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

- **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

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

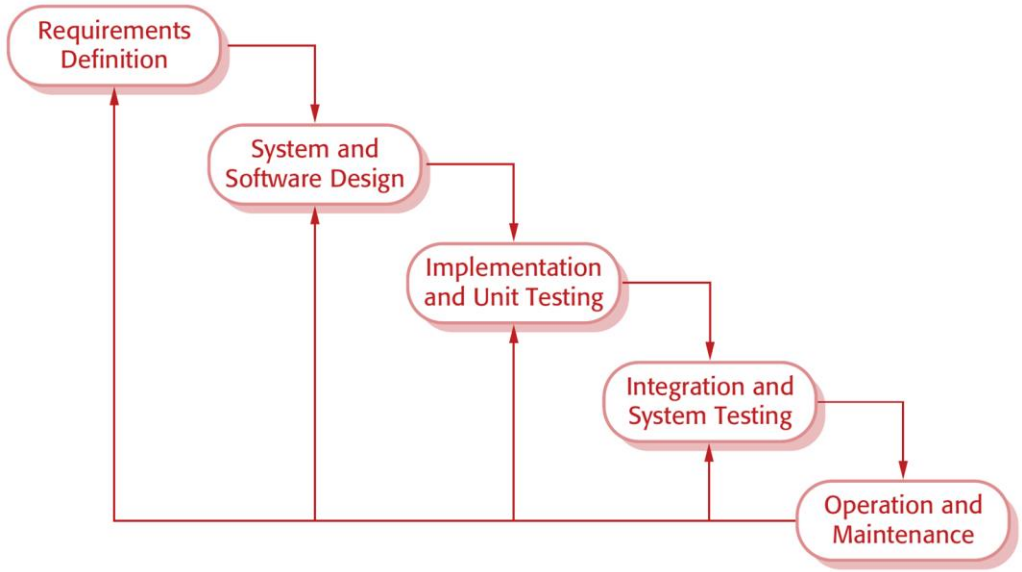
- O projeto de software é realizado como um conjunto de programas ou unidades de programa. O teste de unidade envolve a verificação de que cada unidade atende às suas especificações

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



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

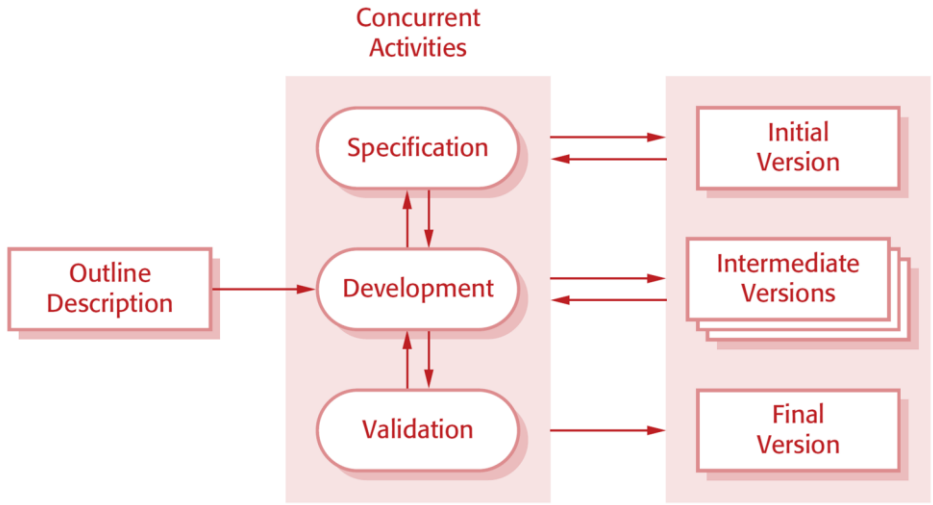
3. O modelo em cascata é usado principalmente para grandes sistemas/projetos onde um sistema é desenvolvido em vários locais

