ADS - Engenharia de Software 2025 - Anotações de aula

Professor Miguel Suez Xve Penteado

2025-02-18

Table of Contents

# Sobre estas anotações

Estas anotações são apenas lembretes das aulas expostas em sala, durante a disciplina de ENGENHARIA DE SOFTWARE.

## 0.1 ACESSO AO GITBOOK CELULAR

## 0.2 <https://miguel7penteado.github.io/ADS-EngenhariaSoftware2025>



## 0.3 APP EPUB ANDROID

## 0.4 **Moon+ Reader**



# 1 Livros Texto da Disciplina

#### 1.0.0.1 “Engenharia de Software” do autor “Roger S Pressman”



|  |  |
| --- | --- |
| **Autor(es)** | [Roger S. Pressman](https://www.indicalivros.com/autores?q=Roger%20S.%20Pressman) |
| **Editora** | Pearson |
| **Idioma** | Português |
| **ISBN** | 8534602379 9788534602372 |
| **Formato** | Capa comum |
| **Páginas** | 1056 |
| **Código Biblioteca** |  |

#### 1.0.0.2 “Engenharia de Software” do autor “Ian Sommerville”



|  |  |
| --- | --- |
| **Autor(es)** | Ian SommerVille |
| **Editora** | Pearson |
| **Idioma** | Português |
| **ISBN** | 9788588639072 |
| **Formato** | Capa comum |
| **Páginas** | 768 |
| **Código Biblioteca** |  |

Calendário das aulas

##### 1.0.0.2.1 FEVEREIRO 2025

|  |  |  |
| --- | --- | --- |
| Data | Dia da semana | Aulas |
| 4 de fevereiro | Terça-feira |  |
| 11 de fevereiro | Terça-feira |  |
| 18 de fevereiro | Terça-feira |  |
| 25 de fevereiro | Terça-feira |  |

##### 1.0.0.2.2 MARÇO 2025

|  |  |  |
| --- | --- | --- |
| Data | Dia da semana | Aulas |
| 4 de março | Terça-feira |  |
| 11 de março | Terça-feira |  |
| 18 de março | Terça-feira |  |
| 25 de março | Terça-feira |  |

##### 1.0.0.2.3 ABRIL DE 2025

|  |  |  |
| --- | --- | --- |
| Data | Dia da semana | Aulas |
| 1 de abril | Terça-feira |  |
| 8 de abril | Terça-feira |  |
| 15 de abril | Terça-feira |  |
| 22 de abril | Terça-feira |  |
| 29 de abril | Terça-feira |  |

##### 1.0.0.2.4 MAIO DE 2025

|  |  |  |
| --- | --- | --- |
| Data | Dia da semana | Aulas |
| 6 de maio | Terça-feira |  |
| 13 de maio | Terça-feira |  |
| 20 de maio | Terça-feira |  |
| 27 de maio | Terça-feira |  |

##### 1.0.0.2.5 JUNHO DE 2025

|  |  |  |
| --- | --- | --- |
| Data | Dia da semana | Aulas |
| 3 de junho | Terça-feira |  |
| 10 de junho | Terça-feira |  |
| 17 de junho | Terça-feira |  |
| 24 de junho | Terça-feira |  |

bookdown::render\_book()

# INTRODUÇÃO A DISCIPLINA DE ENGENHARIA DE SOFTWARE

Do que trata esta disciplina e o que quer dizer o termo que dá nome a ela ?

## 1.1 O que é ENGENHARIA DE SOFTWARE



**Engenharia de Software** *é o processo de desenvolvimento de programas de computador, estruturas de dados e documentos.* (***Roger S. Pressman***)



**Engenharia de Software** *é uma disciplina de engenharia que se preocupa com todo o processo de produção de software. Isso inclui desde a especificação do sistema até a sua manutenção.* (**Ian Sommerville**)

É atribuído a **Margaret Hamilton,** desenvolvedora do programa de navegação da APOLLO 11 a criação do termo ENGENHARIA DE SOFTWARE.

