

Fundamentos de Gerenciamento de Projetos

Ferramentas de planejamento (MS Project, Asana, Trello)



Plataforma completa de aprendizado
contínuo em programação.

#BoostingPeople

rocketseat.com.br

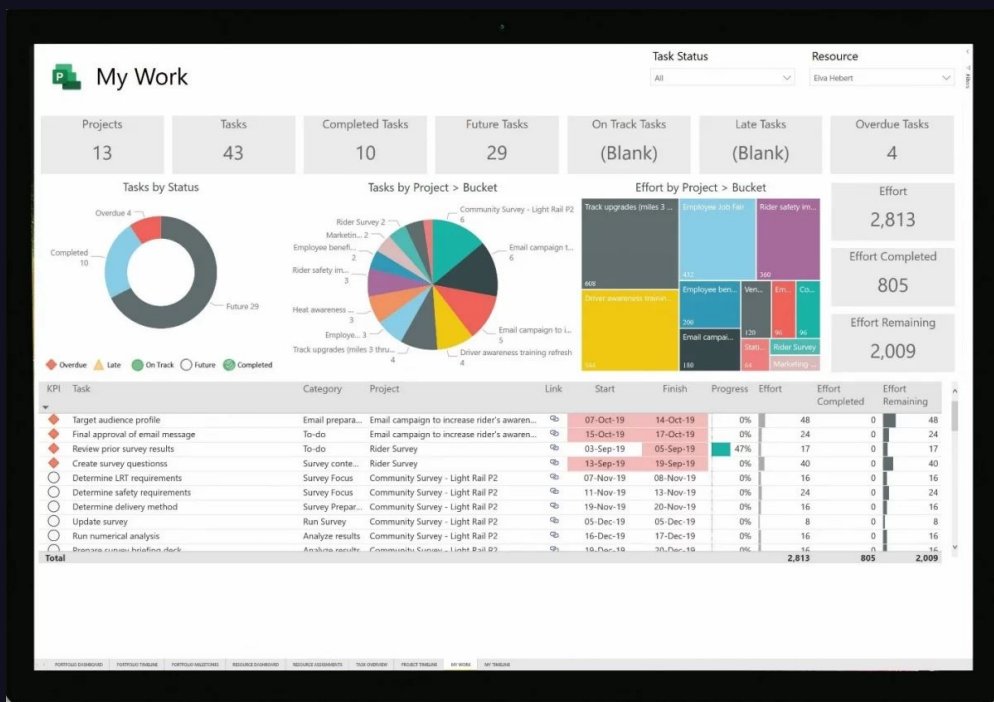
Todos os direitos reservados © Rocketseat S.A.

Aula 1

Ferramentas de planejamento (MS Project, Asana, Trello)

As ferramentas de planejamento são essenciais para organizar tarefas, prazos, recursos e comunicação em projetos. Cada uma tem características distintas e se adapta melhor a diferentes tipos de gestão (tradicional, ágil ou híbrida).

Microsoft Project (MS Project)



Melhor para: Projetos complexos com metodologias tradicionais (PMBOK, Prince2).

Principais Recursos:

