

ENGENHARIA DE SOFTWARE 2

Profa Cristiane Palomar Mercado

Objetivo da disciplina

-
- Compreender e aplicar padrões ao processo de software.
 - Mapear modelos de representação.
 - Empregar os diversos tipos de Arquitetura de Software.
 - Compreender e aplicar padrões de documentação, integração de sistemas e manutenção.

Conteúdo

-
- Modelos de Processo de Desenvolvimento de Software (Espiral e Prototipagem).
 - Modelagem de Sistemas utilizando (UML).
 - Modelagem de Processos do Negócio (BPMN).
 - Técnicas de documentação e Definition of Done (DoD).
 - Padrões de Projeto e Projeto de Arquitetura de Software.
 - Conceitos de Testes de Software.
 - Implantação de Software.
 - Evolução funcional e Manutenção corretiva de Software

Revisão de Conceitos

-
- **Processo de Software, Engenharia de Software.**
 - **Revisão RUP e Scrum.**
 - **Revisão de Requisitos:**
 - **Requisitos de Usuário e Requisitos de Sistema.**
 - **Requisitos Funcionais e Não Funcionais.**
 - **Elicitação de Requisitos.**
 - **Especificação de Requisitos**

AVALIAÇÃO

**Média = (Avaliação Teórica * 20% + Projeto * 50% +
Média Atividades extras * 30%)**

***Trabalhos Extras – Desenvolvidos durante as aulas e fora dos horários das aulas**

*** Projeto – projeto integrador**

Bibliografia Básica

PRESSMAN, Roger; MAXIM, Bruce. Engenharia de Software. 8 ed. São Paulo: McGraw Hill Brasil, 2016.

SOMMERVILLE, Ian. Engenharia De Software. 10 ed. São Paulo: Pearson Brasil, 2019.

WAZLAWICK, R. S. Engenharia de Software: conceitos e práticas. Rio de Janeiro: Elsevier, 2013.

Bibliografia Complementar

LENGHOLM JR. Hélio. Engenharia de Software na Prática. São Paulo: Novatec, 2010.

GUEDES, G. T. A. UML 2 - uma abordagem prática. São Paulo: Novatec, 2011.

HIRAMA, K. Engenharia de Software: qualidade e produtividade com tecnologia. Rio de Janeiro: Elsevier, 2011.

PAULA FILHO, W. P. Engenharia de software. Rio de Janeiro: LTC, 2010.

PETERS, James. Engenharia de software: Teoria e prática. Rio de Janeiro: Campus, 2001.

SBROCCO, J. H. T. C.; MACEDO, P. C. Metodologias Ágeis: Engenharia de Software sob medida. São Paulo: Érica, 2012.

TSUI, F.; KARAM, O. Fundamentos de Engenharia de Software. São Paulo: LTC, 2013.

***VAMOS
COMEÇAR!!***

REVISÃO

O que é um processo de software?

Segundo Sommerville (2011):

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

Segundo Pressman (2016):

Processo de software é definido como uma metodologia para as atividades, ações e tarefas necessárias para desenvolver um software de alta qualidade.

O que é Engenharia de Software

A Engenharia de Software é uma parte da Engenharia de Sistemas que se ocupa de todos os aspectos da produção de software (Sommerville, 2019).

Na concepção de Pressman (2016), a Engenharia de Software abrange um conjunto de três elementos: **métodos, ferramentas e procedimentos.**

Engenharia de software

Métodos

Fornecem as **informações técnicas para desenvolver software** (instruções de trabalho).

- Conjuntos de tarefas com técnicas particulares para cada fase do desenvolvimento de software.
- Os métodos envolvem uma ampla variedade de tarefas, que incluem: **comunicação, análise de requisitos, modelagem de projeto, construção de programa, testes e suporte.**

Engenharia de software

Ferramentas

As ferramentas da engenharia de software **fornece**m suporte automatizado ou semiautomatizado para o processo e para os métodos.

Ferramentas CASE – Computer-Aided Software Engineering.

Engenharia de software

TÉCNICAS

As técnicas/procedimentos da engenharia de software organizam as maneiras que o desenvolvimento de software é realizado.

Definem:

- a sequência em que os métodos devem ser aplicados;
- os produtos que devem ser desenvolvidos (relatórios, formulários, documentos, etc);
- os controles que devem ser aplicados para que a qualidade do software seja garantida;
- as formas de como as mudanças devem ser feitas;
- e os marcos de referência para que o progresso de desenvolvimento do software possa ser avaliado.

