



5954014 - Engenharia de Software

Aula 05 - Gestão de Projetos de Software

Prof. Dr. Denis Martins

DCM | FFCLRP | USP

Caso de Estudo: Projeto UniConnect

Aplicativo Móvel

Aplicativo integrado aos sistemas acadêmicos da Universidade.

Funcionalidades:

- Consultar horários de aulas
- Consultar salas
- Visualizar notas
- Receber avisos acadêmicos
- Consultar calendário acadêmico

Equipe Inicial (8 Pessoas)

Gestão & Requisitos

1 Gerente de Projeto
1 Analista

Desenvolvimento

2 Devs Backend
2 Devs Mobile

Design & Qualidade

1 Profissional de UX
1 Testador

Prazo & Restrição

Prazo Inicial de Execução

5 Meses

Restrição Temporal

Disponibilizar o aplicativo obrigatoriamente **antes** do início do próximo semestre letivo da universidade.

**Se a universidade possui orçamento,
tecnologias adequadas e profissionais
tecnicamente competentes, isso é suficiente
para garantir o sucesso do projeto?**

Objetivos de Aprendizagem

01

Importância & Papéis

Explicar o papel das pessoas e analisar competências técnicas/interpessoais, responsabilidades, organização da equipe, liderança e colaboração.

03

Processo & Escopo

Identificar objetivos, escopo e restrições do projeto, relacionando as características do produto à escolha e adaptação do processo.

02

4 Ps

Explicar e integrar os quatro elementos fundamentais de gerenciamento de software: Pessoas, Produto, Processo e Projeto.

04

Decisões & Impactos

Analisar os impactos de decisões gerenciais em várias dimensões e propor soluções justificadas para problemas reais de projeto.

Projetos de software raramente falham apenas por limitações técnicas.

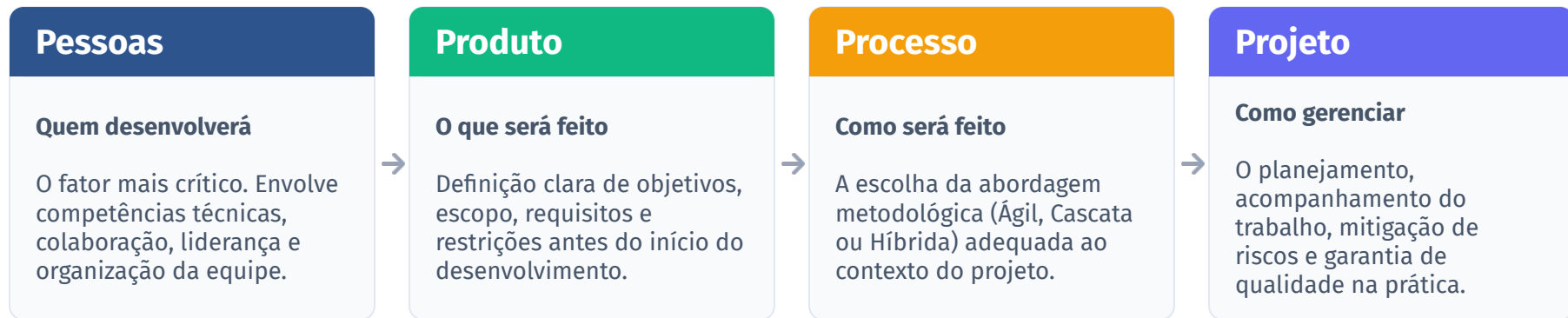
Na prática, **atrasos**, **retrabalho** e **baixa qualidade** muitas vezes surgem de problemas de comunicação, papéis mal definidos, conflitos, baixa colaboração e liderança insuficiente.

