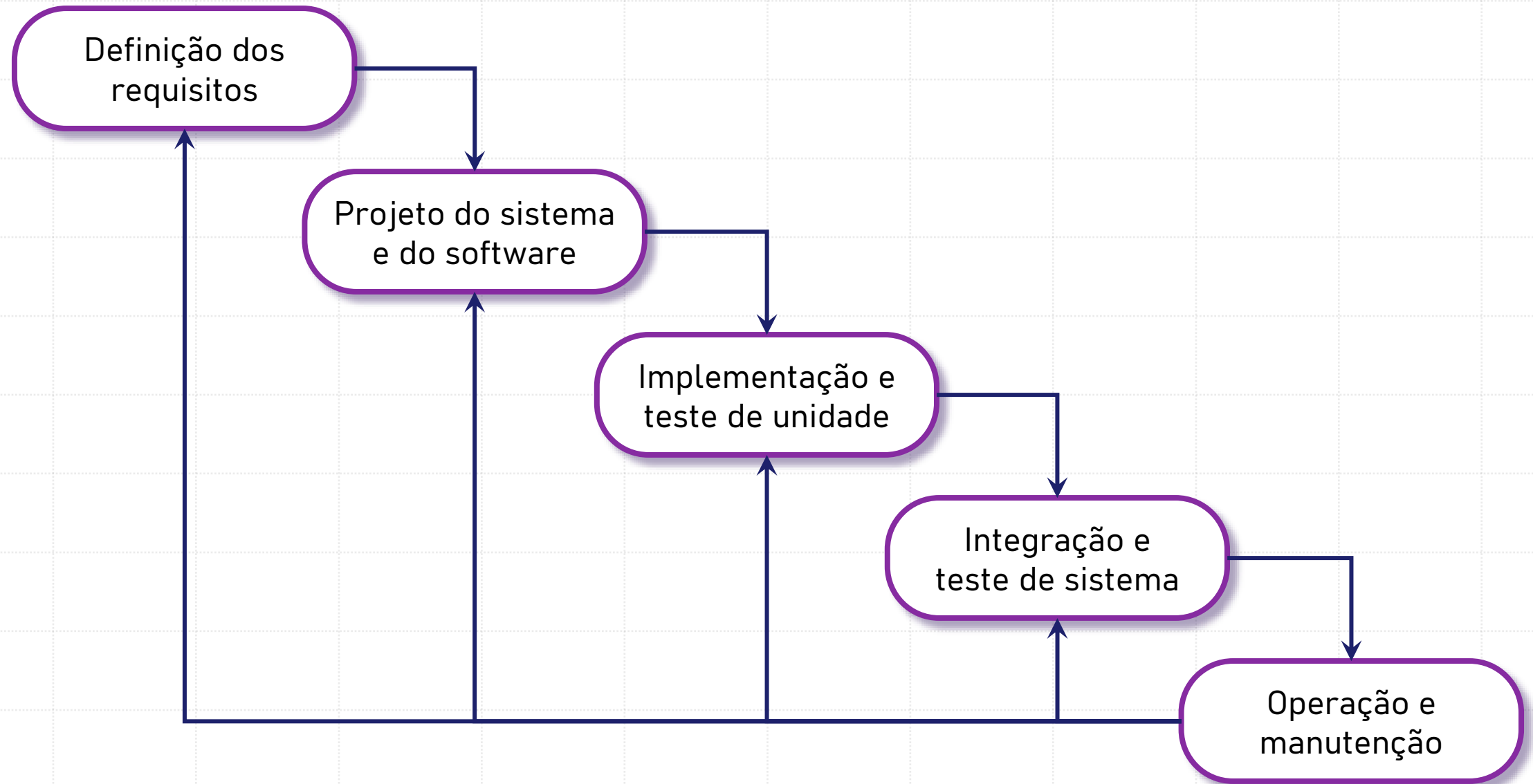
Modelos de Processo de Software



O modelo em cascata

- É um modelo em que o processo é linear e sequencial, dividido em fases distintas;
- **Vantagens** dessa abordagem:
 - Simplicidade;
 - Documentação abrangente;
 - Controle;
- **Desvantagens** dessa abordagem:
 - Inflexibilidade;
 - Feedback tardio;
 - Risco de atrasos;

O modelo em cascata.





O modelo em cascata

- Os estágios do modelo em cascata:
 - **Análise e definição dos requisitos** – os serviços, as restrições e as metas do sistema são estabelecidos por meio de consulta aos usuários;
 - **Projeto do sistema e do software** – o processo de projeto do sistema reparte os requisitos entre requisitos de sistemas de hardware e de software, e estabelece uma arquitetura global do sistema;
 - **Implementação e teste de unidades** – desenvolvimento das unidades de programa e verificação de cada unidade, afim de satisfazer sua especificação;



O modelo em cascata

- Os estágios do modelo em cascata:
 - ***Integração e teste de sistema*** – as unidades de programa são integradas e testadas como um sistema completo;
 - ***Operação e manutenção*** – fase mais longo, onde o sistema é instalado e colocado em uso;

O modelo em cascata



Project

Criação de cronogramas, atribuição de tarefas e monitora o progresso.



Jira

Facilita a gestão de projetos ágeis, mas pode ser adaptado para o modelo em cascata.



Trello

Organiza as tarefas em quadros. É intuitivo e flexível, adaptando-se as diversas metodologias.



O modelo em cascata

- O modelo cascata é mais adequado para:
 - Sistemas embarcados, nos quais o software deve interagir com sistemas de hardware;
 - Sistemas críticos, nos quais há necessidade de ampla análise da segurança e segurança da informação da especificação e do projeto;
 - Grandes sistemas, que fazem parte de sistemas mais amplos, desenvolvidos por várias empresas parceiras.



