

Fundamentos de Gerenciamento de Projetos

PMBOK e as 10 áreas de conhecimento



Plataforma completa de aprendizado
contínuo em programação.

#BoostingPeople

rocketseat.com.br

Todos os direitos reservados © Rocketseat S.A.

Aula 1

PMBOK e as 10 áreas de conhecimento

O **PMBOK** (*Project Management Body of Knowledge - Conjunto de Conhecimentos sobre a Gestão de Projetos*) é um guia de melhores práticas em gerenciamento de projetos, publicado pelo **PMI (Project Management Institute)**.

Ele não é uma metodologia em si, mas um conjunto de padrões e diretrizes que organizam o conhecimento em **10 áreas de conhecimento** e **49 processos**, agrupados em **5 grupos de processos**.

Na área de tecnologia, o PMBOK ajuda a estruturar projetos de TI (como desenvolvimento de software, migração para a nuvem, implementação de ERP, etc.) de forma sistemática, reduzindo riscos e aumentando a eficiência.

10 Áreas de Conhecimento do PMBOK (e sua Aplicação em Tecnologia)

Cada área cobre um aspecto crítico da gestão de projetos, e em TI, elas se adaptam aos desafios específicos da área.

Integração

- **Objetivo:** Garantir que todas as partes do projeto estejam alinhadas.
- **Aplicação em TI:**
 - Coordenar equipes de devs, QA, infraestrutura e negócios.
 - Documentar mudanças de escopo (ex.: ajustes em requisitos de software).
- **Processos-chave:** Desenvolver o *Termo de Abertura do Projeto (TAP)* e o *Plano de Gerenciamento do Projeto*.

Escopo

- **Objetivo:** Definir e controlar o que será entregue.
- **Aplicação em TI:**
 - Evitar *scope creep* (ex.: funcionalidades adicionais não planejadas).
 - Usar técnicas como *backlog* (em Agile) ou *EAP* (*Estrutura Analítica do Projeto*).
- **Processos-chave:** Coletar requisitos e validar entregas.

Cronograma

- **Objetivo:** Gerenciar prazos e dependências.
- **Aplicação em TI:**
 - Criar *roadmaps* com marcos (ex.: sprints, releases).
 - Usar ferramentas como **Gantt (MS Project)** ou **Jira** para acompanhamento.
- **Processos-chave:** Definir atividades, estimar durações e sequenciar tarefas.

Custos

- **Objetivo:** Controlar orçamento.
- **Aplicação em TI:**
 - Calcular custos de licenças, cloud computing, equipe e terceiros.
 - Métricas como *ROI* e *Valor Agregado (EVM)*.
- **Processos-chave:** Estimar custos e determinar orçamento.

Qualidade

- **Objetivo:** Garantir que o produto atenda aos padrões.
- **Aplicação em TI:**
 - Testes automatizados, revisões de código e métricas (ex.: *bug rate*).
 - Frameworks como **ISO 25010** (qualidade de software).
- **Processos-chave:** Planejar a qualidade e realizar auditorias.

Recursos

- **Objetivo:** Gerenciar equipe e infraestrutura.
- **Aplicação em TI:**
 - Alocar devs, DBAs, UX designers, etc.
 - Gerenciar recursos de cloud (ex.: escalabilidade AWS/Azure).
- **Processos-chave:** Mobilizar e gerenciar a equipe.

Comunicações

- **Objetivo:** Garantir fluxo eficiente de informações.
- **Aplicação em TI:**
 - *Daily meetings* (em Agile), relatórios para stakeholders.
 - Ferramentas como **Slack**, **Confluence** ou **Microsoft Teams**.
- **Processos-chave:** Planejar e distribuir informações.

Riscos

- **Objetivo:** Identificar e mitigar ameaças.
- **Aplicação em TI:**
 - Riscos comuns: falhas de segurança, atrasos em integrações, *tech debt*.
 - Técnicas como **Análise SWOT** ou **Matriz de Probabilidade/Impacto**.
- **Processos-chave:** Identificar, analisar e responder a riscos.

Aquisições

- **Objetivo:** Gerenciar contratos e fornecedores.
- **Aplicação em TI:**
 - Terceirização de serviços (ex.: DevOps, testes).
 - Licenciamento de softwares (ex.: SAP, Oracle).
- **Processos-chave:** Realizar aquisições e gerenciar contratos.