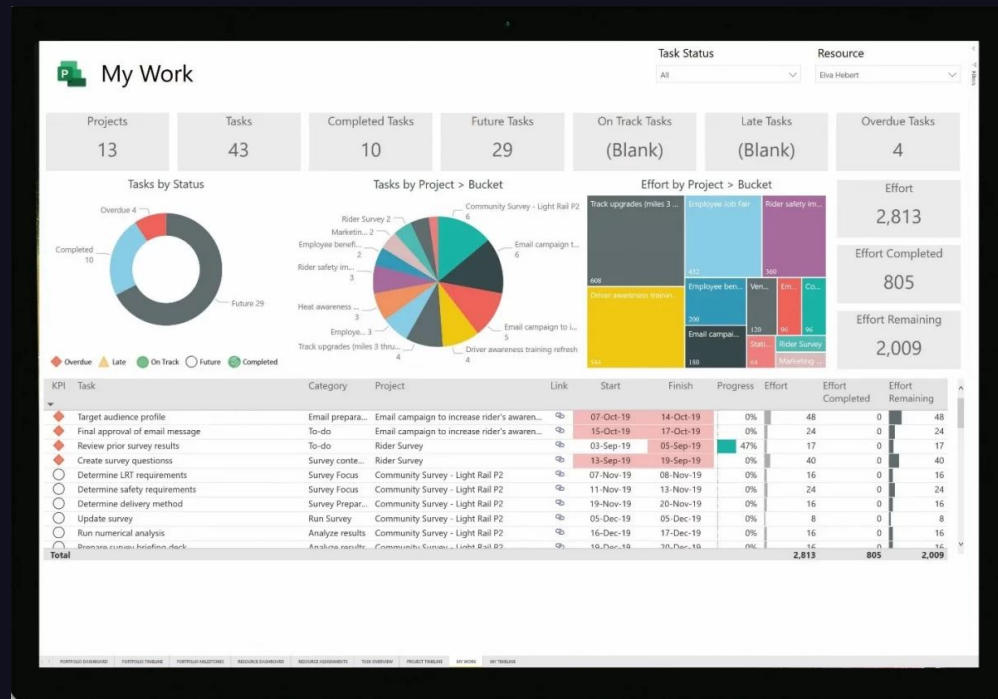
- ✓ **Cronograma Detalhado** (Gráfico de Gantt, caminho crítico).
- ✓ **Gestão de Recursos** (alocação de equipe, custos).
- ✓ **Relatórios Avançados** (EVM, análise de progresso).
- ✓ **Integração com Power BI e Excel.**

Quando Usar?

- Projetos de construção, engenharia ou grandes implantações.
- Quando é necessário controle rigoroso de prazos e custos.

Limitações:

- ❌ Curva de aprendizado mais longa.
- ❌ Preço elevado (versão completa).