Desenvolvimento incremental

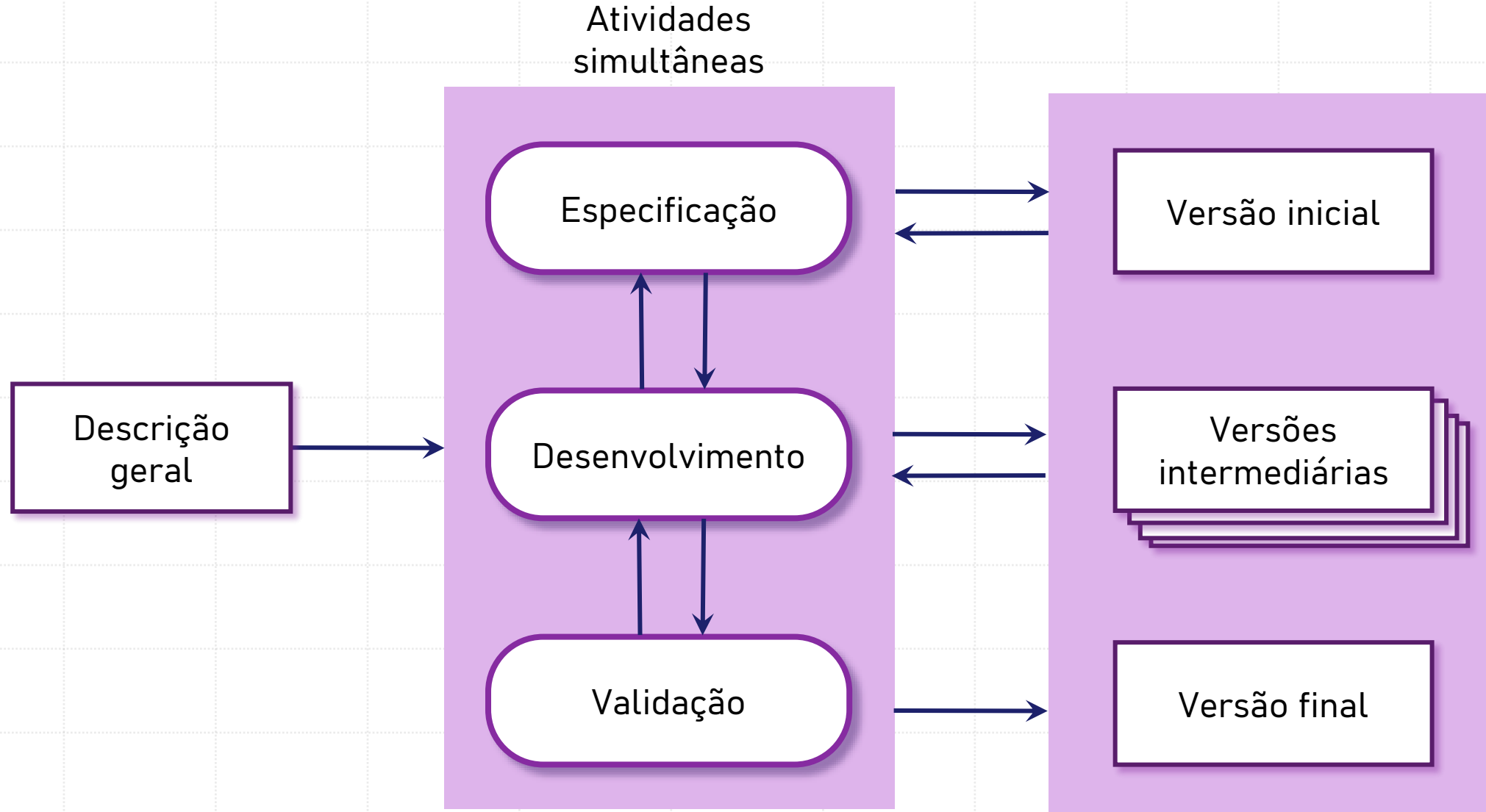
Modelos de Processo de Software



Desenvolvimento incremental

- Baseia-se na ideia de desenvolver uma **implementação inicial**, obter *feedback* dos usuários ou terceiros e fazer o software **evoluir através de várias versões**;
- As atividades de especificação, desenvolvimento e validação são intercaladas, em vez de separadas, com *feedback* rápido ao longo de todas elas;

Desenvolvimento incremental





Desenvolvimento incremental

- É a abordagem **mais comum** para desenvolvimento de aplicações, podendo ser dirigida por **plano**, **ágil** ou **ambos**;
- Na dirigida por plano, os **incrementos** do sistema são identificados **antecipadamente**;
- Na abordagem ágil, os **incrementos iniciais** são identificados, mas o desenvolvimento dos incrementos finais depende do **progresso** e da **prioridades** do cliente;
- O desenvolvimento incremental é mais **barato** e **fácil** de fazer alterações durante o processo de desenvolvimento;



Desenvolvimento incremental

- **Vantagens** do desenvolvimento incremental em relação ao modelo em cascata:
 - O custo de implementação das mudanças nos requisitos é reduzido;
 - É mais fácil obter *feedback* do cliente sobre o trabalho de desenvolvimento;
 - A entrega e a implantação antecipadas de um software útil para o cliente são possíveis, mesmo se toda a funcionalidade não tiver sido incluída;



Desenvolvimento incremental

- **Problemas** do desenvolvimento incremental:
 - Para grandes organizações, com procedimentos mais burocráticos, pode haver uma incompatibilidade entre esses procedimentos e um processo iterativo ou ágil mais informal;
 - Os gerentes precisam de resultados regulares para medir o progresso;
 - A estrutura do sistema tende a se degradar à medida que novos incrementos são adicionados;

Desenvolvimento incremental



Controle de versão que facilita a colaboração e integração contínua.



Facilita a gestão de projetos ágeis, mas pode ser adaptado para o modelo em cascata.



Organiza as tarefas em quadros. É intuitivo e flexível, adaptando-se as diversas metodologias.



