Traduzido do Inglês para o Português - www.onlinedoctranslator.com

INSTITUTO SUPERIOR DE TECNOLOGIAS AVANÇADAS DE LISBOA

Engenharia de software

João Caldeira

Professor convidado

E-mail. joaocarlos.caldeira@my.istec.pt

Mob. +351 917769544

2 de novembro de 2022

INSTITUTO SUPERIOR DE TECNOLOGIAS AVANÇADAS DE LISBOA

Software Engenharia

Processo

2 de novembro de 2022

Engenharia de software

Conjunto de atividades

- **Requisitos**
  - Necessidades de negócios do usuário final/cliente
- **Especificação**
  - A funcionalidade do software e as restrições ao seu funcionamento devem ser definidas
- **Projeto e Implementação**
  - A funcionalidade do software e as restrições ao seu funcionamento devem ser definidas
- **Validação de verificação**
  - O software deve ser validado para garantir que faça o que o usuário final / cliente deseja
- **Evolução**
  - O software deve evoluir para atender às necessidades do cliente em constante mudança

Direitos autorais @ 2022

- **O modelo cascata**
  - As atividades fundamentais do processo de especificação, desenvolvimento, validação e evolução são representadas como fases de processo separadas, como especificação de requisitos, projeto de software, implementação, teste, etc.
- **Desenvolvimento incremental**
  - Essa abordagem intercala as atividades de especificação, desenvolvimento e validação
  - O sistema é desenvolvido como uma série de versões (incrementos), com cada versão adicionando funcionalidade à versão anterior
- **Integração e Configuração (engenharia de software orientada a reutilização)**
  - Esta abordagem baseia-se na existência de um número significativo de componentes reutilizáveis
  - Concentra-se na integração de componentes em um sistema, em vez de desenvolver do zero

Direitos autorais @ 2022

Processo de Engenharia de Software

Fases do modelo em cascata

1. Análise e definição de requisitos

- Os serviços, restrições e objetivos do sistema são estabelecidos por consulta com os usuários do sistema

2. Projeto de sistema e software

- O processo de projeto de sistemas aloca os requisitos para sistemas de hardware ou software, estabelecendo uma arquitetura geral do sistema. O projeto de software envolve identificar e descrever as abstrações fundamentais do sistema de software e seus relacionamentos.

3. Implementação e teste de unidade

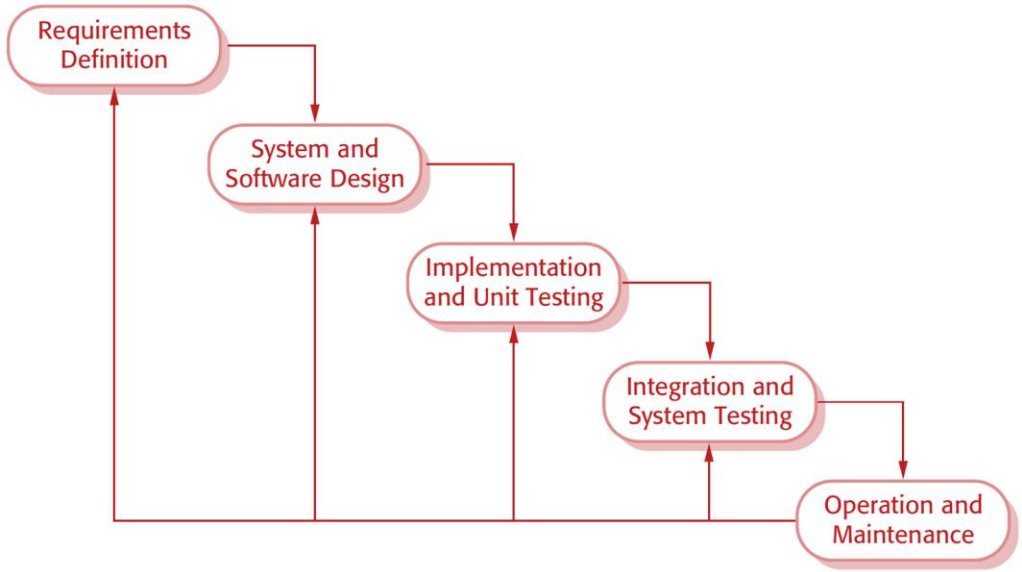
- O projeto de software é realizado como um conjunto de programas ou unidades de programa. O teste de unidade envolve a verificação de que cada unidade atende às suas especificações

4. Integração e teste do sistema

- As unidades ou programas individuais do programa são integrados e testados como um sistema completo para garantir que os requisitos de software sejam atendidos

5. Operação e manutenção

- Normalmente (embora não necessariamente), esta é a fase do ciclo de vida mais longa. O sistema é instalado e colocado em prática. A manutenção envolve a correção de erros que não foram descobertos em estágios anteriores do ciclo de vida



Direitos autorais @ 2022

Processo de Engenharia de Software

Problemas do modelo em cascata

1. A divisão inflexível do projeto em estágios distintos torna difícil responder às mudanças nos requisitos do cliente.
   - Uma fase deve ser concluída antes de passar para a próxima

2. Este modelo só é apropriado quando os requisitos são bem compreendidos e as mudanças serão bastante limitadas
   - Poucos sistemas de negócios têm requisitos estáveis

3. O modelo em cascata é usado principalmente para grandes sistemas/projetos onde um sistema é desenvolvido em vários locais