Para pensar...

Por que existem
tantas falhas de
software?

Quais são os
desafios para
Engenharia de
Software?

Alguns desafios e dificuldades

Complexidade

Falta de utilização de engenharia de software

Lidar com diversidade

Demanda

Prazo de entrega

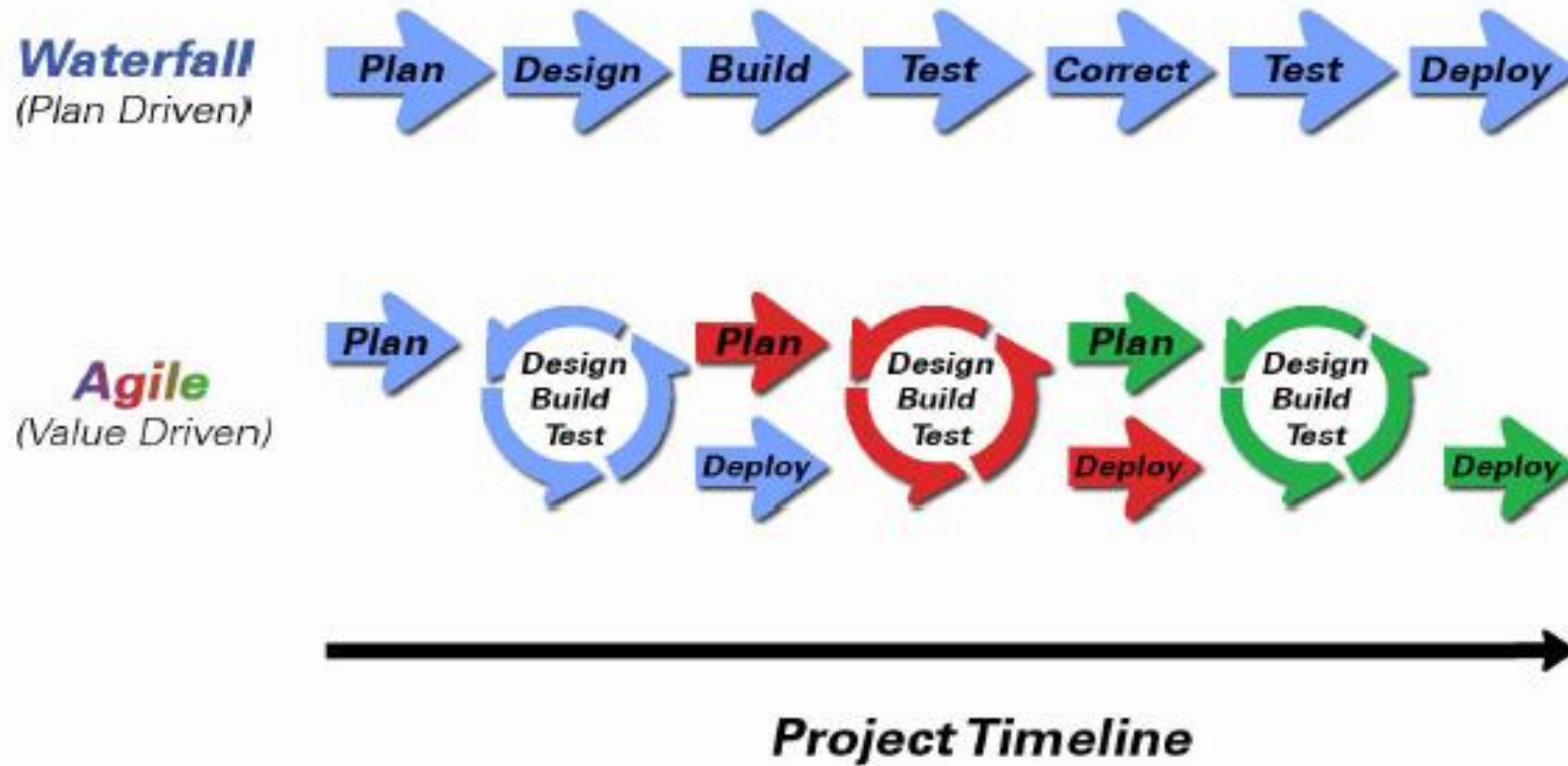
Confiabilidade

Heterogeneidade

Mudanças nos negócios e na sociedade

Segurança da informação e Confiança

METODOLOGIAS



Atividades do Processo de Software

Todo Processo de Software deve incluir de alguma forma as quatro atividades fundamentais:

1. Especificação

2.
Desenvolvimento

3. Validação

4. Evolução

Especificação

Desenvolve-se os **requisitos** de sistema que devem atender os **requisitos** de usuário.