Integração e configuração

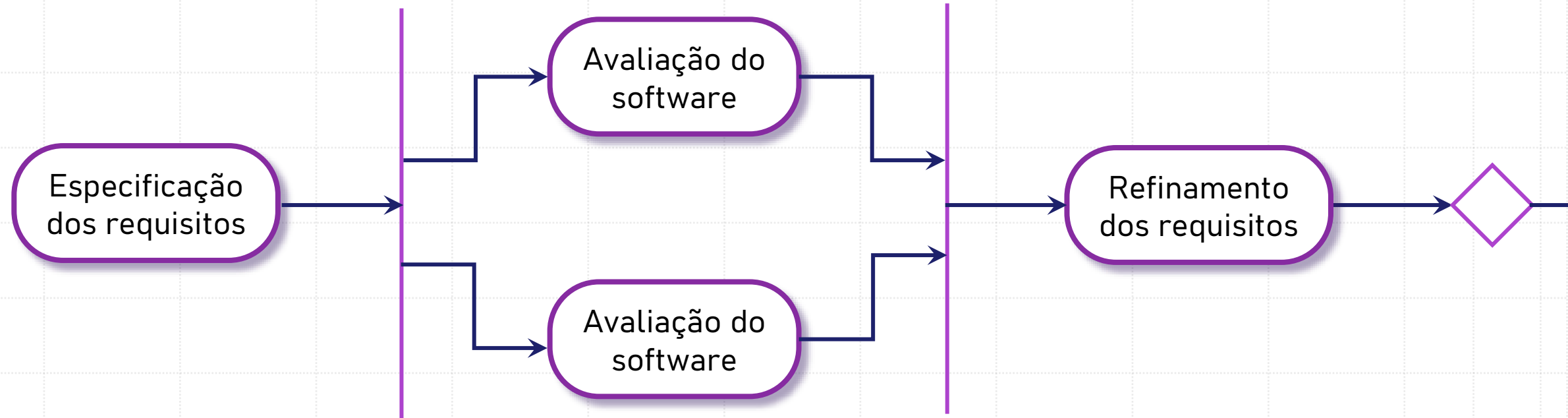
Modelos de Processo de Software



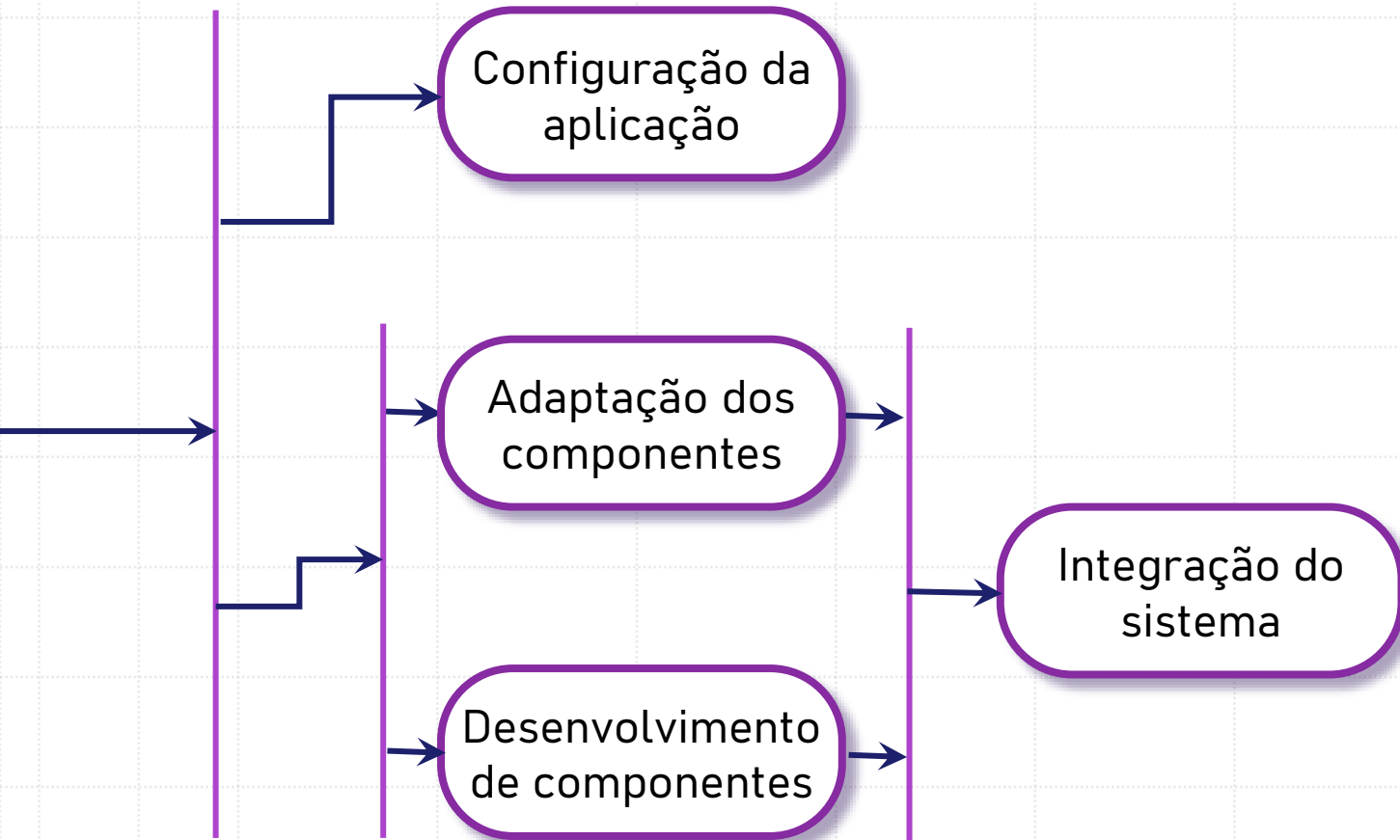
Integração e configuração

- As abordagens orientadas ao reuso contam com bases de componentes de software reusáveis e um *framework* de integração para a composição desses componentes;
- Três tipos de componentes de software são reusados frequentemente:
 - Sistemas de aplicação *stand-alone* configurados para utilização em um ambiente particular;
 - Coleções de objetos desenvolvidos como um componente ou como um pacote a ser intergrdo a um *framework* de componentes;
 - *Web services* desenvolvidos de acordo com os padrões de serviço e que estão disponíveis para uso remoto na internet;

Engenharia de software orientada para o reuso.



Engenharia de software orientada para o reuso.





Integração e configuração

- **Vantagens:**

- Reduz a quantidade de software a ser desenvolvido;
- Diminui custos e riscos;
- Normalmente, levam a entregas mais rápidas;

- **Desvantagens:**

- Pode não resultar nas necessidades reais dos usuários;
- A evolução do sistema se perde, já que os componentes não estão sob o controle da organização que os utiliza;

Integração e configuração

JUnit



Jenkins



Framework para
testes unitários
em Java.

Servidor de
automação de
código aberto.

Facilita o deploy e
o gerenciamento
de containers.



Atividades do Processo

Os processos de softwares são sequências intercaladas de **atividades técnicas, colaborativas e gerenciais**, cujo objetivo global é especificar, projetar, implementar e testar um sistema de software.

Atividades do Processo

- Esse processo é apoiado por ferramentas de gerenciamento, editores de projeto e de programas, ferramentas de testes e depuradores.

Gestão de Projeto

 **Trello**

 **Jira**


asana

 **Project**

Atividades do Processo

- Esse processo é apoiado por ferramentas de gerenciamento, editores de projeto e de programas, ferramentas de testes e depuradores.

Controle de Versão



Atividades do Processo

- Esse processo é apoiado por ferramentas de gerenciamento, editores de projeto e de programas, ferramentas de testes e depuradores.

Colaboração e Comunicação



Google Meet



Atividades do Processo

- Esse processo é apoiado por ferramentas de gerenciamento, editores de projeto e de programas, ferramentas de testes e depuradores.

Desenvolvimento e IDEs



Visual Studio Code



IntelliJ IDEA



Android Studio

Atividades do Processo

- Esse processo é apoiado por ferramentas de gerenciamento, editores de projeto e de programas, ferramentas de testes e depuradores.

Controle de Qualidade

TestNG

JUnit 5

 Selenium⁴

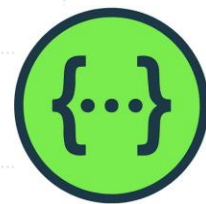
 POSTMAN

 Jenkins

Atividades do Processo

- Esse processo é apoiado por ferramentas de gerenciamento, editores de projeto e de programas, ferramentas de testes e depuradores.

Documentação



SwaggerTM

Atividades do Processo

- Esse processo é apoiado por ferramentas de gerenciamento, editores de projeto e de programas, ferramentas de testes e depuradores.