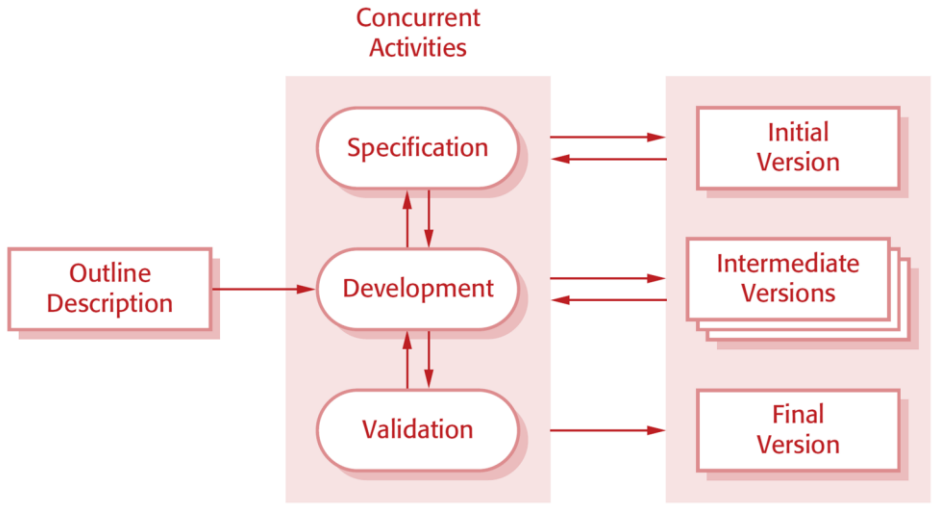
Direitos autorais @ 2022

Processo de Engenharia de Software

Modelo de Desenvolvimento Incremental

- O desenvolvimento incremental é baseado na ideia de desenvolver uma implementação inicial, expondo-a ao comentário do usuário e evoluindo-a através de várias versões até que um sistema adequado seja desenvolvido

- As atividades de especificação, desenvolvimento e validação são intercaladas em vez de separadas, com feedback rápido entre as atividades



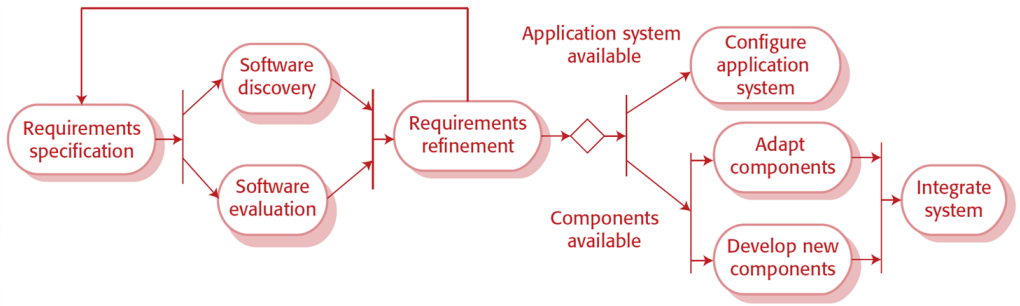
- **Benefícios sobre a Cachoeira**
  - O custo de acomodação**a mudança dos requisitos do cliente é reduzida**. A quantidade de análise e documentação que precisa ser refeita é**muito menos do que é necessário com o modelo em cascata**.
  - **É mais fácil obter feedback do cliente**sobre o trabalho de desenvolvimento que foi feito. Os clientes podem comentar as demonstrações do software e ver o quanto foi implementado. Os clientes acham difícil avaliar o progresso dos documentos de design de software
  - É possível uma entrega e implantação mais rápidas de software útil para o cliente, mesmo que todas as funcionalidades não tenham sido incluídas.**Os clientes podem usar e obter valor do software mais cedo**do que é possível com um processo em cascata

Direitos autorais @ 2022

Processo de Engenharia de Software

Integração e configuração

- Em projetos de software, há alguma reutilização de software
- Os desenvolvedores procuram por eles, modificam-nos conforme necessário e os integram com o novo código que eles desenvolveram
- Desde 2000, os processos de desenvolvimento de software que focam na reutilização de software existente tornaram-se amplamente utilizados
- Abordagens orientadas ao reuso contam com uma base de componentes de software reutilizáveis e uma estrutura de integração para a composição desses componentes



- **Componentes de software frequentemente reutilizados**
  - **Aplicativo autônomo**sistemas configurados para uso em um ambiente específico. Esses sistemas são sistemas de uso geral que possuem muitos recursos, mas precisam ser adaptados para uso em uma aplicação específica.
  - **Coleções de objetos**que são desenvolvidos como um componente ou como um pacote a ser integrado a uma estrutura de componentes, como o**Estrutura Java Spring**
  - **serviços web**que são desenvolvidos de acordo com os padrões de serviço e que estão disponíveis para chamada remota pela Internet

Direitos autorais @ 2022

Engenharia de software

Conjunto de atividades

- **Requisitos**
  - Necessidades de negócios do usuário final/cliente
- **Especificação**
  - A funcionalidade do software e as restrições ao seu funcionamento devem ser definidas
- **Projeto e Implementação**
  - A funcionalidade do software e as restrições ao seu funcionamento devem ser definidas
- **Validação de verificação**
  - O software deve ser validado para garantir que faça o que o usuário final / cliente deseja
- **Evolução**
  - O software deve evoluir para atender às necessidades do cliente em constante mudança