Partes Interessadas (Stakeholders)

- **Objetivo:** Engajar clientes, patrocinadores e equipe.
- **Aplicação em TI:**
 - Alinhar expectativas entre áreas de negócio e TI.
 - Mapear influência e interesse (ex.: matriz de poder).
- **Processos-chave:** Identificar e envolver stakeholders.

Como o PMBOK se aplica em projetos de tecnologia?

- **Metodologias Híbridas:** Combina PMBOK com Agile (ex.: Scrum para desenvolvimento, mas com controle de custos e riscos do PMBOK).
- **Governança de TI:** Alinhamento com frameworks como **COBIT** ou **ITIL**.
- **Documentação:** Criação de artefatos como *plano de gerenciamento de riscos* ou *termo de aceite*.

Ferramentas Associadas ao PMBOK em TI

- **MS Project / ClickUp** (cronograma).
- **Jira + Confluence** (gestão ágil + documentação).
- **Trello** (gestão visual de tarefas).



O PMBOK oferece uma estrutura sólida para projetos de tecnologia, mas deve ser adaptado ao contexto (ex.: projetos ágeis exigem flexibilidade).

Dominar suas áreas de conhecimento ajuda a evitar falhas comuns, como estouros de orçamento ou entregas fora do escopo.

OBRIGADO! :)

Fundamentos de Gerenciamento de Projetos

Metodologias ágeis (Scrum, Kanban, XP)



Plataforma completa de aprendizado
contínuo em programação.

#BoostingPeople

rocketseat.com.br

Todos os direitos reservados © Rocketseat S.A.

Aula 2

Metodologias ágeis (Scrum, Kanban, XP)

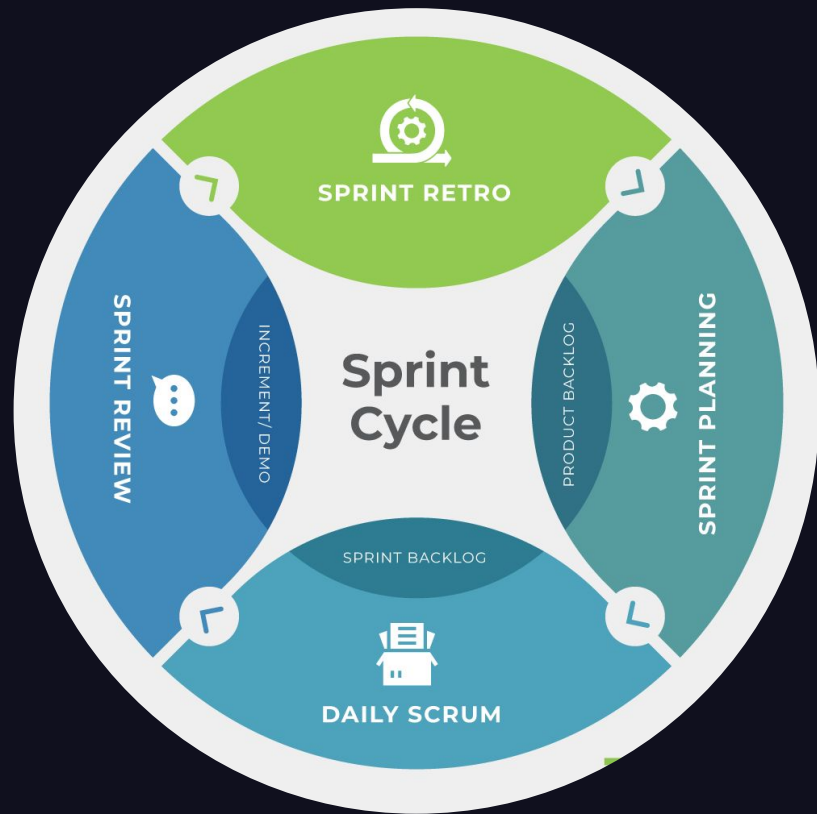
As metodologias ágeis são abordagens iterativas e flexíveis para gerenciamento de projetos, especialmente eficazes em **desenvolvimento de software e projetos de TI**, onde os requisitos mudam frequentemente.

As três principais metodologias são **Scrum, Kanban e Extreme Programming (XP)**, cada uma com suas particularidades.