Gerenciamento de Configuração



ANSIBLE



kubernetes



puppet



docker



Especificação do software

Atividades do Processo

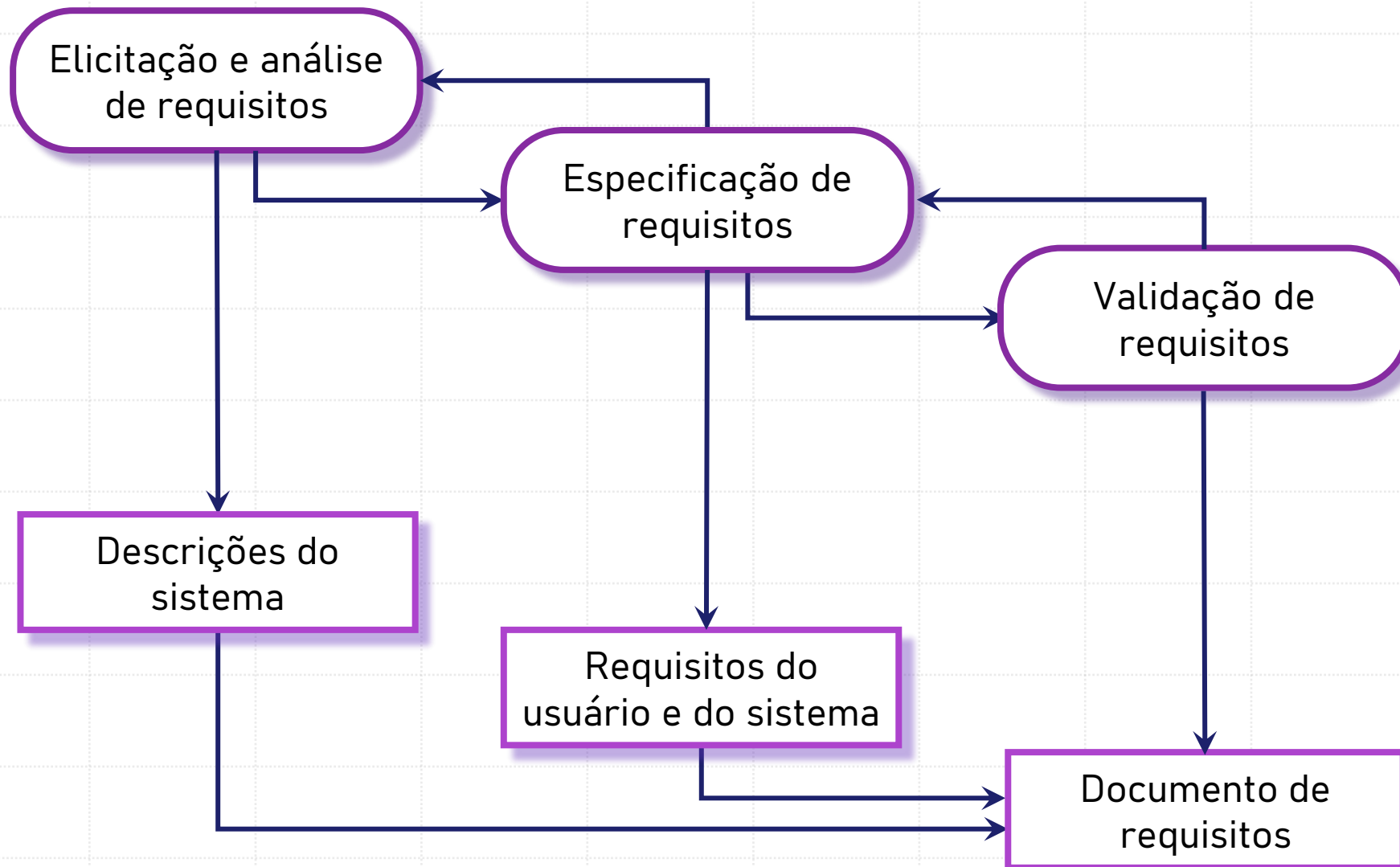
Próxima aula a partir daqui



Especificação do software

- Especificação do software ou **engenharia de requisitos** é o processo de **compreender** e **definir** quais serviços são necessários para o sistema e **identificar** as **restrições** sobre sua operação e desenvolvimento;
- É um **estágio crítico**, já que erros nessa etapa, inevitavelmente geram problemas posteriores;
- Visa à produção de um **documento de requisitos** acordados que especifique um sistema que satisfaça os requisitos dos *stakeholders*.

O processo de engenharia de requisitos.





Projeto e implementação do software

Atividades do Processo



Projeto e implementação do software

- O estágio de implementação no desenvolvimento de software é o processo de elaborar um sistema executável para ser entregue ao cliente;
- O projeto de software é uma descrição da estrutura do software a ser implementado, dos modelos e estruturas de dados utilizados pelo sistema, das interfaces entre os componentes do sistema e, às vezes, do algoritmo utilizado;

Um modelo geral do processo de projeto.