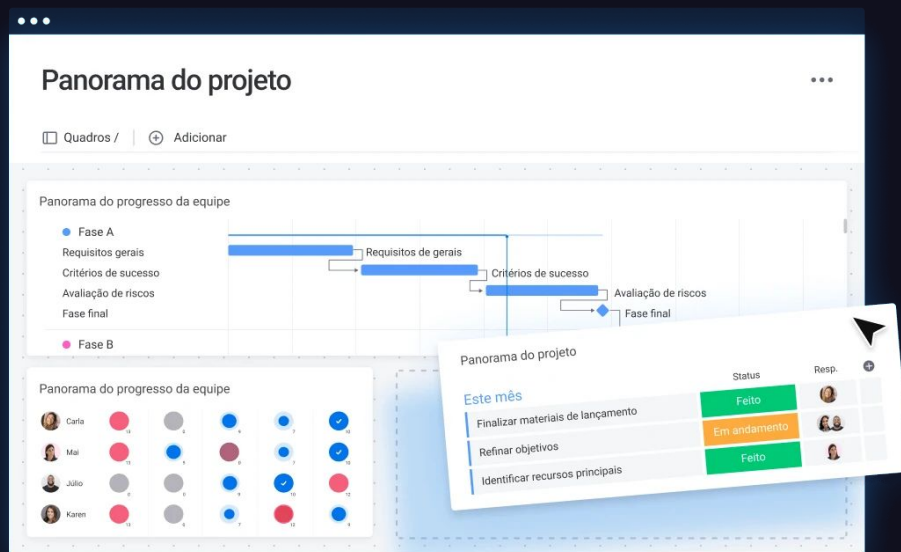


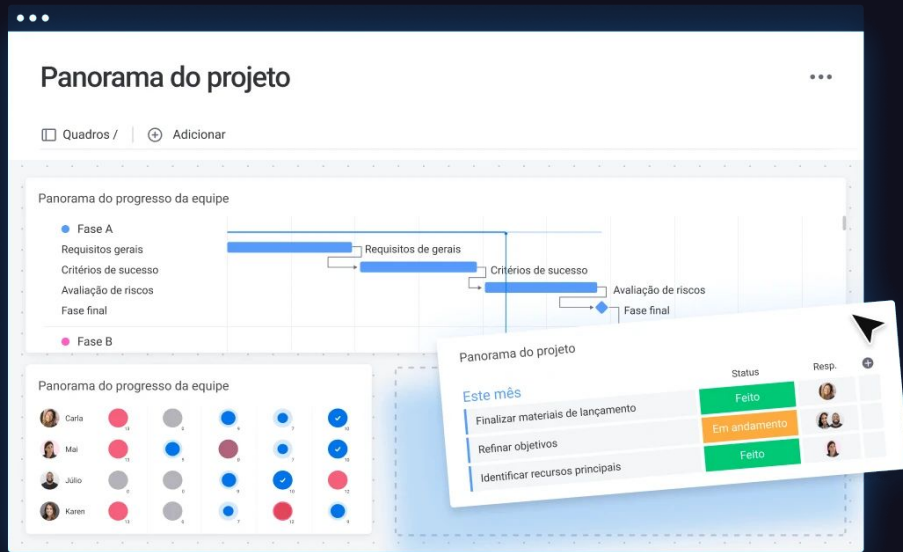
Asana

Melhor para: Equipes ágeis e multidisciplinares.

Principais Recursos:

- ✓ **Quadros Kanban e Listas** (visualização flexível).
- ✓ **Automação de Fluxos** (regras para tarefas recorrentes).
- ✓ **Gestão de Portfólio** (visão geral de múltiplos projetos).
- ✓ **Integrações** (Slack, Google Drive, Zoom).





Quando Usar?

- Projetos de marketing, TI ou produtos digitais.
- Equipes que precisam de colaboração em tempo real.

Limitações:

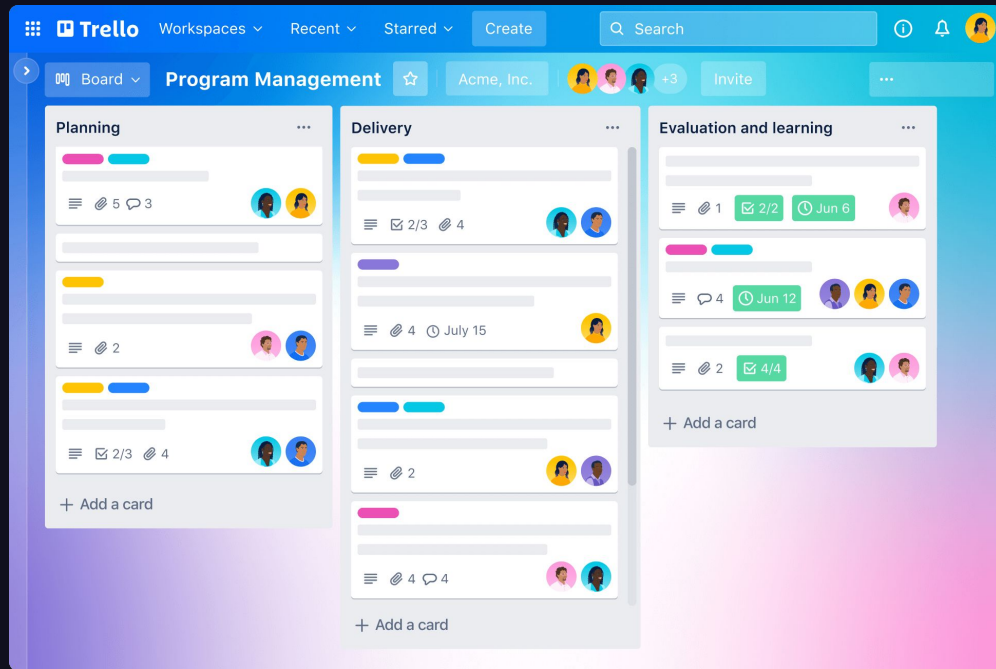
- ✗ Limitação em gerenciamento de custos.
- ✗ Gráfico de Gantt menos robusto que o MS Project.