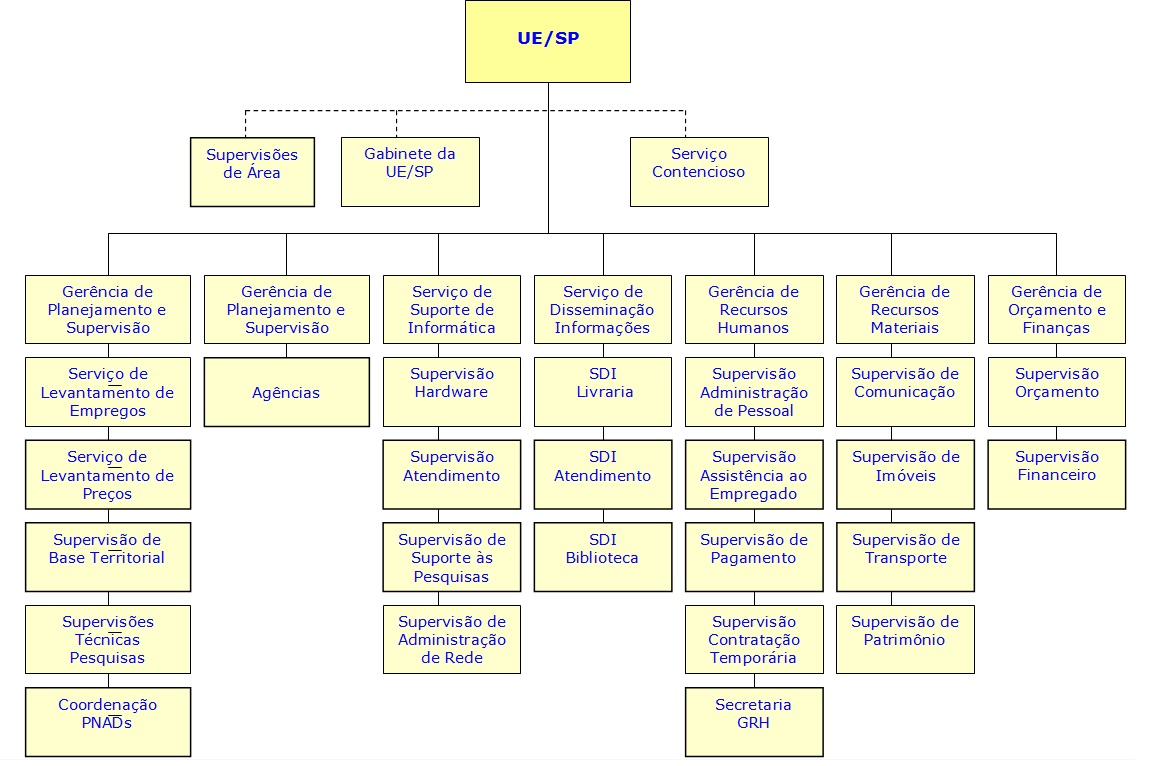


# 2 PROCESSO DE SOFTWARE

## 2.1 CONCEITO DE PROCESSO, ATIVIDADES E TAREFAS

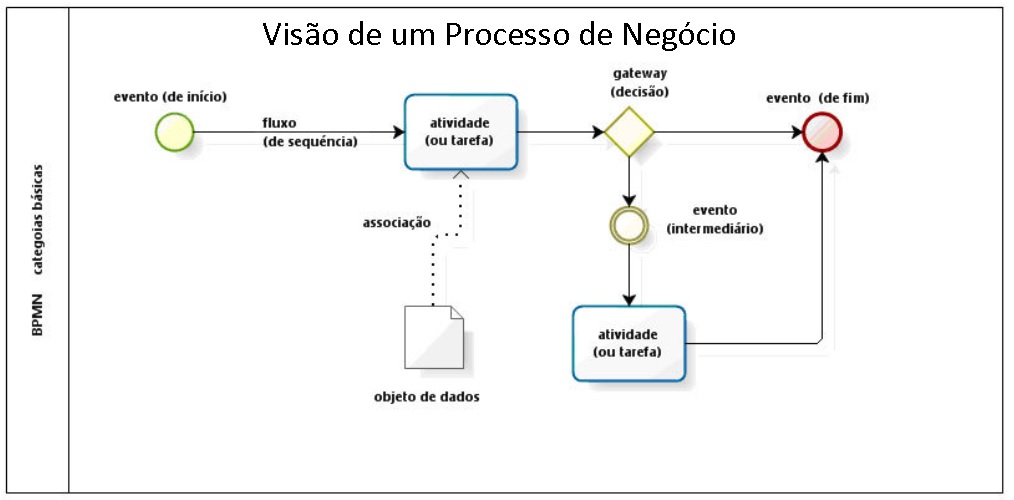
**A ORGANIZAÇÃO**

Um exemplo de organograma : a Unidade do IBGE de São Paulo:



#### 2.1.0.1 OS PROCESSOS

No mundo dos negócios como é conceituado um processo mapeado ?