Entradas de projeto

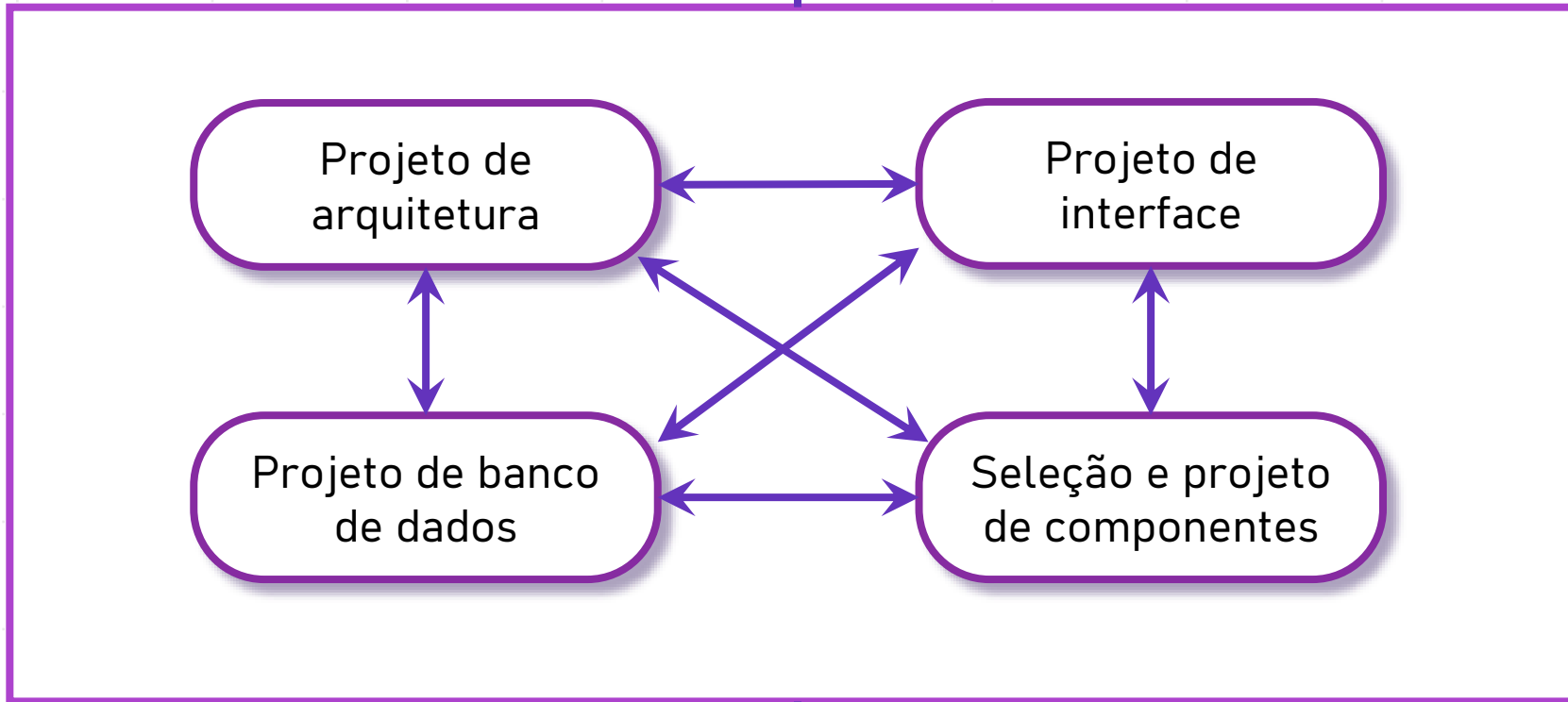
Informações sobre
a plataforma

Requisitos do
software

Descrição dos
dados



Atividades de projeto



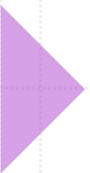
Saídas do projeto

Arquitetura do
sistema

Projeto de banco
de dados

Especificação da
interface

Descrição dos
componentes



Validação do software

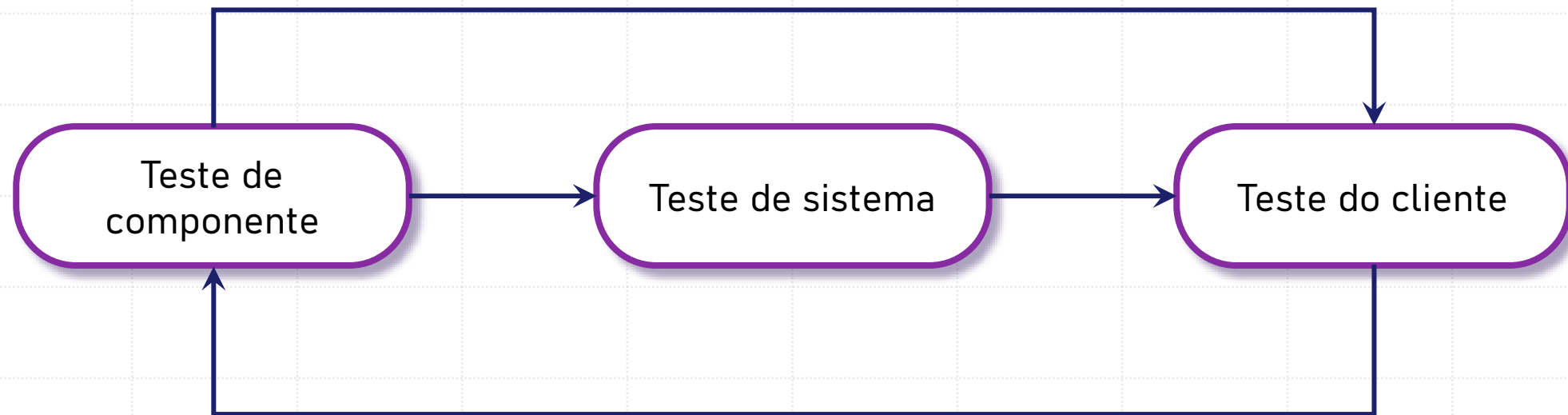
Atividade do Processo



Validação do software

- A validação do software ou verificação e validação destina-se a mostrar que um sistema está em conformidade com sua especificação e que satisfaz as expectativas do cliente;
- A validação pode ocorrer por teste de programa ou conferência, como inspeções e revisões em cada estágio do processo;

Estágios do teste.

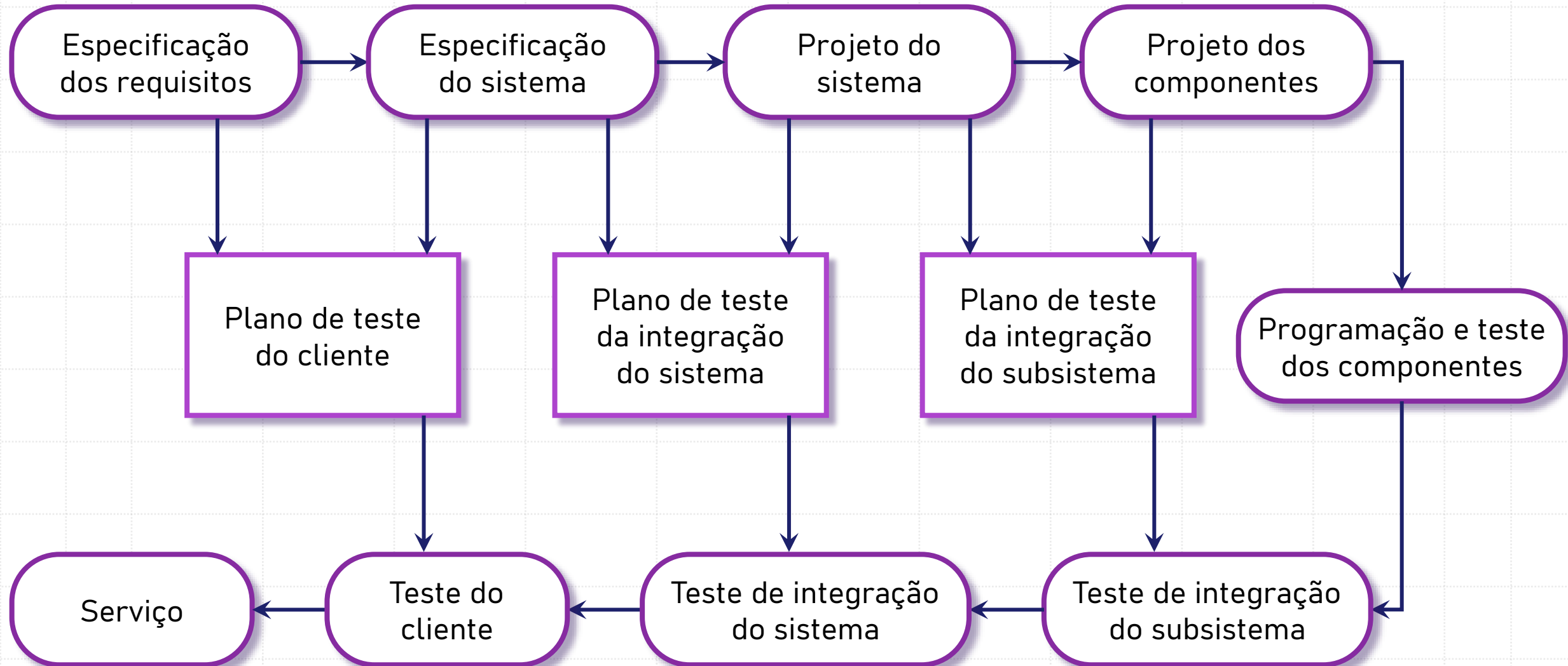




Validação do software

- A medida que os defeitos são descobertos, o programa deve ser depurado;
- Isso pode exigir que outros estágios no processo de teste sejam repetidos;
- Na abordagem incremental, cada incremento deve ser testado enquanto é desenvolvido;
- No desenvolvimento dirigido por testes, eles são desenvolvidos junto com os requisitos, antes do início do desenvolvimento;
- No dirigido por plano, uma equipe independente testam seguindo esses planos de testes;

Fases de teste em um processo de software dirigido por plano.