A Gerência de **Projetos envolve o planejamento, monitoramento e controle das **Pessoas** e **Processos** que ocorrem conforme evolui o desenvolvimento do software (**Produto**).**

Gestão de Projetos de Software: Os 4 Ps

Visão de Pressman & Maxim para estruturar o sucesso além da programação

Pessoas + **Produto** + **Processo** = **Projeto de Software**



Importante: A ordem **não** é arbitrária! Devem ser selecionadas, organizadas e motivadas para realizar o trabalho efetivamente. Muitos problemas atribuídos à tecnologia e à capacidade técnica são, na realidade, falhas humanas e gerenciais.

Gestão de Projetos de Software: Os 4 Ps (Exemplo)

Cenário de Exemplo

"3 devs para entregar um grande sistema em 4 meses."

Esta é uma situação típica de sobrecarga onde o planejamento inicial é irrealista.

Para mitigar o fracasso, o gestor deve agir nos 4 Ps.

Pessoas

Reorganização

Redistribuir tarefas, alinhar papéis de acordo com as competências de cada dev e focar na motivação.

Produto

Revisão do Escopo

Negociar e reduzir o escopo (MVP), priorizando apenas as funcionalidades mais críticas para o negócio.

Processo

Adaptação ágil

Adotar ritos mais simples, remover burocracias e acelerar ciclos de feedback e entrega contínua.

Projeto

Revisão do Planejamento

Ajustar o cronograma de entregas intermediárias, reavaliar riscos de atraso e gerenciar expectativas.

Gestão de Projetos de Software: **Pessoas**

As Relações de Desenvolvimento: Software não existe no vácuo; ele é um produto de colaboração social projetado para resolver problemas humanos reais:

- **Desenvolvido por equipes** multidisciplinares que precisam de sinergia, comunicação clara e papéis bem desenhados.
- Construído especificamente para **atender necessidades** críticas de clientes, usuários finais e demais interessados (stakeholders).

Stakeholders: Stakeholders são **pessoas ou grupos que afetam ou são afetados** pelo projeto, influenciando diretamente o sucesso ou o fracasso do desenvolvimento de software.

Equipe & Gestão

Internos do Projeto

Envolve a equipe técnica (desenvolvedores, designers, QAs) e gestores responsáveis pela execução e liderança.

Clientes & Usuários

Foco do Produto

Aqueles que financiam, adquirem ou utilizam o software no dia a dia, definindo as necessidades e o valor gerado.

Negócio & Parceiros

Alinhamento Organizacional

Áreas de negócio da empresa e fornecedores externos que viabilizam e integram a solução.

Reguladores

Fatores Externos

Órgãos reguladores e de fiscalização que impõem restrições, conformidades e padrões legais ao projeto.

Atenção à Gestão de Expectativas: Identificar e engajar corretamente todos os stakeholders desde o início mitiga riscos críticos de escopo, comunicação e aceitação do produto final.

Atividade Prática: Identificação de Stakeholders

Instruções

Considerando o contexto do projeto **UniConnect**, reúnam-se em grupo e identifiquem os seguintes pontos:

1. **Pessoas e Grupos:** Quem são os principais envolvidos no ecossistema do sistema?
2. **Influência:** Quem são os stakeholders que exercem maior influência direta ou indireta sobre o projeto?
3. **Competências:** Quais habilidades técnicas e interpessoais serão fundamentais para o sucesso do time?
4. **Desafios:** Quais problemas e conflitos relacionados a pessoas podem emergir durante a execução?

Tabela de Stakeholders		
Pessoa / Grupo	Papel no Projeto	Interesse / Responsabilidade
Estudantes	Usuários finais	Utilizar o aplicativo, fornecer feedback e avaliar as funcionalidades no dia a dia.
[Inserir Grupo]	[Ex: Devs, PO, etc]	[Descrever o principal interesse ou responsabilidade técnica/gerencial]

**Todas as pessoas importantes para o projeto
fazem parte da equipe de desenvolvimento?**

Gestão de Projetos de Software: Pessoas

O sucesso no desenvolvimento de software exige um equilíbrio estratégico entre o saber técnico e a excelência comportamental.