*Um PROCESSO é um conjunto de atividades que são executadas para atingir um objetivo* (Business Process Model and Notation - BPMN)

*ATIVIDADES são os trabalhos que devem ser realizados para que um processo seja executado* (Business Process Model and Notation - BPMN)

*TAREFAS são decomposições de atividades e representam um conjunto de passos ou ações executadas para realizar um trabalho. Essas tarefas logicamente relacionadas quando encadeadas completam a atividade.* (Business Process Model and Notation - BPMN)

## 2.2 O PROCESSO DE SOFTWARE

*PROCESSO DE SOFTWARE é uma metodologia que define as atividades necessárias para desenvolver um software de qualidade* (Roger S Pressman)

*PROCESSO DE SOFTWARE é um conjunto de atividades que visam a produção de um software* (Ian Sommerville)

## 2.3 ATIVIDADES FUNDAMENTAIS DE UM PROCESSO DE SOFTWARE

**POR QUAIS ATIVIDADES VOCÊ (NECESSÁRIAMENTE) PASSA ?**

PROCESSO DE SOFTWARE - ATIVIDADES - PRESSMAN

| ATIVIDADE #1 | ATIVIDADE #2 | ATIVIDADE #3 | ATIVIDADE #4 | ATIVIDADE #5 |
| --- | --- | --- | --- | --- |
| comunicação | planejamento | modelagem | construção | entrega |

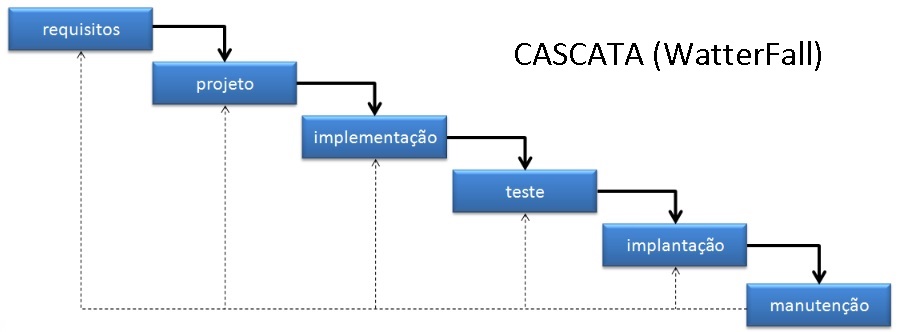
PROCESSO DE SOFTWARE - ATIVIDADES - SOMMERVILLE

| ATIVIDADE #1 | ATIVIDADE #2 | ATIVIDADE #3 | ATIVIDADE #4 | ATIVIDADE #5 |
| --- | --- | --- | --- | --- |
| especificação | projeto | implementação | validação | evolução |