Direitos autorais @ 2022

Processo de Engenharia de Software

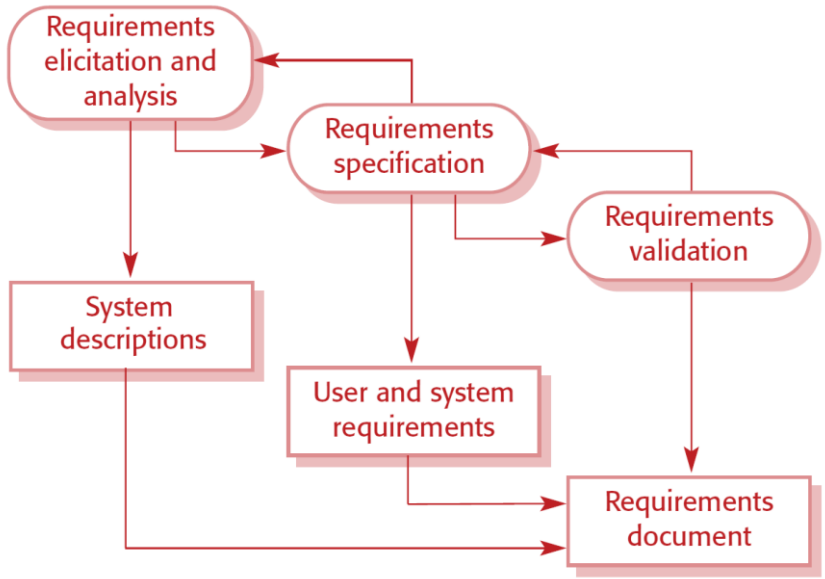
Especificação de software

Existem três atividades principais no processo de engenharia de requisitos:

1. **Levantamento e análise de requisitos.** Este é o processo de derivar os requisitos do sistema através da observação dos sistemas existentes, discussões com potenciais usuários e compradores, análise de tarefas e assim por diante. Isso pode envolver o desenvolvimento de um ou mais modelos e protótipos de sistemas.
   Estes ajudam você a entender o sistema a ser especificado

2. **Especificação de requisitos.** A especificação de requisitos é a atividade de traduzir as informações coletadas durante a análise de requisitos em um documento que define um conjunto de requisitos. Dois tipos de requisitos podem ser incluídos neste documento. Os requisitos do usuário são declarações abstratas dos requisitos do sistema para o cliente e o usuário final do sistema; os requisitos do sistema são uma descrição mais detalhada da funcionalidade a ser fornecida

3. **Validação de requisitos.** Esta atividade verifica os requisitos de realismo, consistência e integridade. Durante este processo, erros no
   documento de requisitos são inevitavelmente descobertos. Ele deve então ser modificado para corrigir esses problemas



Direitos autorais @ 2022

Processo de Engenharia de Software

Projeto e Implementação de Software

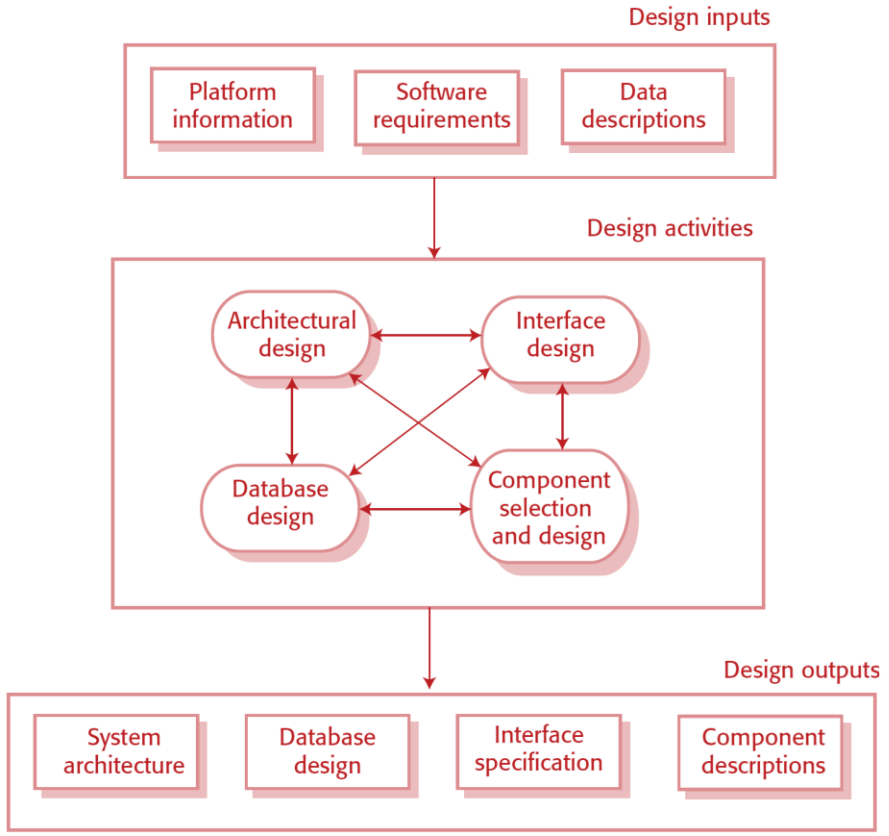
As atividades no processo de projeto variam, dependendo do tipo de sistema que está sendo desenvolvido. Normalmente, quatro atividades que podem fazer parte do processo de design de sistemas de informação:

1. **Projeto arquitetônico.** Onde você identifica a estrutura geral do sistema, os principais componentes (às vezes chamados de subsistemas ou módulos), seus relacionamentos e como eles são distribuídos

2. **Projeto de banco de dados.** Onde você projeta as estruturas de dados do sistema e como elas devem ser representadas em um banco de dados. Novamente, o trabalho aqui depende se um banco de dados existente deve ser reutilizado ou se um novo banco de dados deve ser criado

3. **Projeto de interface.** Onde você define as interfaces entre os componentes do sistema. Esta especificação de interface deve ser inequívoca. Com uma interface precisa, um componente pode ser usado por outros componentes sem que eles precisem saber como ele é implementado. Uma vez que as especificações da interface são acordadas, os componentes podem ser projetados e desenvolvidos separadamente.