Competências Técnicas

Conhecimentos e habilidades práticas fundamentais para executar com rigor as atividades operacionais do projeto:

- **Construção & Engenharia:** Programação estruturada, arquitetura de sistemas robustas e modelagem de banco de dados.
- **Processos & Qualidade:** Engenharia de requisitos, testes automatizados e design de experiência de usuário (UX).
- **Domínio de Negócio:** Entendimento prático das regras do setor para gerar soluções de software alinhadas às necessidades reais.

Competências Interpessoais

Habilidades comportamentais e sociais que determinam a capacidade de trabalhar bem com outras pessoas:

- **Comunicação & Sinergia:** Colaboração ativa, escuta empática, troca de feedback construtivo e clareza na transmissão de ideias.
- **Negociação & Mediação:** Resolução saudável de conflitos no dia a dia e alinhamento de expectativas técnicas e de prazos.
- **Responsabilidade:** Comprometimento com o sucesso mútuo da equipe e com as entregas planejadas.

Em um projeto real, o que seria **mais difícil de resolver: uma **deficiência técnica** ou um **conflito persistente** entre membros da equipe?**

Gestão de Projetos de Software: **Pessoas**

A definição clara de funções e deveres assegura o alinhamento operacional e a eficiência do time.

O que é um Papel?

Representa a **função formal ou informal** que um profissional assume e exerce ativamente no contexto do projeto. **Papel != Cargo.**

Define como a pessoa se posiciona no fluxo de trabalho e qual é o seu escopo de atuação técnica ou de liderança.

O que é Responsabilidade?

Refere-se àquilo pelo qual o profissional **deve responder diretamente**, prestando contas sobre entregas e resultados.

Garante que cada atividade crítica tenha um dono responsável por assegurar sua conclusão com qualidade.

Equipes e Flexibilidade:

Em equipes pequenas, **uma mesma pessoa pode exercer mais de um papel** simultaneamente. Essa versatilidade otimiza a alocação de recursos, embora exija atenção redobrada para evitar sobrecarga e conflitos de interesse.

Leia também: Quiet Quitting (desistência/demissão silenciosa) em <https://conexao.pucminas.br/blog/dicas/quiet-quitting/> e <https://cadernosdepsicologia.org.br/index.php/cadernos/article/view/308/194>

Gestão de Projetos de Software: Pessoas

P-CMM

Modelo desenvolvido pelo SEI para avaliar e evoluir a capacidade de gerenciar pessoas.

- Organiza a evolução das práticas de gestão de pessoas em níveis de maturidade
- Busca aumentar competência, desempenho e retenção de profissionais
- Enfatiza o desenvolvimento contínuo de competências
- Relaciona práticas de pessoas aos objetivos organizacionais
- Complementa modelos de maturidade voltados a processos de software

Leia na íntegra: [Documento Oficial P-CMM \(SEI/CMU\)](#)



The 5 Levels of PCMM – A Hierarchical View of Organizational Maturity

Fonte da Imagem: holistiquetraining.com

Características de um bom Engenheiro de Software

Segundo Pressman & Maxim.

Combina conhecimento técnico com postura profissional.

Não basta apenas dominar linguagens e algoritmos; o sucesso depende de **competências** que viabilizam o **trabalho em equipe** e **entregas de valor**.

Capacidade Analítica

Decomposição de problemas complexos em partes menores e desenho de soluções lógicas estruturadas.

Colaboração

Atuação ativa em equipe, facilidade em compartilhar conhecimentos e buscar o sucesso conjunto do projeto.

Responsabilidade

Comprometimento com prazos acordados, qualidade do código e impactos das decisões arquiteturais.

Gestão de Projetos de Software: Pessoas

Construindo uma Equipe: Uma equipe consistente **mantém estabilidade, confiança e entendimento compartilhado** ao longo do tempo. A permanência dos membros reduz drasticamente o custo de coordenação e acelera entregas.