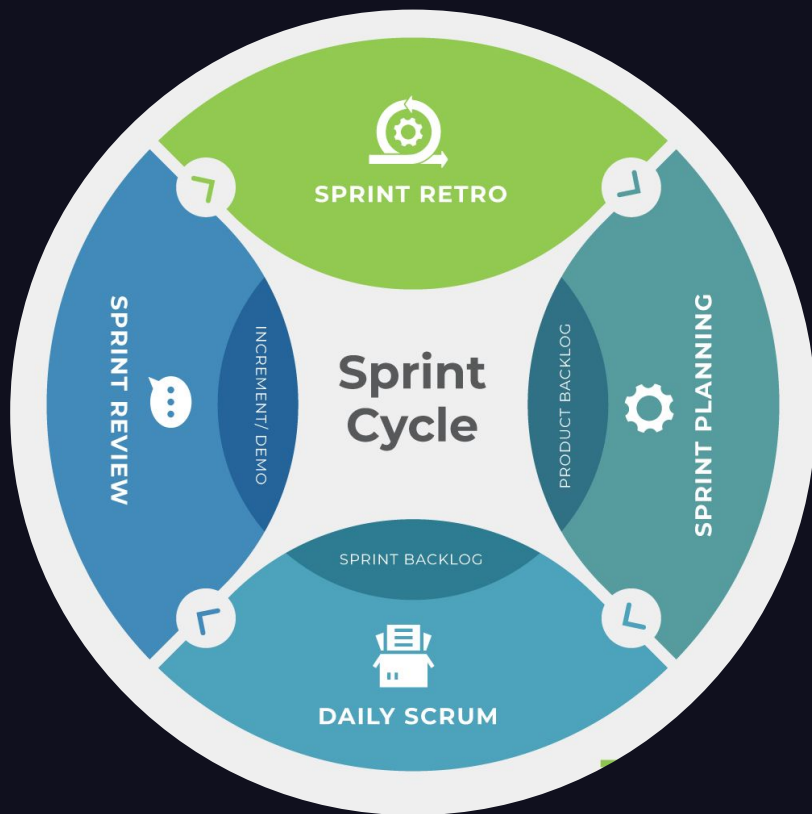
Scrum

O que é?

- Framework ágil estruturado em **sprints** (ciclos curtos de trabalho, geralmente 2–4 semanas).
- Focado em **entregas incrementais**, priorizando funcionalidades de maior valor.