Na especificação passa-se da perspectiva do problema (**requisitos** de usuário) para a perspectiva da solução (**requisitos** de sistema).

O que é um requisito?

Os requisitos são a **descrição do sistema, funções e restrições** que são gerados durante o processo de engenharia de requisitos. (Sommerville, 2019)



Requisitos Funcionais e Não-Funcionais

Requisitos Funcionais

São declarações de funções que o sistema deve fornecer, como o sistema deve reagir a entradas específicas e como deve se comportar em determinadas situações.

-> Pode ser o que o sistema não deve fazer.

Requisitos não-Funcionais

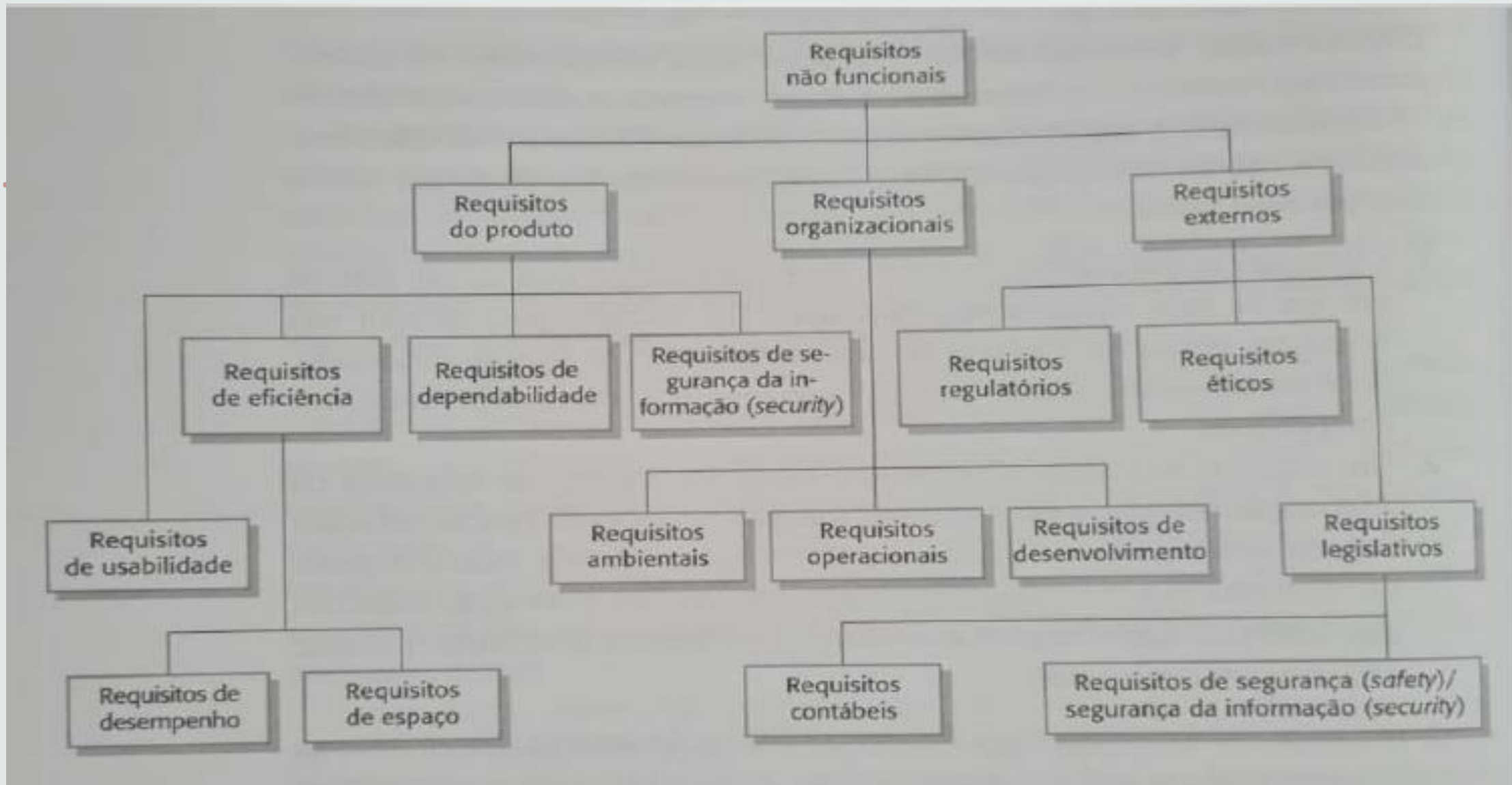
São restrições sobre os serviços ou as funções oferecidas pelo sistema.