4. **Seleção e projeto de componentes.** Onde você procura componentes reutilizáveis e, se nenhum componente adequado estiver disponível, projeta novos componentes de software. O projeto neste estágio pode ser uma descrição simples do componente com os detalhes de implementação deixados para o programador. Alternativamente, pode ser uma lista de alterações a serem feitas em um componente reutilizável ou um modelo de projeto detalhado expresso na UML. O modelo de design pode então ser usado para gerar automaticamente uma implementação.



Direitos autorais @ 2022

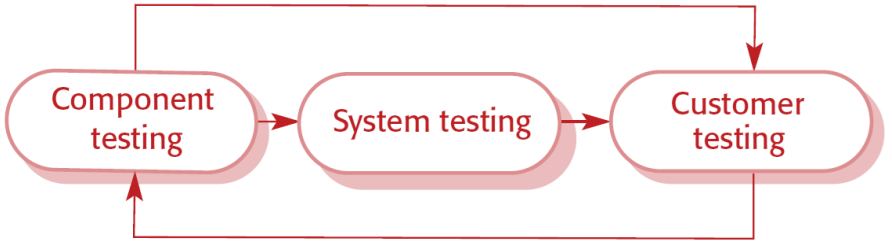
Processo de Engenharia de Software

Teste de software

As etapas do processo de teste são:

1. **Teste de componentes.** Os componentes que compõem o sistema são testados pelas pessoas que desenvolvem o sistema. Cada componente é testado independentemente, sem outros componentes do sistema. Os componentes podem ser entidades simples, como funções ou classes de objetos, ou podem ser agrupamentos coerentes dessas entidades.

2. **Teste do sistema.** Os componentes do sistema são integrados para criar um sistema completo. Este processo está preocupado em encontrar erros que resultam de interações inesperadas entre componentes e problemas de interface de componentes. Também se preocupa em mostrar que o sistema atende aos seus requisitos funcionais e não funcionais e testar as propriedades emergentes do sistema.

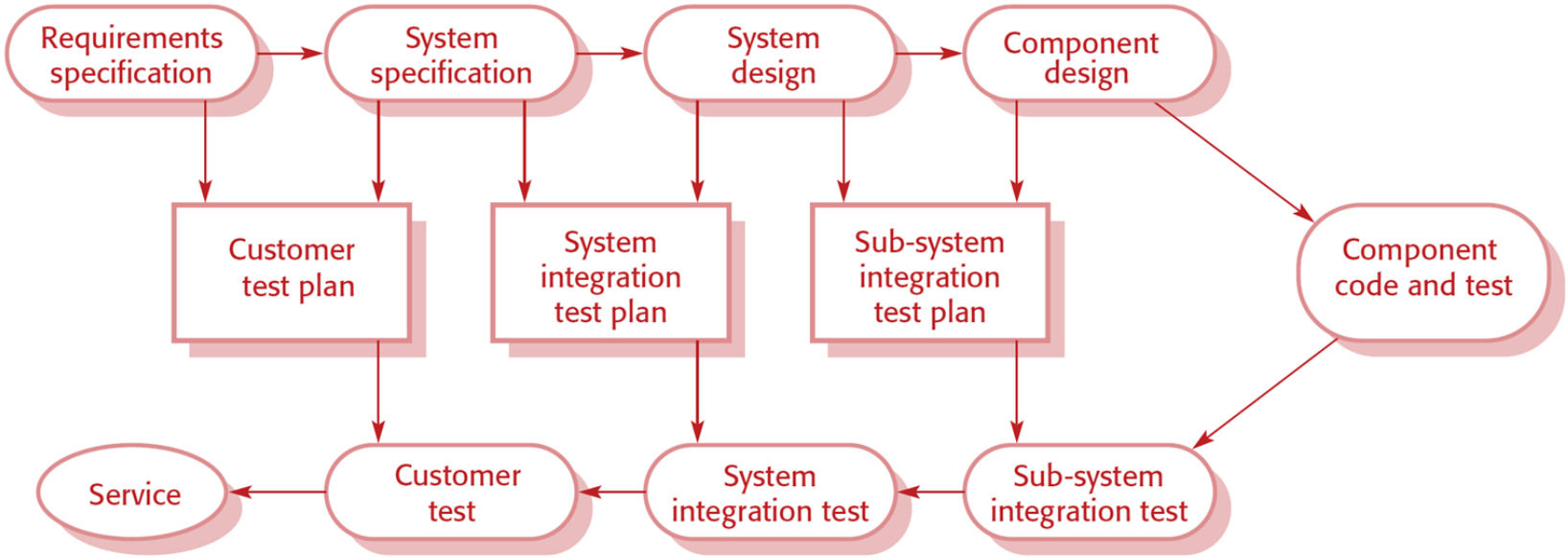


3. **Teste do cliente.** Este é o estágio final no processo de teste antes que o sistema seja aceito para uso operacional. O sistema é testado pelo cliente do sistema (ou cliente potencial) e não com dados de teste simulados. Para software customizado, o teste do cliente pode revelar erros e omissões na definição dos requisitos do sistema, porque os dados reais exercitam o sistema de maneiras diferentes dos dados de teste