Principais características do Scrum

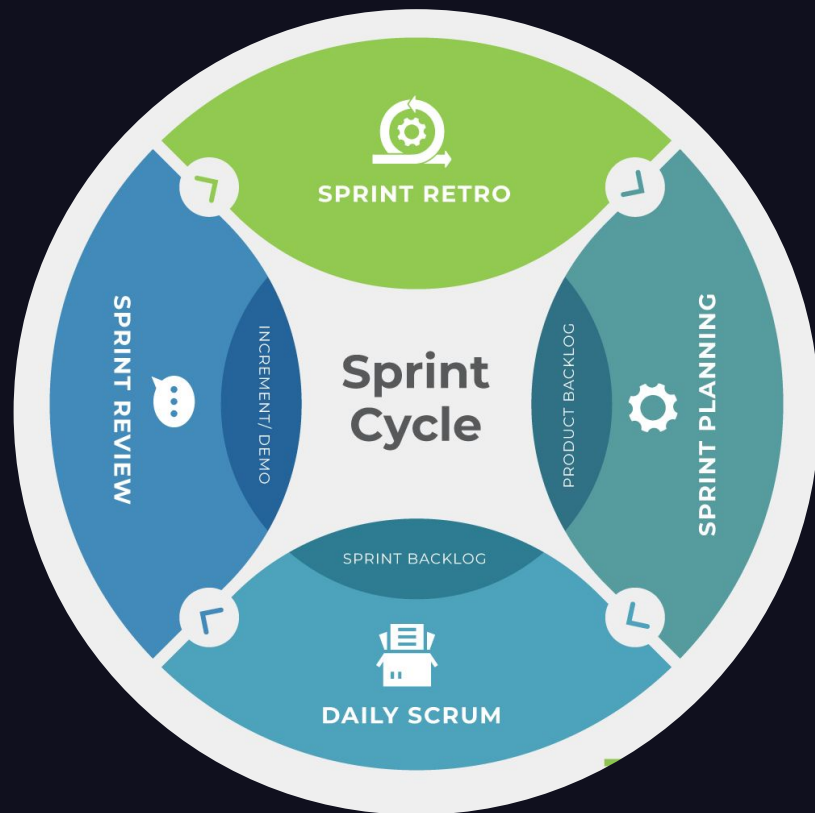


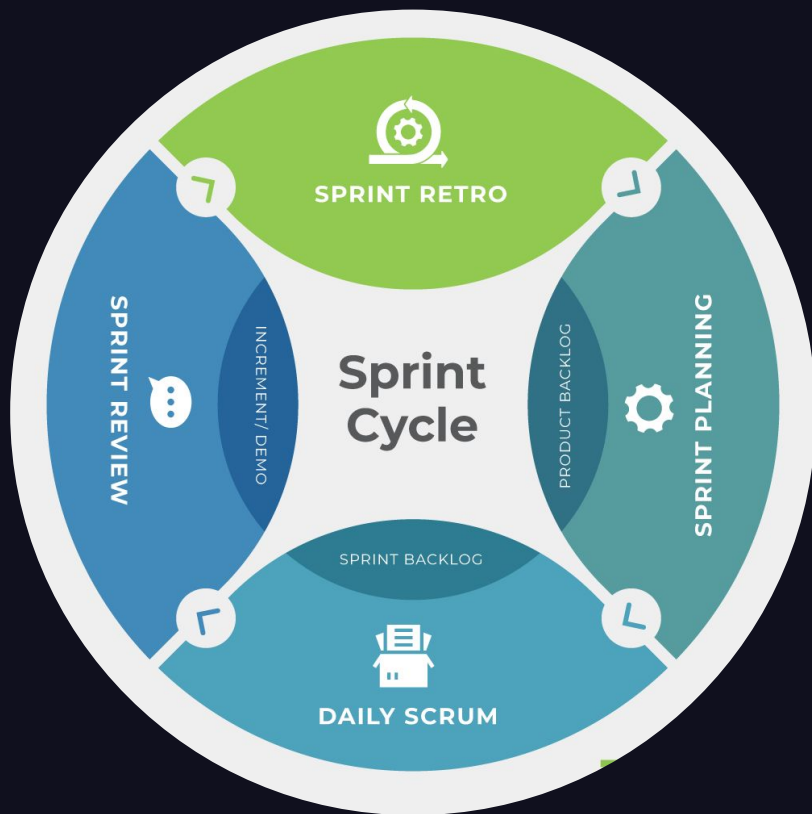
✓ Papéis Definidos:

- **Product Owner (PO):** Define o backlog e prioriza tarefas.
- **Scrum Master:** Remove obstáculos e garante que o time siga o Scrum.
- **Time de Desenvolvimento:** Executa as tarefas (devs, testadores, etc.).

✓ Eventos (Cerimônias):

- **Sprint Planning:** Planejamento do trabalho da sprint.
- **Daily Stand-up:** Reunião diária de 15 min para alinhamento.
- **Sprint Review:** Demonstração da entrega para stakeholders.
- **Sprint Retrospective:** Melhorias para a próxima sprint.





✓ Artefatos:

- **Product Backlog:** Lista de funcionalidades priorizadas.
- **Sprint Backlog:** Tarefas selecionadas para a sprint atual.
- **Incremento:** Versão funcional do produto ao final da sprint.

Quando Usar?

- Projetos com requisitos dinâmicos (ex.: startups, produtos digitais).
- Equipes com até 9 pessoas (ideal para times pequenos e multidisciplinares).

Ferramentas Comuns:

- **Jira, Trello, Azure DevOps** (para gestão do backlog e sprints).

Kanban



O que é?

- Método visual de gestão de fluxo de trabalho, focado em **melhoria contínua**.
- Não tem sprints fixas; o trabalho é puxado conforme a capacidade da equipe.

Principais características do Kanban

✓ Quadro Kanban (To Do, Doing, Done):

- Visualiza o fluxo de trabalho em colunas.

✓ Limite de Trabalho em Andamento (WIP – Work in Progress):

- Evita sobrecarga da equipe (ex.: máximo de 3 tarefas em "Doing").

✓ Métricas:

- **Lead Time:** Tempo desde a criação até a conclusão de uma tarefa.
- **Throughput:** Quantidade de tarefas concluídas por período.



Quando Usar?

- Equipes de suporte/manutenção (ex.: DevOps, helpdesk).
- Projetos com demandas imprevisíveis (ex.: correções de bugs).
- Pode ser combinado com Scrum (**Scrumban**).