Sinergia e Aprendizado Coletivo

Com o tempo, os membros da equipe aprendem como trabalhar juntos de forma natural e intuitiva:

- **Redução de Conflitos:** Alinhamento de expectativas minimiza atritos operacionais e divergências cotidianas.
- **Comunicação Fluida:** Desenvolvimento de jargões próprios e canais de comunicação mais ágeis e eficientes.

Estabilidade e Confiança

A segurança psicológica e a previsibilidade criam um ambiente propício à inovação:

- **Melhoria da Coordenação:** Processos internos e papéis se consolidam, eliminando gargalos de fluxo de trabalho.
- **Cultura de Confiança:** Menos necessidade de microgerenciamento e maior autonomia com responsabilidade compartilhada.

Papéis Claros: Definição precisa de responsabilidades individuais para evitar duplicidade de tarefas ou lacunas operacionais.

Atividade Prática: Monte a Equipe do Projeto UniConnect

Cenário e Restrição Orçamentária

A universidade informou que a equipe poderá contar com apenas **cinco profissionais diretamente alocados** ao projeto. Os grupos devem estruturar e priorizar a equipe ideal enfrentando decisões difíceis de trade-off.

Perfis Disponíveis

- Gerente de Projeto
- Analista de Requisitos
- Desenvolvedor Backend
- Desenvolvedor Mobile
- UX/UI Designer
- Testador
- DBA (Administrador de Banco)
- Especialista em Segurança
- Especialista em Sist. Acadêmico

Matriz de Alocação

Monte a tabela preenchendo as colunas abaixo para cada um dos **cinco perfis selecionados**:

- ▶ **Perfil Escolhido**: Qual a função priorizada?
— Exemplo: Desenvolvedor Backend
- ▶ **Responsabilidade**: O que fará no projeto?
- ▶ **Competências Técnica e Interpessoal necessárias**

Qual perfil vocês decidiram deixar de fora da equipe e qual o principal risco que essa ausência cria para o projeto?

Gestão de Projetos de Software: Pessoas

A toxicidade ocorre quando comportamentos nocivos prejudicam diretamente a **confiança, a segurança psicológica e a colaboração**. Identificar esses padrões precocemente é vital para a sobrevivência do projeto.

Sinais de Alerta Ativos

Comportamentos expressivos e diretos que geram atrito imediato e hostilidade no ambiente de trabalho:

- **Comunicação Agressiva:** Interações baseadas em hostilidade, interrupções constantes, sarcasmo ou tom de voz inadequado.
- **Culpabilização (Blame Culture):** Foco excessivo em apontar culpados por falhas em vez de buscar soluções coletivas e aprendizado.
- **Falta de Respeito:** Desvalorização de ideias, descumprimento de acordos e menosprezo por perspectivas alheias.

Padrões de Corrosão Silenciosa

Atitudes passivo-agressivas e barreiras estruturais que isolam membros e bloqueiam o desenvolvimento:

- **Isolamento:** Exclusão deliberada de membros em discussões importantes ou na tomada de decisão informal.
- **Competição Destrutiva:** Busca pelo sucesso individual acima do sucesso do projeto, ocultando informações estratégicas.
- **Resistência a Feedback:** Recusa em escutar avaliações construtivas, mantendo comportamentos defensivos e ineficientes.

Gestão de Projetos de Software: Pessoas

Liderança: A capacidade de orientar e capacitar a equipe para atingir os objetivos estratégicos

Direcionamento e Suporte

Definir Prioridades: Estabelecer com clareza os focos de atuação, alinhando o esforço diário aos marcos do projeto.

Apoiar Tomada de Decisão: Fornecer contexto técnico e segurança para descentralizar escolhas cruciais.

Resolver Impedimentos: Atuar de forma ativa para remover barreiras organizacionais que travam a evolução.

Facilitação e Coesão

Lidar com Conflitos: Mediar divergências de forma construtiva.

Estimular Colaboração: Fomentar a troca contínua de conhecimento, integrando competências.