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

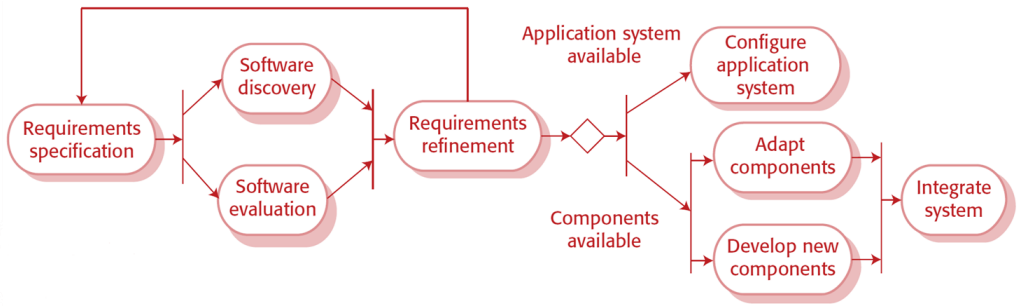


- **Benefícios sobre a Cachoeira**
  - O custo de acomodação**a mudança dos requisitos do cliente é reduzida**. A quantidade de análise e documentação que precisa ser refeita é**muito menos do que é necessário com o modelo em cascata**.
  - **É mais fácil obter feedback do cliente**sobre o trabalho de 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

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

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

- **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

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.

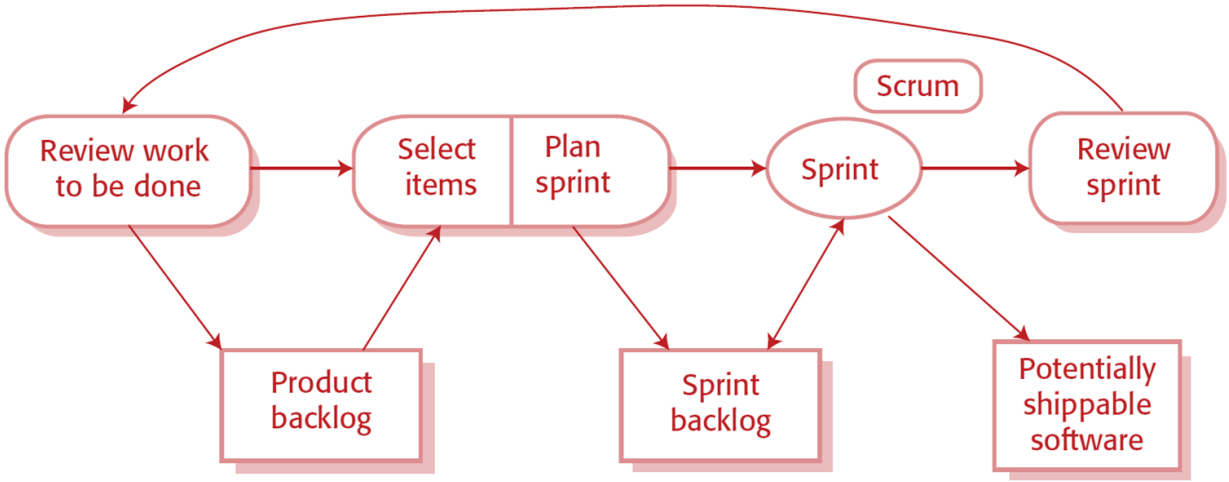
- Os métodos ágeis foram particularmente bem-sucedidos para dois tipos de desenvolvimento de sistemas:

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

Processo de Engenharia de Software

Gerenciamento Ágil de Projetos



Scrum term	Definition
Development team	A self-organizing group of software developers, which should be no more than seven people. They are responsible for developing the software and other essential project documents.
Potentially shippable product increment	The software increment that is delivered from a sprint. The idea is that this should be "potentially shippable," which means that it is in a finished state and no further work, such as testing, is needed to incorporate it into the final product. In practice, this is not always achievable.
Product backlog	This is a list of "to do" items that the Scrum team must tackle. They may be feature definitions for the software, software requirements, user stories, or descriptions of supplementary tasks that are needed, such as architecture definition or user documentation.
Product owner	An individual (or possibly a small group) whose job is to identify product features or requirements, prioritize these for development, and continuously review the product backlog to ensure that the project continues to meet critical business needs. The Product Owner can be a customer but might also be a product manager in a software company or other stakeholder representative.
Scrum	A daily meeting of the Scrum team that reviews progress and prioritizes work to be done that day. Ideally, this should be a short face-to-face meeting that includes the whole team.
ScrumMaster	The ScrumMaster is responsible for ensuring that the Scrum process is followed and guides the team in the effective use of Scrum. He or she is responsible for interfacing with the rest of the company and for ensuring that the Scrum team is not diverted by outside interference. The Scrum developers are adamant that the ScrumMaster should not be thought of as a project manager. Others, however, may not always find it easy to see the difference.
Sprint	A development iteration. Sprints are usually 2 to 4 weeks long.
Velocity	An estimate of how much product backlog effort a team can cover in a single sprint. Understanding a team's velocity helps them estimate what can be covered in a sprint and provides a basis for measuring improving performance.

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**

ISTEC 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