Evolução do software

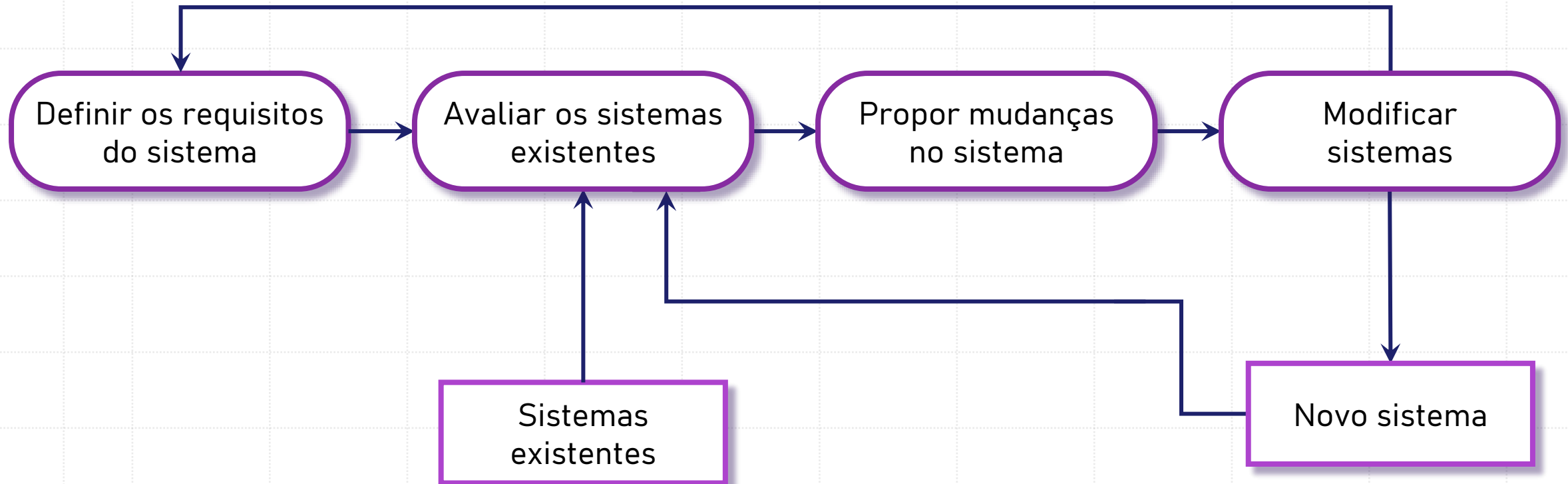
Atividade do Processo



Evolução do software

- Alterações no software podem ser feitas a qualquer momento, durante ou depois do desenvolvimento do sistema;
- O desenvolvimento e a manutenção, atualmente, são praticamente uma coisa só, considerando como um processo evolutivo, no qual o software é alterado continuamente ao longo de sua vida útil;

Evolução do sistema de software.





Lidando com Mudanças

A mudança é inevitável em todos os grandes projetos de software.



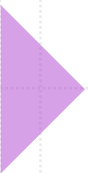
Lidando com Mudanças

- Os requisitos mudam à medida que as empresas reagem a pressões externas;
- Novas tecnologias são disponibilizadas;
- A mudança eleva os custos de desenvolvimento de software, já que normalmente significa retrabalho;
- Duas abordagens podem ser utilizadas para reduzir os custos de retrabalho:
 - Antecipação da mudança;
 - Tolerância à mudança



Lidando com Mudanças

- As mudanças e variações de requisitos do sistema podem ser percebidos através:
 - **Prototipação do sistema** – uma versão ou parte do sistema é desenvolvida para validar requisitos e avaliar a viabilidade;
 - **Entrega incremental** – são entregues ao clientes incrementos, permitindo experimentações e comentários;



Prototipação

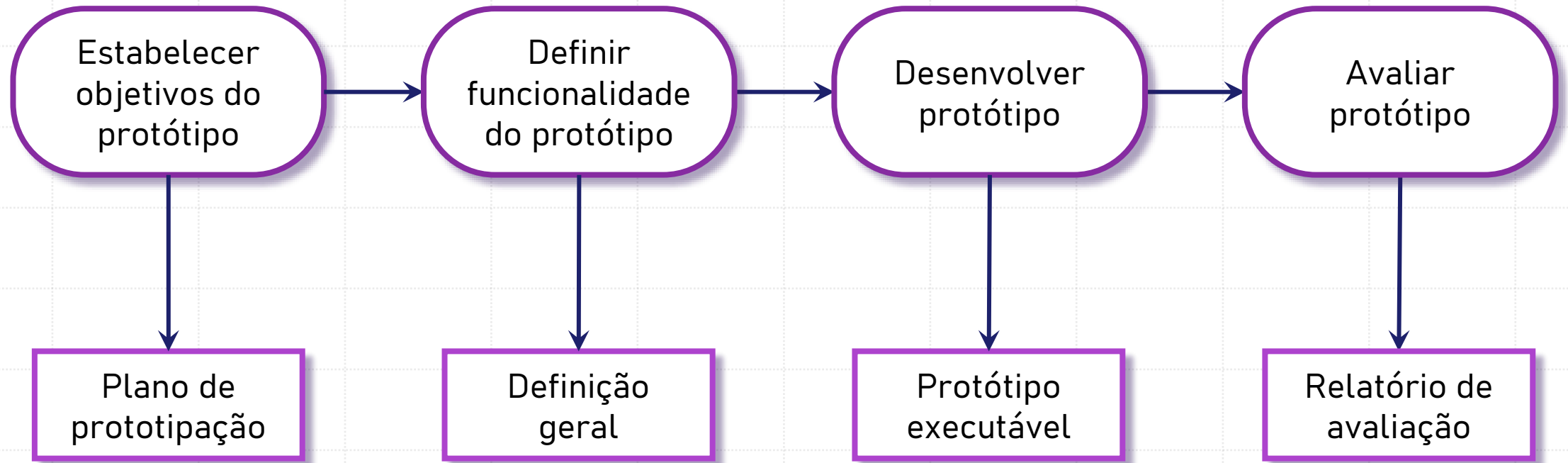
Lidando com Mudanças

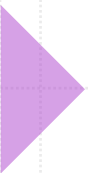


Prototipação

- É uma versão inicial de um sistema utilizado para demonstrar conceitos, experimentar opções de projeto e descobrir possíveis problemas e suas possíveis soluções;
- Pode ser utilizado em um processo de desenvolvimento para ajudar a antecipar as mudanças que podem ser necessárias:
 - Na engenharia de requisitos – um protótipo pode ajudar na elicitação e validação;
 - No projeto do sistema – um protótipo pode ser utilizado para explorar soluções e no desenvolvimento de uma interface com usuário do sistema;

Desenvolvimento de um protótipo.