Entre eles destacam-se restrições de tempo, restrições sobre o processo de desenvolvimento, padrões, entre outros

Requisito



Requisitos não funcionais podem ser mais importantes que requisitos funcionais individuais pois a falha em não cumprir um requisito não funcional pode tornar o sistema inútil.



Métricas para especificar requisitos Não-Funcionais

Propriedade	Métrica
Velocidade	Transações processadas/segundo
	Tempo de resposta do usuário/evento
	Tempo de atualização da tela
Tamanho	Megabytes/número de chips de ROM
Facilidade de uso	Tempo de treinamento
	Número de quadros de ajuda
Confiabilidade	Tempo médio até a falha
	Probabilidade de indisponibilidade
	Taxa de ocorrência de falhas
	Disponibilidade
Robustez	Tempo para reiniciar após a falha
	Porcentagem de eventos causando falhas
	Probabilidade de corromper dados em uma falha
Portabilidade	Porcentagem de declarações dependentes do sistema-alvo
	Número de sistemas-alvo

Elicitação de Requisitos

É a descoberta de requisitos. Envolve a equipe técnica trabalhando com os clientes para encontrar o domínio da aplicação, os serviços que o sistema deve fornecer e as restrições operacionais do sistema.

Pode envolver usuários finais, gerentes, engenheiros envolvidos em manutenção, especialistas de domínio, etc. **Estes são chamados de stakeholders.**

Engenharia de Requisitos



Fonte: (Sommerville, 2019 p.97)

TÉCNICAS - Elicitação de Requisitos

- **Entrevistas:** questionário
- **Etnografia:** Observação ou etnografia, em que se observa as pessoas executando seu trabalho para ver quais artefatos elas usam, como usam, etc.
- **Histórias e cenários:** descrição de como um sistema pode ser utilizado em uma tarefa específica

Especificação de Requisitos

É o processo de escrever os requisitos de usuário e de sistema em um documento de requisitos.