Manter Foco nos Objetivos: Blindar a equipe de distrações externas, garantindo o alinhamento com a entrega final.



Projetos de software precisam equilibrar o que será construído, como será desenvolvido e como o trabalho será organizado.

Gestão de Projetos de Software: **Produto**

O **produto** representa o sistema final concebido para sanar uma necessidade concreta, atuando como o elemento central que orienta todas as frentes de engenharia.

Alcance do Software

Primeira Atividade

A determinação do **alcance** é o passo inicial e crucial da gerência de projetos de software.

Objetivo Central

Delimitar e compreender as fronteiras, entradas, saídas, funções e restrições de desempenho do sistema.

Perguntas Principais para a Determinação:

- Como o software a ser construído se **encaixa** em um sistema, produto ou contexto de negócio **maior**?
- Que **objetos de dados** visíveis para o cliente são produzidos como **saída** do software?
- Que **objetos de dados** são exigidos como **entrada**?
- Que **funções** o software realiza para transformar dados de entrada em saída?
- Existe alguma característica especial de **desempenho** a ser considerada?

Gestão de Projetos de Software: **Produto**

O **produto** representa o sistema final concebido para sanar uma necessidade concreta, atuando como o elemento central que orienta todas as frentes de engenharia.

Objetivos

Direção Estratégica

Estabelecem com clareza o que o produto pretende **alcançar** de fato no mercado ou na organização.

Alinhamento

Orientam diretamente as decisões sobre **funcionalidades**, **prioridades** de desenvolvimento, escopo e os critérios de aceitação e sucesso do projeto.

Exemplo inadequado:

Criar um aplicativo moderno para estudantes.

Exemplo mais claro:

Desenvolver um aplicativo que permita aos estudantes consultar horários de aulas, salas e alterações na programação acadêmica.

Objetivos ajudam a orientar decisões posteriores.

Gestão de Projetos de Software: **Produto**

O **produto** representa o sistema final concebido para sanar uma necessidade concreta, atuando como o elemento central que orienta todas as frentes de engenharia.

Escopo

Limites de Entrega

Delimitam com precisão as fronteiras do produto e tudo o que será construído ao longo do ciclo.

Foco e Priorização

Determina explicitamente o que **faz parte das entregas**, o que ficará de fora do planejamento inicial e quais recursos terão prioridade máxima.

O escopo ajuda a responder:

- O que **faz parte** do produto e o que **não faz parte**;
- Quais **funcionalidades** principais e limites existem.

Exemplo: Sistema para restaurante

Dentro do escopo:

- Cadastro de produtos
- Realização de pedidos
- Controle básico de estoque
- Pagamento

Fora do escopo:

- Sistema contábil completo
- Folha de pagamento
- Previsão automática de demanda

Gestão de Projetos de Software: **Produto**

O **produto** representa o sistema final concebido para sanar uma necessidade concreta, atuando como o elemento central que orienta todas as frentes de engenharia.

Restrições

Fatores Condicionantes

Variáveis do ambiente que limitam e moldam as decisões de arquitetura e gerência de projeto.

Principais Eixos

Envolvem limites estritos de **prazo, orçamento, equipe** disponível, decisões tecnológicas, marcos regulatórios ou conformidade com sistemas legados.

Exemplos de Restrições comuns:

- Orçamento limitado
- Prazo de entrega estrito
- Tecnologias existentes
- Equipe disponível
- Legislação e normas
- Infraestrutura de TI
- Sistemas legados integrados

Exemplo de Aplicação Prática

"O sistema deve entrar em operação em **quatro meses** e obrigatoriamente utilizar o **banco de dados já adotado** pela empresa."

Neste cenário real, identificamos claramente duas restrições fundamentais para o planejamento: uma de **prazo** (4 meses) e outra **tecnológica** (SGBD predefinido).

Gestão de Projetos de Software: **Processo**

Pergunta central: **Como desenvolveremos o produto?** O processo define a dinâmica de trabalho.