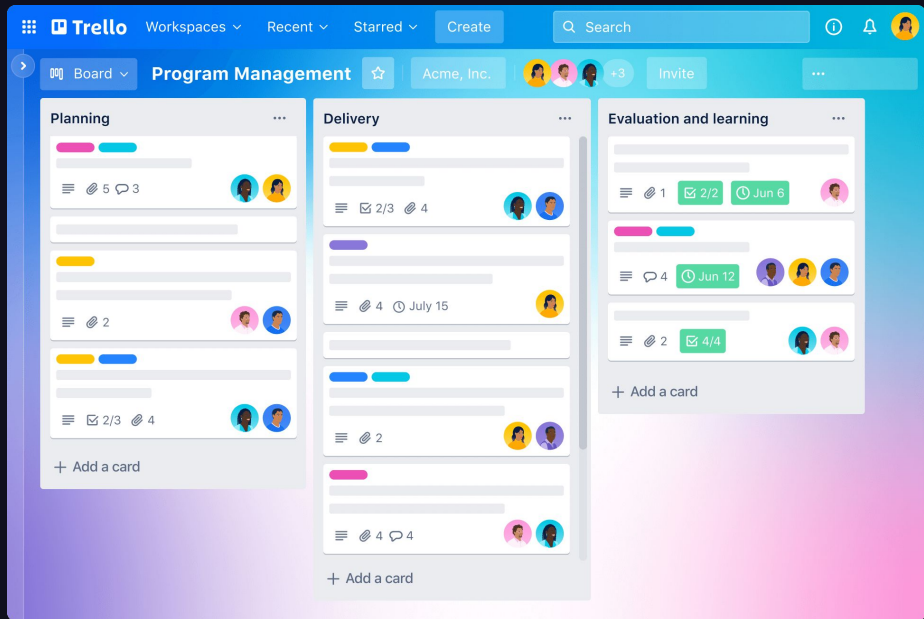
Trello

Melhor para: Projetos simples e metodologias visuais (Kanban).

Principais Recursos:

- ✓ **Quadros e Cartões** (simplicidade de uso).
- ✓ **Power-Ups** (extensões para cronogramas, integrações).
- ✓ **Colaboração em Tempo Real** (comentários, anexos).
- ✓ **Gratuito para uso básico.**





Quando Usar?

- Pequenos projetos ou equipes startups.
- Gerenciamento de tarefas pessoais ou fluxos simples.

Limitações:

- ✗ Poucas opções para projetos complexos.
- ✗ Sem gestão avançada de recursos ou custos.

Comparativo

Ferramenta	Melhor Para	Destaque	Preço
MS Project	Projetos complexos	Gráfico de Gantt detalhado	\$\$\$ (Licença paga)
Asana	Equipes ágeis	Automação e integrações	\$\$ (Plano Premium)
Trello	Projetos simples/Kanban	Simplicidade e gratuidade	\$ (Free ou Power-Ups)

Como escolher a melhor ferramenta?

- Projetos com escopo fixo e muitos recursos → **MS Project**.
- Equipes ágeis e colaboração → **Asana**.
- Tarefas rápidas e visão visual → **Trello**.

Dica: Combine ferramentas! Ex.: Use **Trello** para tarefas diárias e **MS Project** para o cronograma mestre.

Alternativas adicionais

- **ClickUp:** Versátil (combina Gantt, Kanban e listas).
- **Jira:** Ideal para desenvolvimento de software (SCRUM).
- **Smartsheet:** Híbrido entre planilha e gestão de projetos.

A escolha da ferramenta certa **aumenta a produtividade** e reduz falhas de comunicação. Avalie as necessidades do seu projeto antes de decidir!

OBRIGADO! :)

Fundamentos de Gerenciamento de Projetos

Software de gestão ágil (Jira, Azure DevOps)



Plataforma completa de aprendizado
contínuo em programação.

#BoostingPeople

rocketseat.com.br

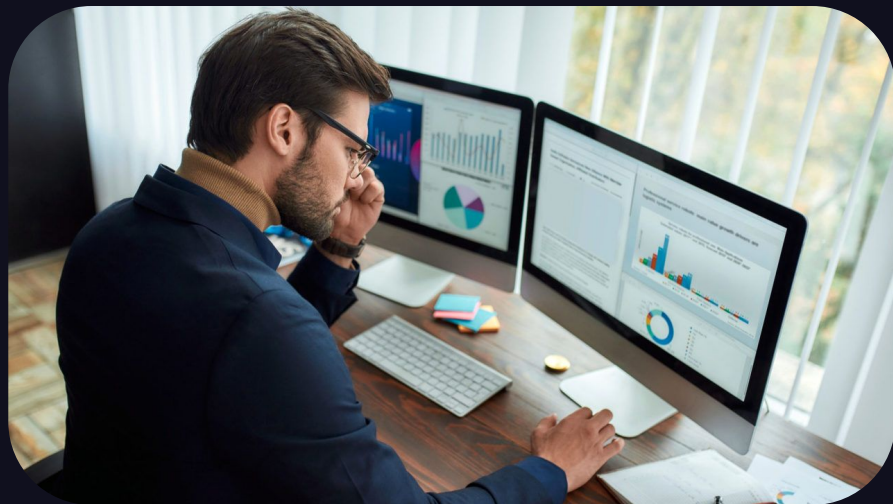
Todos os direitos reservados © Rocketseat S.A.