Direitos autorais @ 2022

Processo de Engenharia de Software

Fases de teste de software - Processo de software orientado a planos

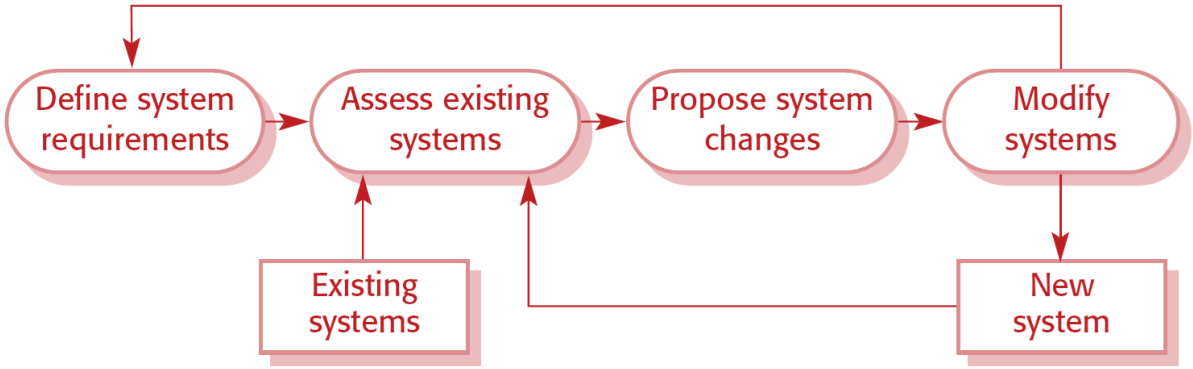


Direitos autorais @ 2022

Processo de Engenharia de Software

Evolução do Software

- A flexibilidade do software é uma das principais razões pelas quais cada vez mais software está sendo incorporado em sistemas grandes e complexos.
- Uma vez tomada a decisão de fabricar o hardware, é muito caro fazer alterações no design do hardware
- No entanto, as alterações podem ser feitas no software a qualquer momento durante ou após o desenvolvimento do sistema
- Esta distinção entre desenvolvimento e manutenção é cada vez mais irrelevante
- Em vez de dois processos separados, é mais realista pensar na engenharia de software como um processo evolutivo



Direitos autorais @ 2022

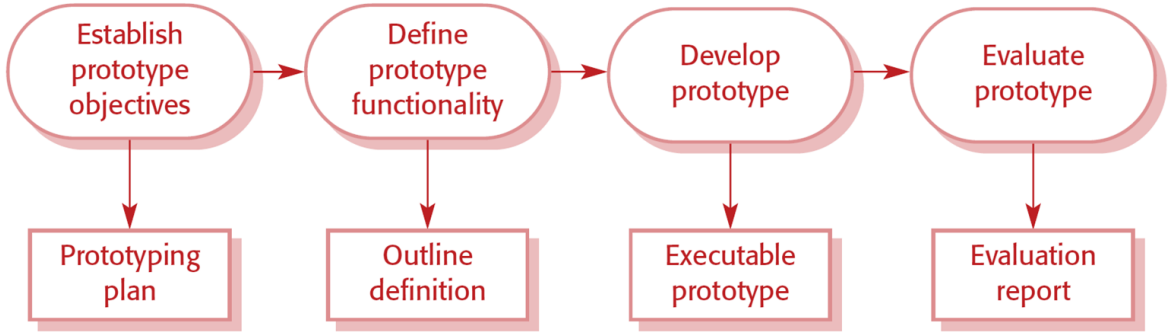
Processo de Engenharia de Software

Lidando com a Mudança - Desenvolvimento de Protótipos

Um protótipo de software pode ser usado em um processo de desenvolvimento de software para ajudar a antecipar mudanças que podem ser necessárias:

1. No processo de engenharia de requisitos, um protótipo pode ajudar na elicitação e validação dos requisitos do sistema

2. No processo de projeto do sistema, um protótipo pode ser usado para explorar soluções de software e no desenvolvimento de uma interface de usuário para o sistema



Direitos autorais @ 2022

Processo de Engenharia de Software

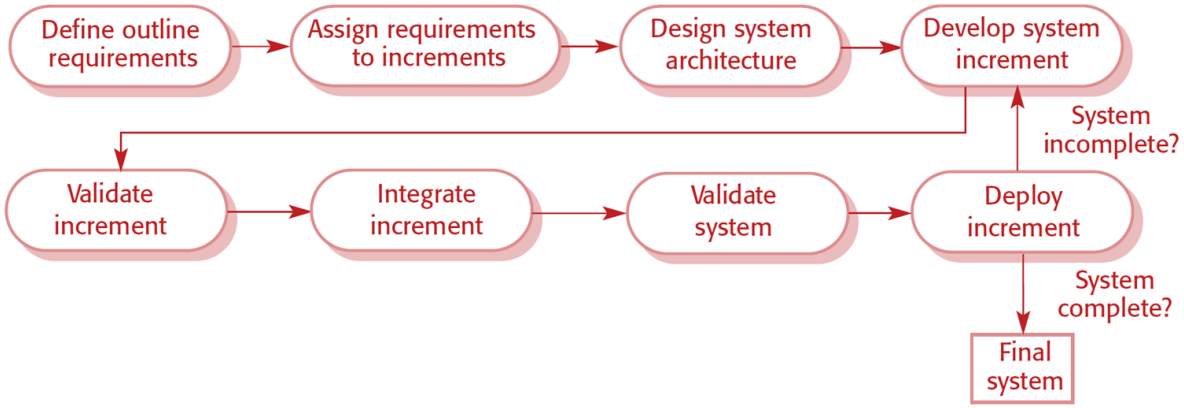
Entrega incremental

A entrega incremental tem várias vantagens:

1. Os clientes podem usar os primeiros incrementos como protótipos e ganhar experiência que informa seus requisitos para incrementos de sistema posteriores

2. Os clientes não precisam esperar até que todo o sistema seja entregue antes que possam ganhar valor com ele. O primeiro incremento satisfaz seus requisitos mais críticos, para que eles possam usar o software imediatamente

3. O processo mantém os benefícios do desenvolvimento incremental, pois deve ser relativamente fácil incorporar mudanças no sistema



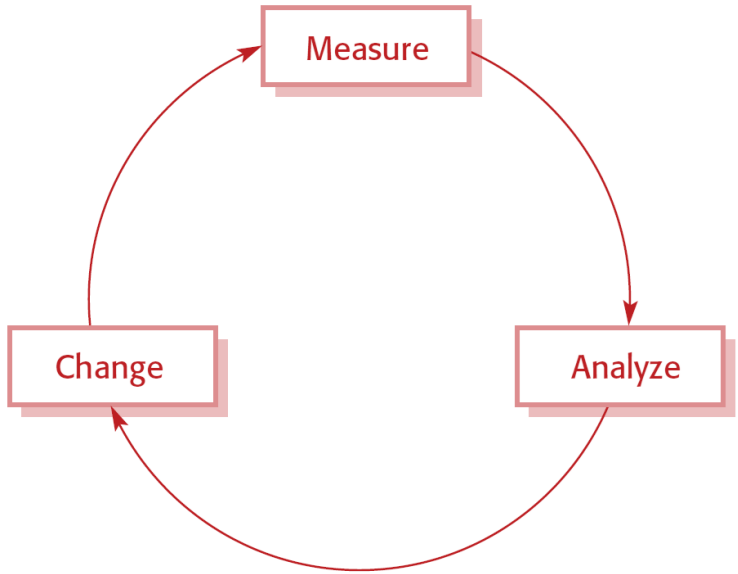
Direitos autorais @ 2022

Processo de Engenharia de Software

Ciclo de Melhoria de Processo

O processo geral de melhoria de processos subjacente à abordagem de maturidade de processos é um processo cíclico, pois as etapas desse processo são:

1. **Medição do processo.** Você mede um ou mais atributos do processo ou produto de software. Essas medições formam uma linha de base que ajuda você a decidir se as melhorias do processo foram eficazes
2. **Análise de processos.** O processo atual é avaliado e as fraquezas e gargalos do processo são identificados. Modelos de processo (às vezes chamados de mapas de processo) que descrevem o processo podem ser desenvolvidos durante este estágio
3. **Mudança de processo.** Mudanças de processo são propostas para resolver algumas das deficiências de processo identificadas. Estes são introduzidos e o ciclo é retomado para coletar dados sobre a eficácia das mudanças



Direitos autorais @ 2022

Processo de Engenharia de Software

Níveis de maturidade de capacidade

Os níveis no modelo de maturidade do processo são:

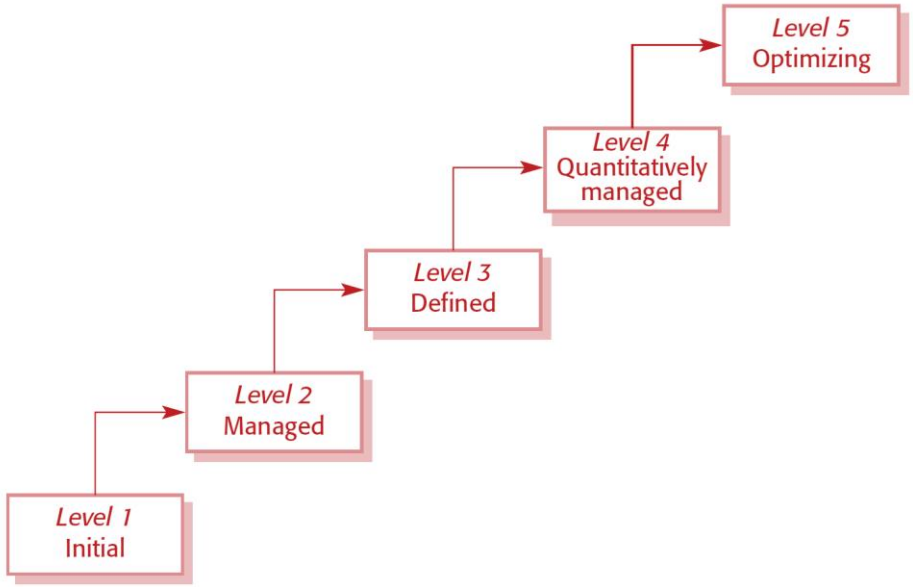
1. **Inicial.**Os objetivos associados à área de processo são atendidos e, para todos os processos, o escopo do trabalho a ser executado é explicitamente definido e comunicado aos membros da equipe.