Ferramentas Comuns:

- Trello, Jira, Kanbanize, Microsoft Planner.

Extreme Programming (XP)

O que é?

- Metodologia ágil focada em **qualidade técnica** e **alta adaptabilidade**.
- Muito usado em desenvolvimento de software com **mudanças frequentes**.



Principais práticas do Extreme Programming (XP)



✓ Engenharia de Software:

- **Pair Programming:** Dois devs trabalhando juntos no mesmo código.
- **Testes Automatizados (TDD – Test-Driven Development):** Desenvolver testes antes do código.
- **Integração Contínua (CI):** Código integrado e testado várias vezes ao dia.

✓ Valores do XP:

- **Comunicação:** Trabalho próximo entre devs e clientes.
- **Simplicidade:** Código mínimo necessário para funcionar.
- **Feedback Rápido:** Ciclos curtos de entrega (1–2 semanas).

Quando usar?

- Projetos com requisitos muito voláteis.
- Equipes pequenas e altamente colaborativas.
- Ambientes onde **qualidade do código** é crítica (ex.: sistemas financeiros).

Ferramentas Comuns:

- GitHub/GitLab (CI/CD), Selenium (testes), Jenkins (automação).

Critério	Scrum	Kanban	XP
Estrutura	Sprints fixas	Fluxo contínuo	Ciclos curtos (1–2 semanas)
Foco	Entrega incremental	Visualização do fluxo	Qualidade do código
Papéis	PO, Scrum Master, Devs	Não tem papéis fixos	Devs e cliente próximo
Melhor Para	Produtos novos	Suporte/manutenção	Projetos com mudanças frequentes
Ferramentas	Jira, Trello	Trello, Kanbanize	GitHub, Jenkins

Como escolher a metodologia certa?

- **Use Scrum se:**
 - Seu projeto tem um roadmap claro, mas requisitos podem mudar.
 - Você precisa de entregas frequentes e previsíveis.
- **Use Kanban se:**
 - Sua equipe lida com demandas imprevisíveis (ex.: incidentes, bugs).
 - Você quer melhorar a eficiência do fluxo de trabalho.
- **Use XP se:**
 - A qualidade do código é crítica.
 - O cliente está envolvido diretamente e os requisitos mudam muito.

Cada metodologia ágil tem seu propósito:

- **Scrum** é ótimo para produtos digitais com evolução constante.
- **Kanban** é ideal para fluxos contínuos (suporte, operações).
- **XP** é focado em excelência técnica e adaptação rápida.

Muitas empresas combinam essas abordagens (ex.: **Scrum + Kanban = Scrumban**). O segredo é adaptar o método à realidade do projeto e da equipe.

OBRIGADO! :)

Fundamentos de Gerenciamento de Projetos

Comparação entre abordagens tradicionais e ágeis



Plataforma completa de aprendizado
contínuo em programação.

#BoostingPeople

rocketseat.com.br

Todos os direitos reservados © Rocketseat S.A.

Aula 3

Comparação entre abordagens
tradicionais e ágeis

A escolha entre **metodologias tradicionais** (como o modelo em cascata) e **metodologias ágeis** (como Scrum, Kanban e XP) depende do tipo de projeto, da flexibilidade necessária e da cultura da organização.

Modelo Tradicional (Cascata/Waterfall)

Principais Características

♦ Sequencial e Linear:

- O projeto avança em fases bem definidas (requisitos → design → desenvolvimento → testes → entrega).
- Cada fase deve ser concluída antes de passar para a próxima.

♦ Documentação Extensa:

- Requisitos são definidos no início e documentados em detalhes.
- Mudanças são difíceis e custosas após o início do projeto.

♦ Controle Rígido de Escopo, Tempo e Custo:

- Planejamento detalhado no início, com cronograma e orçamento fixos.

♦ Entrega Única no Final:

- O cliente só vê o produto completo ao término do projeto.

Vantagens

- ✓ **Previsibilidade:** Cronograma e custos bem definidos desde o início.
- ✓ **Melhor para Projetos com Requisitos Estáveis** (ex.: construção civil, sistemas legados com regras fixas).
- ✓ **Documentação Completa:** Facilita auditorias e compliance.