## 2.4 MODELOS DE PROCESSO DE SOFTWARE CLÁSSICOS

**COMO VOCÊ PASSA POR ESSAS ATIVIDADES ?**

### 2.4.1 ABORDAGEM CASCATA



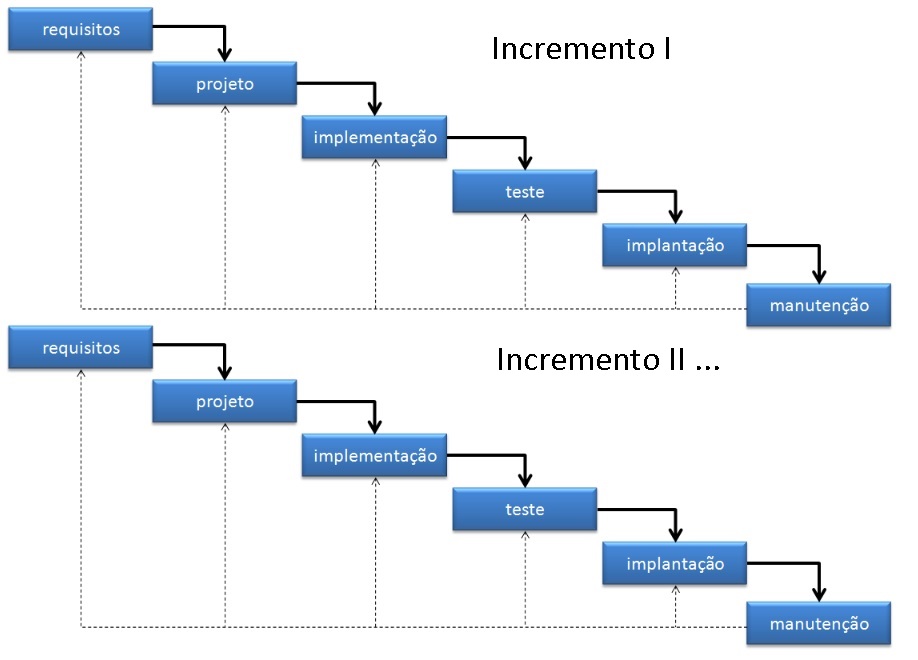
#### 2.4.1.1 VANTAGENS DESSE MODELO

|  |  |  |
| --- | --- | --- |
| Vantagem | Descrição | Exemplo |
| Simplicidade | * fácil de entender e implementar; * pois possui fases sequenciais bem definidas, com entregas e revisões em cada etapa; * facilita o gerenciamento do projeto, especialmente para equipes menores; |  |
| Documentação | * Natureza estruturada do * Exige a produção de documentação completa em cada fase; * Útil para rastrear o progresso, facilitar a comunicação entre as partes; |  |
| Controle | * maior controle sobre o projeto, escopo, os prazos e os custos são definidos no início e seguidos rigorosamente; * Cliente tenha previsão precisa do resultado final e investimento necessário.; |  |
| Adequado para projetos estáveis | * Adequado para projetos com requisitos bem definidos e estáveis e que não devem sofrer muitas alterações ao longo do desenvolvimento. |  |

#### 2.4.1.2 DESVANTAGENS DESSE MODELO

|  |  |  |
| --- | --- | --- |
| Desvantagem | Descrição | Exemplos |
| Inflexibilidade | * Rigidez. Uma vez que uma fase é concluída, é difícil voltar atrás e fazer alterações. * problemático se os requisitos mudarem ou se surgirem problemas inesperados | Se o projeto é feito em um produto comercial (biblioteca) e ela deixa de ser suportada pelo fabricante, o projeto vai ter que nascer com uma tecnologia morta; Seu projeto nasceu no skype, não vai poder migrar para o teams no meio da construção; |
| Feedback tardio | * O cliente só tem contato com o produto final nas últimas fases do projeto; * Difícil identificação de erros ou inadequações; * Atrasos, custos adicionais e insatisfação do cliente; | O cliente iniciou um projeto de aplicativo na plataforma ANDROID. Durante o desenvolvimento o cliente que o aplicativo rode em plataforma IOS (iPHONE). Não é possível, mesmo que o cliente tenha um pequeno montante de dinheiro para investir; |
| Longos prazos | * prazos de entrega mais longos, * cada fase precisa ser concluída antes da próxima ser iniciada. * problemático em projetos com restrições de tempo ou em mercados dinâmicos. | Você definiu o prazo de entrega em 6 meses, mas devido a rotatividade de funcionários, seu projeto atrasou 3 meses; Os funcionários de equipes de etapas dependentes ficam ociosos naquele projeto; |
| Risco de insucesso | Falta de flexibilidade e feedback tardio podem aumentar o risco de o projeto não atender às expectativas do cliente ou às necessidades do mercado. | O cliente do estacionamento comprou um sistema ERP para uma loja de calçados; O sistema foi desenhado para cadastrar calçados; Contudo a loja é uma magazine e o cliente durante o projeto começou a vender roupas; O cadastro de mercadorias não suporta roupas; O cliente só descobriu a falta de suporte no momento da entrega do software; |