2. **Gerenciado.**Nesse nível, os objetivos associados à área de processo são atendidos e as políticas organizacionais estão em vigor que definem quando cada processo deve ser usado. Deve haver planos de projeto documentados que definam os objetivos do projeto. Os procedimentos de gerenciamento de recursos e monitoramento de processos devem estar em vigor

3. **Definido.**Este nível concentra-se na padronização organizacional e implantação de processos. Cada projeto tem um processo gerenciado que é adaptado aos requisitos do projeto a partir de um conjunto definido de processos organizacionais. Ativos de processo e medições de processo devem ser coletados e usados para futuras melhorias de processo

4. **Gerenciado quantitativamente.**Nesse nível, há uma responsabilidade organizacional de usar métodos estatísticos e outros métodos quantitativos para controlar os subprocessos. Ou seja, as medições coletadas do processo e do produto devem ser usadas no gerenciamento de processos.

5. **Otimização.**Nesse nível mais alto, a organização deve usar as medições do processo e do produto para impulsionar a melhoria do processo. As tendências devem ser analisadas e os processos adaptados às necessidades de negócios em constante mudança



Direitos autorais @ 2022

- **A filosofia por trás dos métodos ágeis é refletida no manifesto ágil (http://agilemanifesto.org)**
- **Este manifesto afirma:**

  - Indivíduos e interações sobre processos e ferramentas
  - Software que trabalha sobre uma documentação completa
  - Colaboração do cliente sobre a negociação do contrato
  - Responder à mudança ao invés de seguir um plano

Direitos autorais @ 2022

Processo de Engenharia de Software

Métodos ágeis

- Todos os métodos ágeis sugerem que o software**deve ser desenvolvido e entregue de forma incremental**
- Esses métodos são baseados em diferentes processos ágeis, mas compartilham um conjunto de princípios, baseados no manifesto ágil, e por isso têm muito em comum.

- Os métodos ágeis foram particularmente bem-sucedidos para dois tipos de desenvolvimento de sistemas:

1. Desenvolvimento de produto onde uma empresa de software está desenvolvendo um produto de pequeno ou médio porte para venda. Praticamente todos os produtos de software e aplicativos agora são desenvolvidos usando uma abordagem ágil

2. Desenvolvimento de sistema personalizado dentro de uma organização, onde há um claro compromisso do cliente em se envolver no processo de desenvolvimento e onde há poucos stakeholders externos e regulamentações que afetam o software

Direitos autorais @ 2022

Processo de Engenharia de Software

Técnicas de desenvolvimento ágil

- **XP (Programação Extrema)**
- **Histórias de usuário**
- **Reestruturação**
- **Desenvolvimento de teste**
- **Programação em pares**