Desvantagens

- ✗ **Pouca Flexibilidade:** Mudanças são difíceis e caras.
- ✗ **Risco de Entregar um Produto Desatualizado** (em projetos longos, o mercado pode mudar).
- ✗ **Feedback Tardio:** Cliente só vê o produto no final.

Quando usar?

- Projetos com requisitos **claros e imutáveis** (ex.: sistemas bancários legados).
- Ambientes altamente regulados (ex.: saúde, aviação).

Metodologias Ágeis (Scrum, Kanban, XP)

Principais Características

♦ Iterativo e Incremental:

- O trabalho é dividido em ciclos curtos (sprints, iterações).
- Entregas parciais frequentes (ex.: a cada 2 semanas).

♦ Flexibilidade a Mudanças:

- Requisitos podem evoluir ao longo do projeto.
- Priorização contínua do backlog.

♦ Colaboração com o Cliente:

- Stakeholders participam ativamente (ex.: revisões de sprint).

♦ Foco em Valor de Negócio:

- Entrega das funcionalidades mais importantes primeiro.

Vantagens

- ✓ **Adaptabilidade:** Permite ajustes conforme feedback do cliente.
- ✓ **Entrega Rápida de Valor:** MVP (Minimum Viable Product) em poucas semanas.
- ✓ **Melhor Engajamento da Equipe:** Trabalho colaborativo e autonomia.

Desvantagens

- ✗ **Menos Previsibilidade em Prazos Exatos** (depende da evolução do backlog).
- ✗ **Exige Cliente/Stakeholder Envolvido** (nem sempre disponível).
- ✗ **Pode Ter Falhas de Documentação** se não for bem gerenciado.

Quando usar?

- Projetos com **requisitos dinâmicos** (ex.: startups, apps, software SaaS).
- Ambientes de **alta incerteza** (ex.: inovação, transformação digital).

Critério	Tradicional (Cascata)	Ágil (Scrum/Kanban/XP)
Estrutura	Sequencial (fases fixas)	Iterativa (sprints/ciclos)
Flexibilidade	Baixa (mudanças custosas)	Alta (backlog ajustável)
Documentação	Extensa e detalhada	Leve e just-in-time
Entrega	Única, no final	Incremental (entregas frequentes)
Envolvimento do Cliente	Limitado (apenas no início/fim)	Contínuo (revisões de sprint)
Previsibilidade	Alta (cronograma fixo)	Variável (depende do backlog)
Melhor Para	Projetos com requisitos estáveis	Projetos dinâmicos e inovadores

Híbrido: combinando o melhor dos dois mundos

Muitas empresas adotam abordagens **híbridas**, como:

- **Cascata Adaptada:** Fases sequenciais, mas com revisões iterativas.
- **Scrum com Elementos de PMBOK:** Usando Scrum para desenvolvimento, mas controlando custos e riscos como no modelo tradicional.

Exemplo de Uso Híbrido:

Fase Inicial (Cascata):

- Levantamento de requisitos e arquitetura.

Fase de Desenvolvimento (Ágil):

- Sprints para construção e entregas incrementais.

Fase Final (Cascata):

- Testes de integração e implantação controlada.



Qual escolher?

- **Use cascata se:**
 - Seu projeto tem requisitos **claros e imutáveis**.
 - Você precisa de **alto controle de cronograma e custos**.
 - O ambiente exige **documentação robusta** (ex.: compliance).
- **Use ágil se:**
 - Os requisitos podem **mudar com frequência**.
 - Você precisa de **entregas rápidas e feedback contínuo**.
 - A equipe e o cliente estão **dispostos a colaborar ativamente**.
- **Use híbrido se:**
 - Partes do projeto são estáveis (ex.: infraestrutura), mas outras são dinâmicas (ex.: desenvolvimento de features).

A escolha certa depende do **contexto do projeto**, da **maturação da equipe** e da **cultura organizacional**. Muitas empresas hoje adotam frameworks ágeis, mas ainda usam princípios tradicionais para governança e controle.

OBRIGADO! :)

Fundamentos de Gerenciamento de Projetos

Hibridização de metodologias



Plataforma completa de aprendizado
contínuo em programação.

#BoostingPeople

rocketseat.com.br

Todos os direitos reservados © Rocketseat S.A.

Aula 4

Hibridização de metodologias

A hibridização de metodologias representa a evolução natural na gestão de projetos, combinando o melhor dos mundos ágil e tradicional para criar abordagens sob medida.