Estruturação & Foco

Elementos Indispensáveis:

- atividades a realizar
- entregas planejadas
- responsabilidades e papéis
- verificação e garantia
- feedback de stakeholders
- evolução do produto

Foco em Resultados

Garante que o esforço coletivo caminhe para entregas consistentes e com previsibilidade.

Características que Moldam o Processo:

Estabilidade de Requisitos

Saber se mudam muito ou pouco ao longo do desenvolvimento.

Criticidade e Impacto

O tamanho do risco associado a falhas no produto final.

Tamanho do Projeto

A escala global de esforço, equipe e artefatos gerados.

Demanda por Feedback

A frequência necessária de iterações com stakeholders.

Experiência da Equipe

Domínio técnico das tecnologias e do processo de trabalho.

Restrições Ambientais

Fatores externos, técnicos e limites do próprio ecossistema.

Gestão de Projetos de Software: **Projeto**

Distinção Básica: **Produto** é aquilo que será construído | **Projeto** é o esforço organizado para construí-lo.

Pergunta central: *Como organizar e controlar o trabalho para produzir o software?*

Frentes de Trabalho

Definição de objetivos, delimitação de escopo, divisão do trabalho e atribuição de responsabilidades.

Execução e Controle

Acompanhamento contínuo, estimativas de recursos/prazos e identificação precoce de problemas.

Estimativas, riscos, cronograma e planos de projeto serão detalhados nas próximas unidades.

Quatro dimensões de Projeto

As quatro dimensões interdependentes que limitam e moldam cada decisão de engenharia:

Escopo, Prazo, Custo/Recursos, Qualidade

Alterar qualquer uma dessas dimensões sem impactar as demais é matematicamente inviável.

Gestão de Projetos de Software: **Projeto**

Princípio 5W2H: Um conjunto de perguntas essenciais usado para caracterizar o projeto e guiar a elaboração do plano de projeto, conectando problema, produto, equipe, processo e recursos.

Perguntas Essenciais

Why? Por que o sistema é necessário?

What? O que será entregue e construído?

When? Quando as entregas ocorrerão?

Who? Quem participará e decidirá?

Where? Onde o trabalho será realizado?

How? Como o processo será conduzido?

How much? Quanto de esforço e recurso?

Exemplo: App Acadêmico

Why?

Reduzir dúvidas e melhorar comunicação acadêmica.

What?

Aplicativo móvel integrado ao sistema acadêmico.

When?

Primeira versão funcional em 5 meses.

Who?

Equipe técnica, secretaria, alunos e gestores.

How?

Entregas incrementais com validação constante.

Erros Comuns de Alunos

Respostas Genéricas

Preencher as perguntas com dados óbvios ou abstratos sem utilidade real.

Desejos Excessivos

Confundir "What?" com uma lista infinita de funcionalidades inviáveis.

How Much Reducionista

Tratar o custo estritamente como valor financeiro, ignorando esforço de equipe.

Atividade Prática: 4Ps e Impactos de Mudanças

Em grupos. Instruções em:

denmartins.github.io/labs/4Ps-impactos



⌚ Ao final, cada grupo terá aproximadamente **1 minuto** para apresentar os resultados.

01. Problema

O problema ou cenário de mudança recebido pelo grupo.

02. P Afetado

O P diretamente afetado (Pessoas, Produto, Processo ou Projeto).

03. Decisão

A decisão de gerenciamento recomendada para mitigar o impacto.

04. Trade-off

O principal compromisso ou trade-off identificado na escolha.

Conclusão

Gerenciar projetos de software significa coordenar **pessoas**, definir adequadamente o **produto**, adotar um **processo** compatível com o contexto e organizar o **projeto** considerando seus objetivos e restrições.

Pessoas

Equipes eficazes precisam de confiança, comunicação e objetivos comuns para atuar em sintonia.

Produto

Precisa ter objetivos claros, escopo bem definido e todas as restrições mapeadas e conhecidas.