Entrega incremental

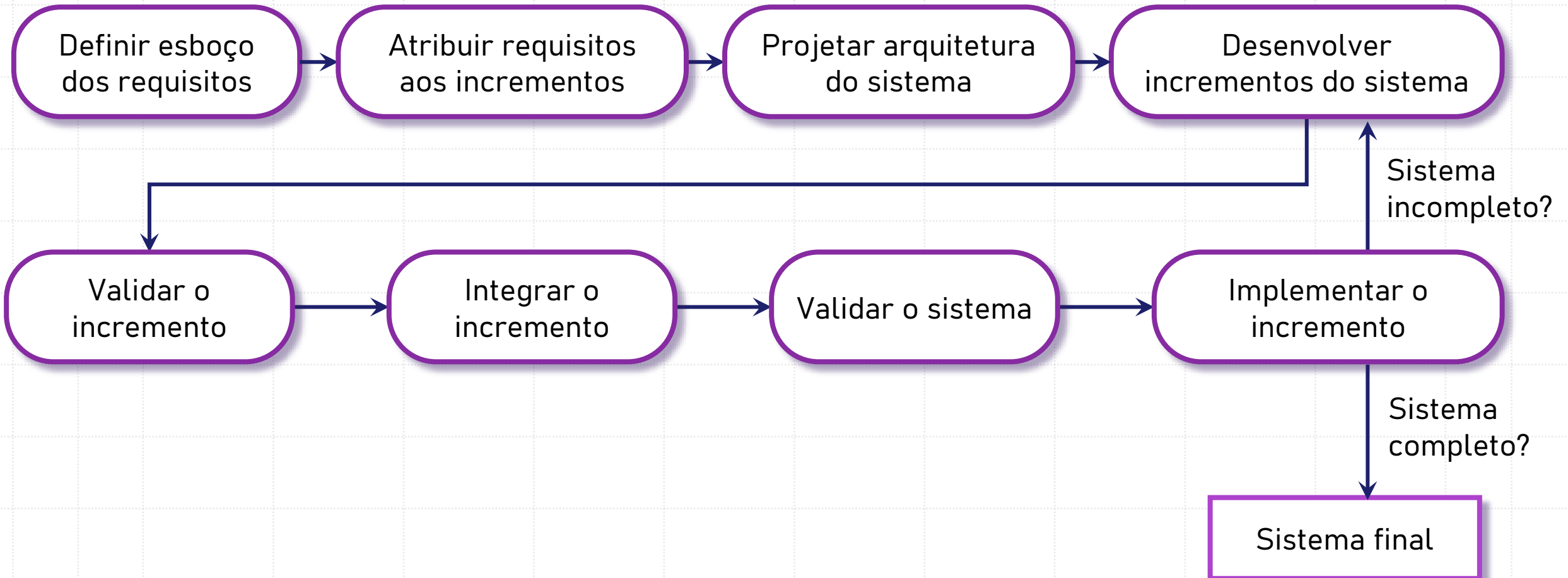
Lidando com Mudanças



Entrega incremental

- É uma abordagem para o desenvolvimento de software em que alguns dos incrementos desenvolvidos são fornecidos para o cliente;
- Nesse caso, os clientes definem quais serviços são mais e menos importantes;
- Os serviços mais importantes são alocados e implementados com maior prioridade;

Entrega incremental.





Entrega incremental

- A medida que um incremento é concluído, ele é instalado no ambiente de trabalho normal do cliente;
- Isso ajudará a esclarecer os requisitos dos incrementos posteriores;
- À medida que novos incrementos são concluídos, eles são integrados aos existentes;



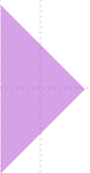
Entrega incremental

- **Vantagens** da entrega incremental:
 - Os incrementos iniciais podem ser utilizados como protótipos e adquirir experiência, que servirá para informar seus requisitos para incrementos posteriores;
 - Não é preciso esperar o sistema todo ser concluído;
 - O processo mantém os benefícios do desenvolvimento incremental, tornando fácil incorporar mudanças no sistema;
 - Os serviços de prioridade mais elevada recebem a maioria dos testes, uma vez que são fornecidos primeiro;



Entrega incremental

- **Problemas** da entrega incremental:
 - Quando o novo sistema substituirá um preexistente;
 - A maioria dos sistemas requer um conjunto de recursos básicos que são utilizados por diferentes partes do sistema;
 - No processo iterativo a especificação é desenvolvida em conjunto com o software, entretanto, na maioria das organizações a especificação completa faz parte do contrato de desenvolvimento;



Melhoria de Processo



Melhoria de Processo

- A melhoria dos processos significa:
 - Compreender os processos existentes e modifica-los para aumentar a qualidade do produto;
 - Reduzir os custos e o tempo de desenvolvimento;
- Podem ser empregadas duas abordagens para melhoria e mudança de processos:
 - Abordagem de maturidade do processo;
 - Abordagem ágil;



Melhoria de Processo

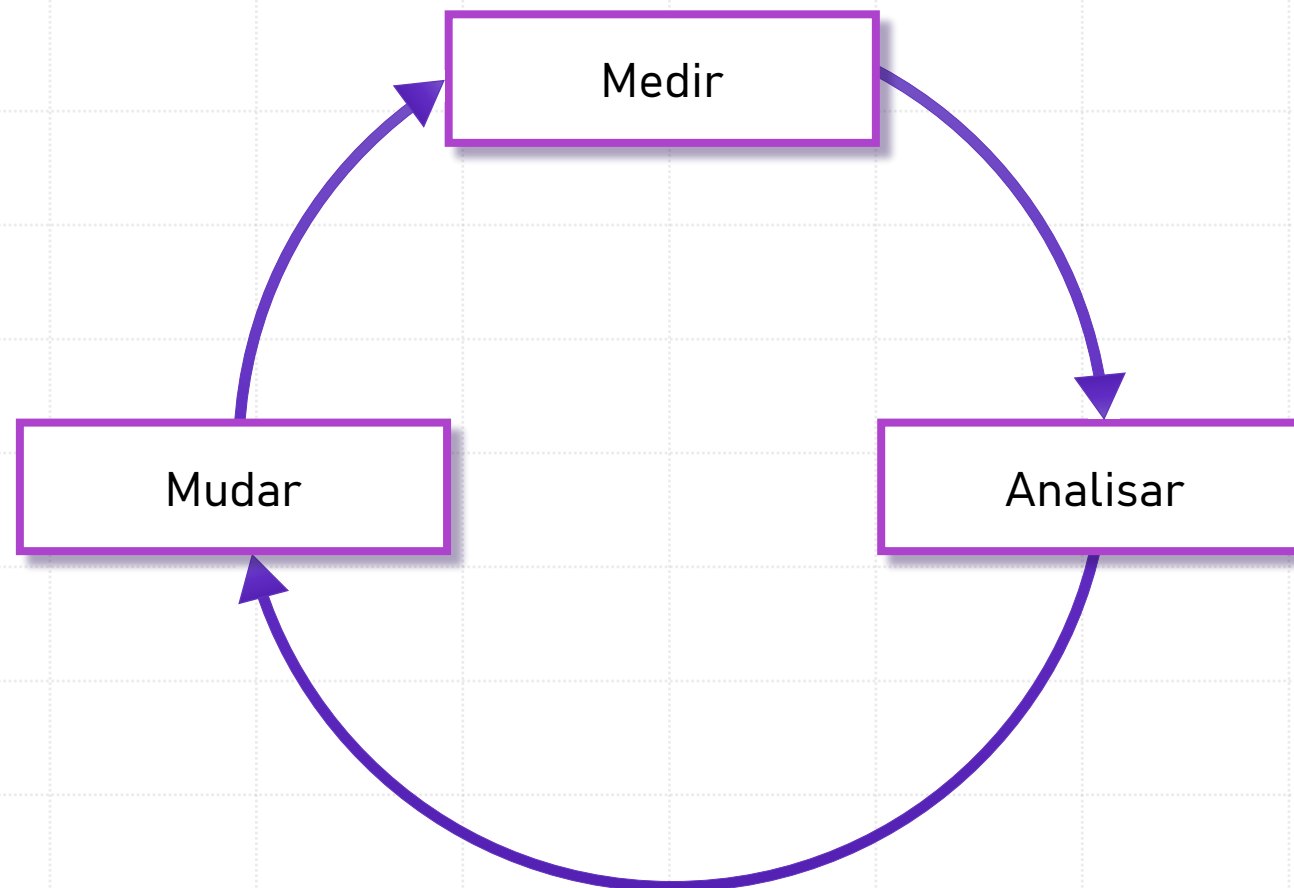
- A abordagem de **maturidade do processo** concentra na melhoria dos processos e do gerenciamento do projeto e na introdução de práticas recomendadas de engenharia de software;
- O nível de maturidade reflete o grau em que as práticas foram adotadas no processo;
- Os objetivos dessa abordagem é maior qualidade do produto e a previsibilidade do processo;