Fundamentos da Híbridação

A fusão metodológica não é simplesmente "pegar um pouco de cada", mas sim uma integração estratégica:

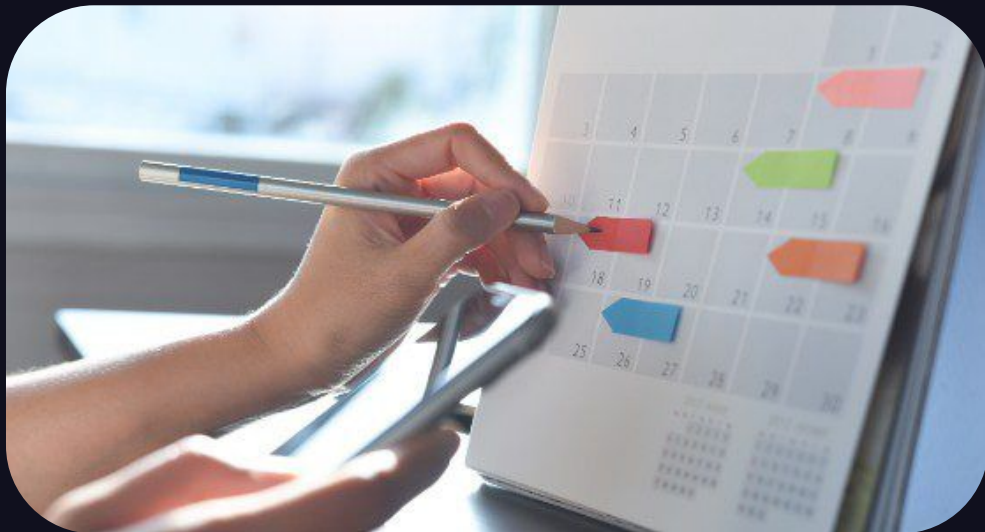
- **Camadas de Governança:** Utilizar PMBOK para macrogestão (portfólio) e Scrum para execução
- **Ciclo de Vida Adaptativo:** Fases iniciais e finais em Waterfall, com núcleo ágil
- **Framework Fluido:** Permite customização contínua conforme maturidade organizacional

Modelos avançados de hibridização

Agile – Waterfall Bridge (Ponte Ágil – Tradicional)

- **Fase 1** (Waterfall): Arquitetura e planejamento estratégico
- **Fase 2** (Ágil): Desenvolvimento iterativo
- **Fase 3** (Híbrida): Testes integrados com automação ágil
- **Fase 4** (Waterfall): Implantação controlada





DevOps Pipeline Híbrido

- Planejamento: OKRs tradicionais
- Desenvolvimento: Scrum + XP
- Entrega: Kanban + ITIL
- Monitoramento: SRE (Site Reliability Engineering)

Técnicas de Implementação

Matriz de Adaptação

Elemento	Tradicional	Ágil	Híbrido
Requisitos	Documento formal	Backlog dinâmico	Baseline + evolução
Métricas	Gantt, EVM	Velocity, CFD	Roadmap + burnup
Controle	Fases gateadas	Inspect & adapt	Checkpoints flex

Ferramentas Integradas

- **Enterprise:** Jira Align + Clarity PPM
- **Midsize:** Azure DevOps + MS Project
- **Startups:** ClickUp + Smartsheet

Casos Reais de Sucesso

Setor Financeiro:

- Camada regulatória: Waterfall
- Desenvolvimento: Scrum banco (Scrum + SAFe)
- Compliance: Automação ágil de testes

Framework de decisão para hibridização

Análise de Variáveis Críticas:

- Grau de incerteza técnica
- Nível de exigência regulatória
- Maturidade da equipe
- Complexidade de integração

Tendências Emergentes

- **Hybrid 4.0:** Incorporação de IA para balanceamento automático de abordagens
- **Quantum Project Management:** Adaptação em tempo real baseada em dados
- **Bio-inspired Agility:** Modelos adaptativos inspirados em sistemas biológicos

Recomendação Prática:

Implemente um MVP de hibridização com:

1. Piloto em 1–2 projetos
2. Métricas comparativas
3. Retrospectiva ampliada
4. Modelo de melhoria contínua

Esta abordagem sofisticada permite criar sistemas de gestão verdadeiramente adaptativos, capazes de responder tanto a exigências de controle quanto a necessidades de inovação acelerada.

OBRIGADO! :)