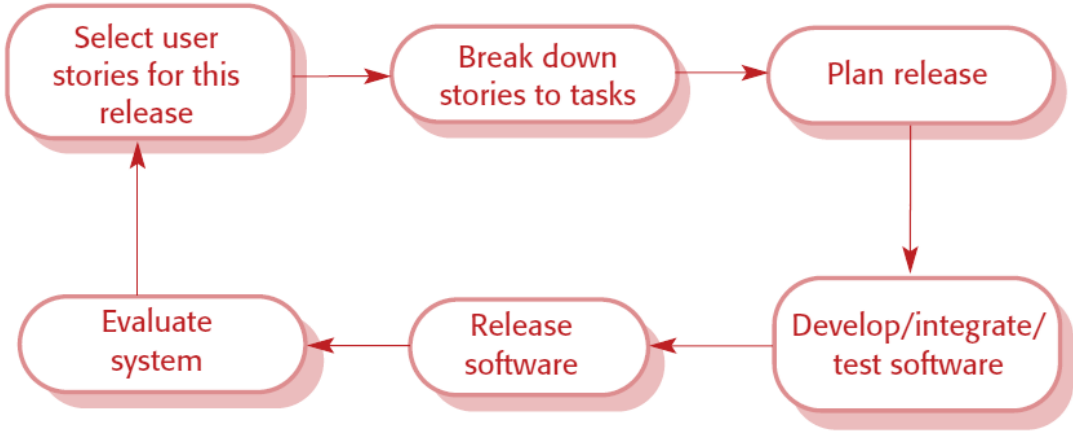
Direitos autorais @ 2022

Processo de Engenharia de Software

Técnicas de desenvolvimento ágil

Histórias de usuário

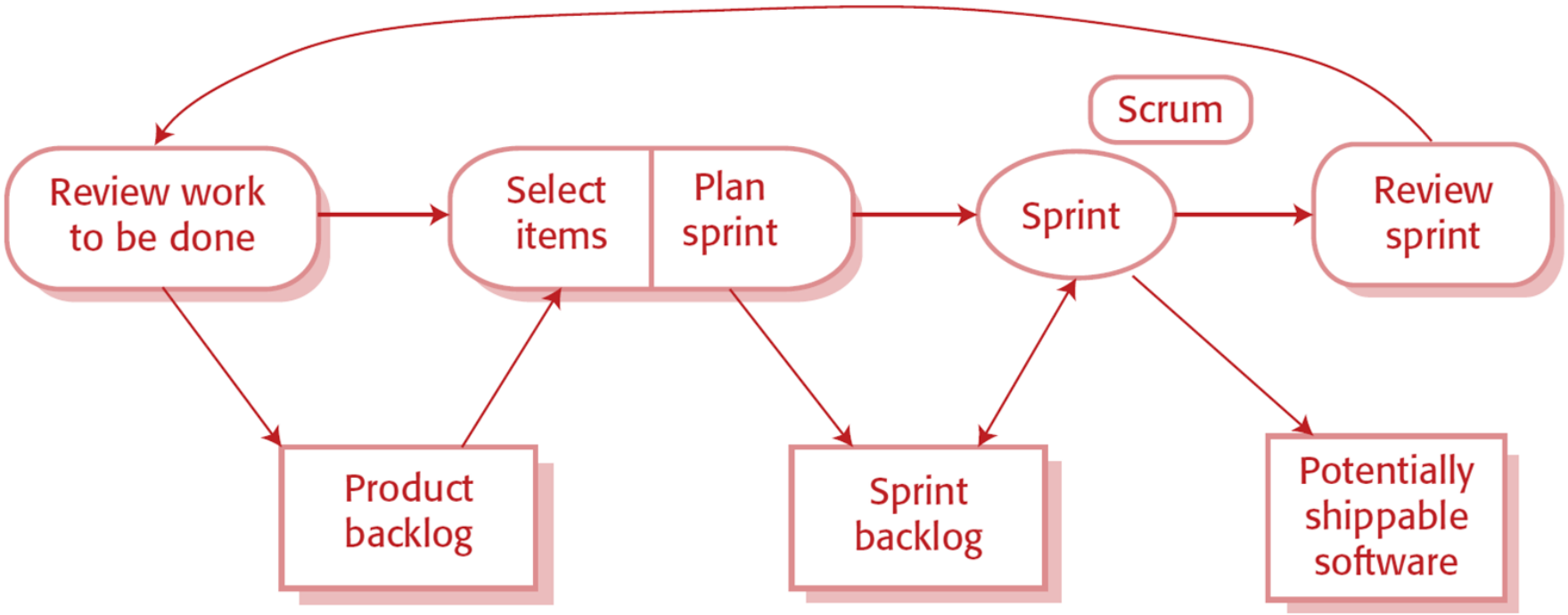
- Os requisitos de software sempre mudam
- Para lidar com essas mudanças, os métodos ágeis não possuem uma atividade de engenharia de requisitos separada. Em vez disso, eles integram a elicitação de requisitos com o desenvolvimento
- Para facilitar isso, foi desenvolvida a ideia de “histórias de usuário” onde uma história de usuário é um cenário de uso que pode ser vivenciado por um usuário do sistema



Direitos autorais @ 2022

Processo de Engenharia de Software

Gerenciamento Ágil de Projetos



Direitos autorais @ 2022

Processo de Engenharia de Software

Gerenciamento Ágil de Projetos

Termo do Scrum	Definição
Equipe de desenvolvimento	Um grupo auto-organizado de desenvolvedores de software, que não deve ter mais de sete pessoas. Eles são responsáveis por desenvolver o software e outros documentos essenciais do projeto
Incremento de produto potencialmente entregável	O incremento de software que é entregue a partir de um sprint. A ideia é que isso seja "potencialmente despachável", o que significa que está em um estado finalizado e nenhum trabalho adicional, como testes, é necessário para incorporá-lo ao produto final. Na prática, isso nem sempre é possível
Lista de pendências do produto	Esta é uma lista de itens "a fazer" que a equipe Scrum deve abordar. Eles podem ser definições de recursos para o software, requisitos de software, histórias de usuários ou descrições de tarefas complementares necessárias, como definição de arquitetura ou documentação do usuário.
Proprietário do produto	Um indivíduo (ou possivelmente um pequeno grupo) cujo trabalho é identificar recursos ou requisitos do produto, priorizá-los para desenvolvimento e revisar continuamente o backlog do produto para garantir que o projeto continue atendendo às necessidades críticas de negócios. O Product Owner pode ser um cliente, mas também pode ser um gerente de produto em uma empresa de software ou outro representante das partes interessadas
Scrum	Uma reunião diária da equipe Scrum que revisa o progresso e prioriza o trabalho a ser feito naquele dia. Idealmente, esta deve ser uma curta reunião presencial que inclua toda a equipe
Scrum Master	O ScrumMaster é responsável por garantir que o processo Scrum seja seguido e orienta a equipe no uso eficaz do Scrum. Ele ou ela é responsável pela interface com o resto da empresa e por garantir que a equipe Scrum não seja desviada por interferência externa. Os desenvolvedores Scrum estão convencidos de que o ScrumMaster não deve ser pensado como um gerente de projeto. Outros, no entanto, podem nem sempre achar fácil ver a diferença.
Corrida	Uma iteração de desenvolvimento. Sprints geralmente duram de 2 a 4 semanas
Velocidade	Uma estimativa de quanto esforço de backlog do produto uma equipe pode cobrir em um único sprint. Compreender a velocidade de uma equipe os ajuda a estimar o que pode ser coberto em um sprint e fornece uma base para medir a melhoria do desempenho

Direitos autorais @ 2022

Processo de Engenharia de Software

Problemas de métodos ágeis

- **Para sistemas grandes e de longa vida útil desenvolvidos por uma empresa de software para um cliente externo, o uso de uma abordagem ágil apresenta vários problemas:**

  - A informalidade do desenvolvimento ágil**é incompatível com a abordagem legal do contrato**definição que é comumente usada em grandes empresas
  - Métodos ágeis**são mais apropriados para o desenvolvimento de novos softwares do que para a manutenção de softwares**. Ainda a maioria dos**custos de software em grandes empresas vêm da manutenção**seus sistemas de software existentes
  - Métodos ágeis**são projetados para pequenas equipes co-localizadas**, mas muito desenvolvimento de software**agora envolve equipes distribuídas em todo o mundo**

Direitos autorais @ 2022