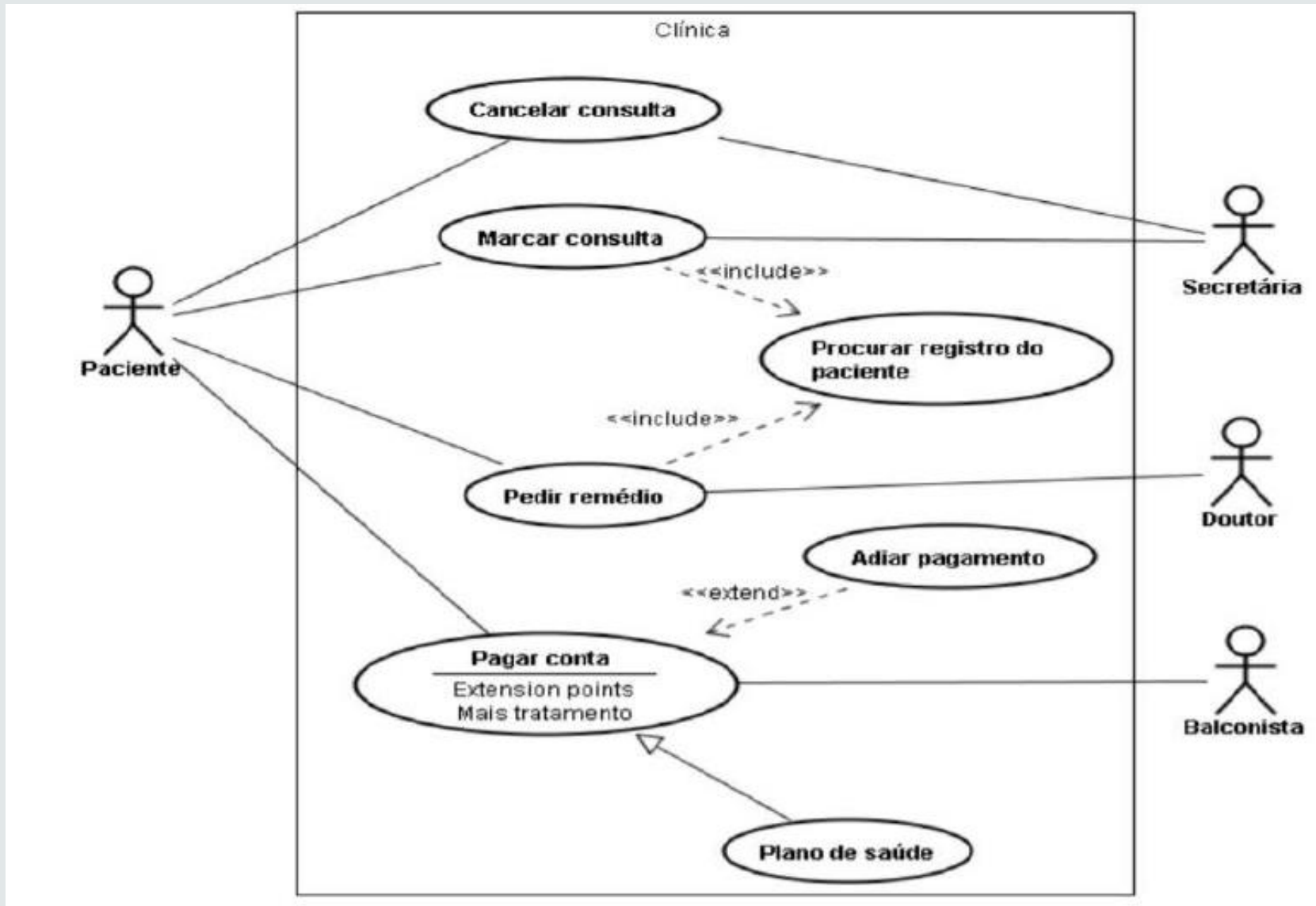
- Os requisitos do usuário devem ser compreensíveis pelos usuários finais e clientes que não possuem conhecimento técnico.
- Os requisitos podem fazer parte de um contrato para o desenvolvimento de sistema.
- Portanto, é importante que estes sejam tão completos quanto possível.

Especificação de Requisitos

Segundo Pressman (2011) na documentação da Especificação de Requisitos os Requisitos Funcionais e os Requisitos Não Funcionais são documentados; nesta etapa podem ser utilizados Diagramas de Casos de Uso (UML).

- Os casos de uso são uma maneira de escrever as interações entre os usuários e um sistema usando um modelo gráfico e um texto estruturado. (Sommerville,2019).

Diagrama de Caso de Uso



Validação de Requisitos

Conferir se os requisitos definem o sistema que o cliente realmente quer.

- Custos de erros de requisitos são altos e, desse modo, a validação é muito importante
- O custo da reparação de um erro de requisitos após a entrega pode equivaler a 100 vezes o custo de reparação de um erro de implementação. (Barry Boehm)

Validação de Requisitos – Como?

Revisões de requisitos:

- Análise manual sistemática dos requisitos.
- Prototipação
- Uso de um modelo executável do sistema para verificar os requisitos.
- Geração de casos de teste.
- Desenvolvimento de testes para verificar os requisitos implementados.

Tarefa 1 – Mapa Mental

Assistir ao vídeo sobre Modelos de Processo de software e Atividades de software

[https://www.youtube.com/watch?v=kO1PSkzTsYc&feature=youtu.](https://www.youtube.com/watch?v=kO1PSkzTsYc&feature=youtu)

Tarefa 2 – Fazer o diagrama de caso de uso

Situação:

A Faculdade Alpha está criando uma aplicação para utilização de vários tipos de usuários (ATORES).

- Os alunos podem: realizar a matrícula pela aplicação, solicitar histórico, solicitar transferência, consultar notas e frequência.
- Quando os alunos solicitam transferência a mesma deverá ser aprovada pelo funcionário da secretaria.
- A secretaria também pode cadastrar, alterar, excluir ou consultar um aluno e realizar a matrícula de um aluno também.
- Os professores também podem incluir, alterar, excluir ou consultar notas dos alunos e a mesma coisa com relação a frequência.
- Todos os usuários necessitam fazer login no sistema e podem fazer logout e recuperar a senha, caso tenham esquecido.