Processo

Deve ser flexível e adequado às características da equipe e ao contexto específico do projeto.

Projeto

Organiza, acompanha e controla de forma ativa todo o trabalho necessário para entregar o software.

Dinâmica de Interdependência & Trade-offs

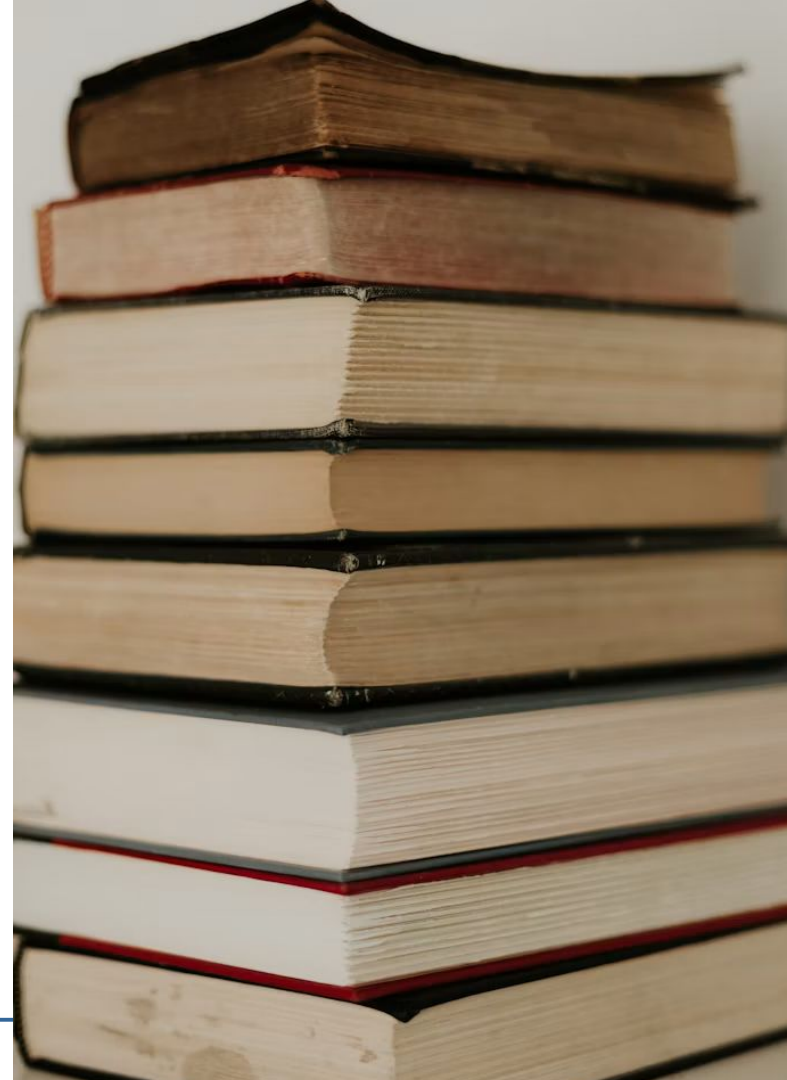
Pessoas, Produto, Processo e Projeto são dimensões interdependentes. Mudanças em prazo, equipe ou requisitos exigem ajustes constantes. Gerenciar software envolve tomar decisões difíceis e lidar continuamente com **trade-offs**.

Referências

PRESSMAN, R. S.; MAXIM, B. R. Engenharia de Software: Uma Abordagem Profissional. 9. ed.
Capítulos 5 e 24.

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

IEEE COMPUTER SOCIETY. Guide to the Software Engineering Body of Knowledge — SWEBOK Guide.
Version 4.0. IEEE Computer Society, 2024



Dúvidas e Discussão

Veja também:

<https://pt.linkedin.com/pulse/os-4-ps-da-engenharia-de-software-um-framework-para-crescer-ghisi>

[Communicating and Working with Stakeholders | Google Project Management Certificate](#)