Melhoria de Processo

- A abordagem **ágil** concentra-se no desenvolvimento iterativo e na redução dos custos gerais do processo de software;
- Os objetivos são a entrega rápida da funcionalidade e a rapidez de resposta para os requisitos mutáveis;

O ciclo de melhoria do processo.

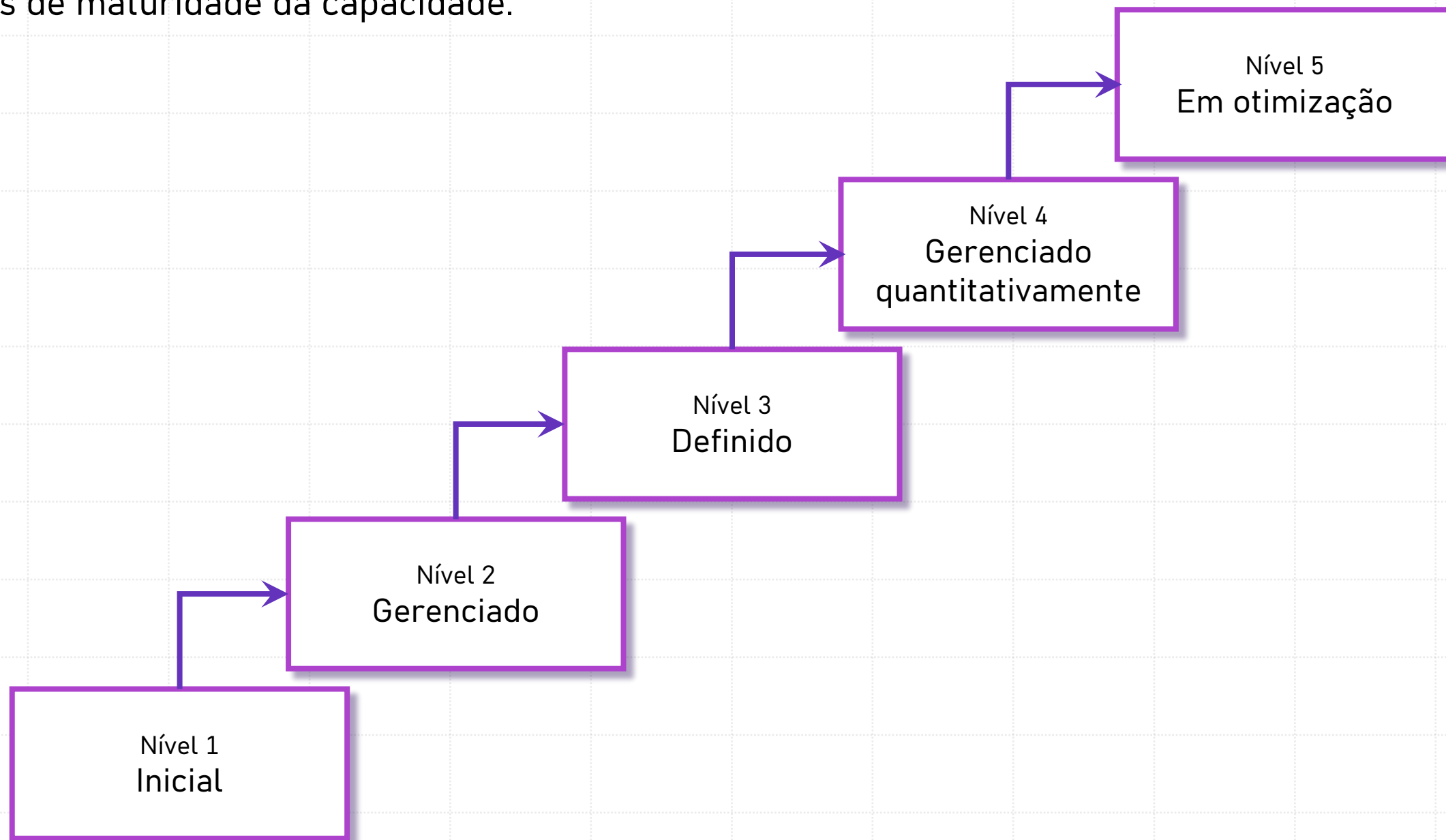


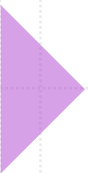


Melhoria de Processo

- A melhoria dos processos é uma atividade de longo prazo e contínua;
- A maturidade dos processos de uma empresa de software reflete seu gerenciamento de processos, sua medição e o uso que ela faz de práticas recomendadas em engenharia de software;
- Existem cinco níveis de maturidade do processo;

Níveis de maturidade da capacidade.





Resumo



Resumo

- Processos de software são as atividades envolvidas na produção de um sistema de software. Os modelos de processos de software são representações abstratas desses processos;
- Modelos de processos gerais descrevem a organização dos processos de software. Exemplos gerais incluem o modelo em cascata, o desenvolvimento incremental e a configuração e integração de componentes reusáveis;



Resumo

- A engenharia de requisitos é o processo de desenvolver uma especificação de software. Especificações se destinam a comunicar as necessidades do cliente para os desenvolvedores do sistema;
- Os processos de projeto (*design*) e implementação se preocupam em transformar uma especificação de requisitos em um sistema de software executável;
- A validação de software é o processo de conferir se o sistema está em conformidade com a sua especificação e se isso satisfaz as necessidades reais dos usuários do sistema;



Resumo

- A evolução do software ocorre quando você modifica os sistemas de software existentes para atender a novos requisitos. As mudanças são contínuas e o software deve evoluir para continuar sendo útil;
- Os processos devem incluir atividades para lidar com a mudança. Isso pode envolver uma fase de prototipação, que ajuda a evitar as más decisões de requisitos e projeto. Os processos podem ser estruturados para desenvolvimento e entrega iterativos, de modo que as mudanças possam ser feitas sem perturbar o sistema como um todo;



Resumo

- A melhoria dos processo é a atividade de aperfeiçoar os processos de software existentes, elevando sua qualidade, diminuindo os custos ou reduzindo o tempo de desenvolvimento. Trata-se de um processo cíclico que envolve medição, análise e modificação de processos.



Referências Bibliográficas

- SOMMERVILLE, Ian. **Engenharia de software**. 10. ed. São Paulo, SP: Pearson, 2018.