### 2.4.2 ABORDAGEM INCREMENTAL



#### 2.4.2.1 VANTAGENS DESSE MODELO

|  |  |  |
| --- | --- | --- |
| Vantagem | Descrição |  |
| Entrega antecipada de software funcional | * permite que o cliente tenha acesso a funcionalidades do software em estágios iniciais do projeto. * possibilita feedback do cliente; |  |
| Melhor gerenciamento de riscos | * Divide o projeto em incrementos menores; * Riscos menores possível identificar e corrigir problemas em etapas iniciais; |  |
| Flexibilidade para mudanças | * Mais adaptável a mudanças nos requisitos do cliente, sem comprometer o andamento do projeto. |  |
| Feedback contínuo do cliente | * incentiva o envolvimento do cliente durante todo o processo de desenvolvimento; * Cliente testa e avalia as funcionalidades em cada incremento; * Produto final torna-se mais alinhado com as expectativas do cliente. |  |
| Melhor aproveitamento de recursos | * Equipe de desenvolvimento trabalhe mais eficiente, concentrando-se em um incremento de cada vez; * “Pode” reduzir os custos do projeto; |  |

#### 2.4.2.2 DESVANTAGENS DESSE MODELO

|  |  |  |
| --- | --- | --- |
| Desvantagem | Descrição |  |
| Requisitos bem definidos | * Se os requisitos não estão bem definidos, o orçamento vai estourar; | Um software foi projetado para ter frontend em janelas e computadores; No primeiro incremento o cliente resolve que é melhor acrescentar suporte a WEB (frontend no navegador); No segundo incremento o cliente resolve que o software deve ter suporte a frontend de dispositivos móveis (app); |
| Planejamento cuidadoso | * Se o incremento não for bem planejado, gera mais incrementos ; Bola de Neve | O time de Planejamento definou que o software seria no code e seria feito em plataforma Buble (onde o Banco de dados é interno da solução). Surgiu a necessidade de integração com outros bancos de dados pré-existentes de terceiros. Precisa mudar a solução para WeWEB. Surgiu a necessidade de inserir gateway de pagamento. Melhor migrar o projeto para python e FLASK; |
| Integração dos incrementos | * Incrementos não são necessáriamente lineares se existirem funcionalidades forem dependentes; | Seu sistema gera um relatório de uso de veículos da empresa por funcionários; Você vai construir banco de dados próprio, mas precisa acessar a base de dados do RH e a base de dados da Gerencia de Materiais da empresa; O acesso da tabela depende de aprovação das áreas; Seu sistema não anda enquanto dos outrso bancos não sair; |
| Necessidade de equipe experiente | * Exige um alinhamento e comunicação muito grande entre a equipe de desenvolvimento; Time entrosado; | Tanto na atividade de planejamento quanto na de execução, Equipe de frontend precisa de informações de APIs da equipe de backend; Equipe de backend precisa de acessos a fontes de dados externas, caso seja necessário; |

### 2.4.3 ABORDAGEM Espiral



### 2.4.4 ABORDAGEM RUP (Rational Unified Process)



## 2.5 MODELOS DE PROCESSO DE SOFTWARE ÁGEIS

### 2.5.1 ABORDAGEM(ENS) ÁGIL(EIS)

Coming soon

# 3 MODELAGEM DE SOFTWARE

Coming soon

# 4 GESTÃO DE QUALIDADE DE SOFTWARE

Coming soon

# 5 GERÊNCIA DE PROJETOS

Coming soon

# 6 Sharing your book

## 6.1 Publishing

HTML books can be published online, see: <https://bookdown.org/yihui/bookdown/publishing.html>

## 6.2 404 pages

By default, users will be directed to a 404 page if they try to access a webpage that cannot be found. If you’d like to customize your 404 page instead of using the default, you may add either a \_404.Rmd or \_404.md file to your project root and use code and/or Markdown syntax.

## 6.3 Metadata for sharing

Bookdown HTML books will provide HTML metadata for social sharing on platforms like Twitter, Facebook, and LinkedIn, using information you provide in the index.Rmd YAML. To setup, set the url for your book and the path to your cover-image file. Your book’s title and description are also used.

This gitbook uses the same social sharing data across all chapters in your book- all links shared will look the same.

Specify your book’s source repository on GitHub using the edit key under the configuration options in the \_output.yml file, which allows users to suggest an edit by linking to a chapter’s source file.

Read more about the features of this output format here:

<https://pkgs.rstudio.com/bookdown/reference/gitbook.html>

Or use:

?bookdown::gitbook