Aula 2

Software de gestão ágil (Jira, Azure DevOps)

As ferramentas de gestão ágil, como **Jira** e **Azure DevOps**, são essenciais para equipes que seguem metodologias como **Scrum, Kanban ou SAFe**.

Ambas oferecem recursos para **planejamento, execução e monitoramento** de projetos, mas possuem diferenças significativas.

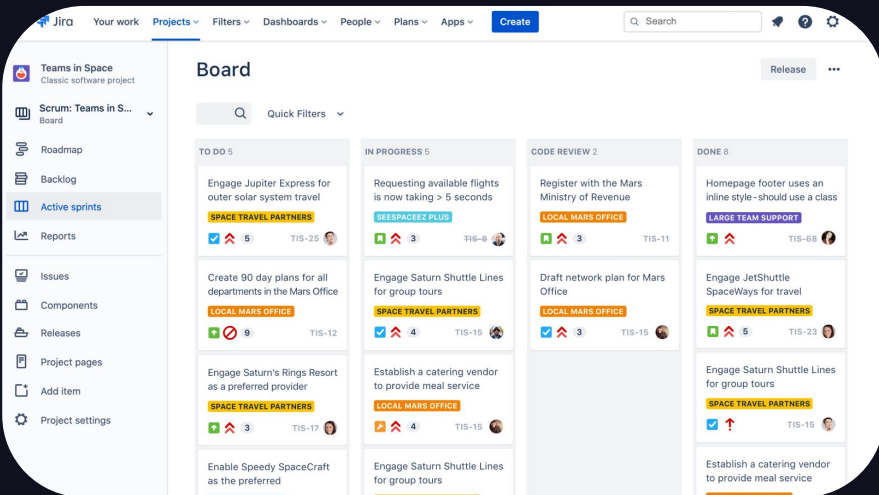


Jira (Atlassian)

Melhor para: equipes de desenvolvimento de software e projetos ágeis.

Principais Recursos:

- ✓ **Quadros Scrum e Kanban** – Visualização flexível de tarefas.
- ✓ **Gestão de Sprints** – Planejamento, daily meetings e retrospectivas.
- ✓ **Roadmaps** – Visão estratégica do produto.
- ✓ **Integrações** – Confluence, Bitbucket, Slack, Trello.
- ✓ **Automação** – Regras personalizadas para fluxos de trabalho.

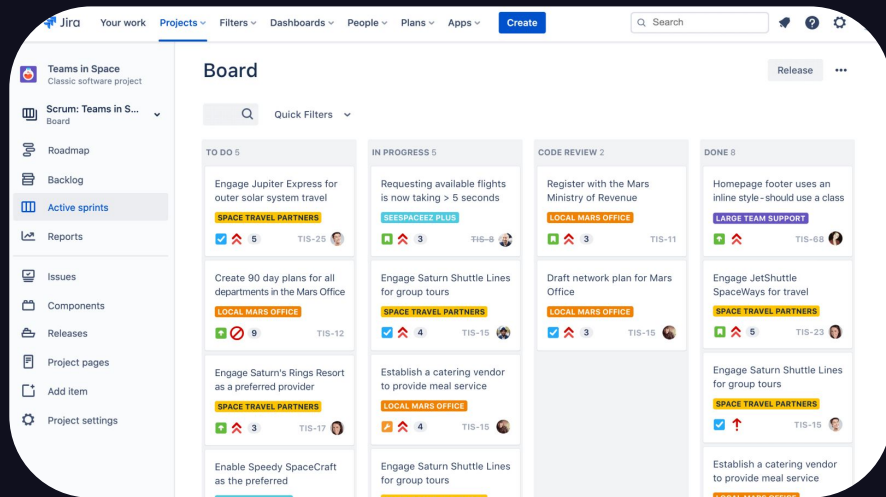


Quando usar?

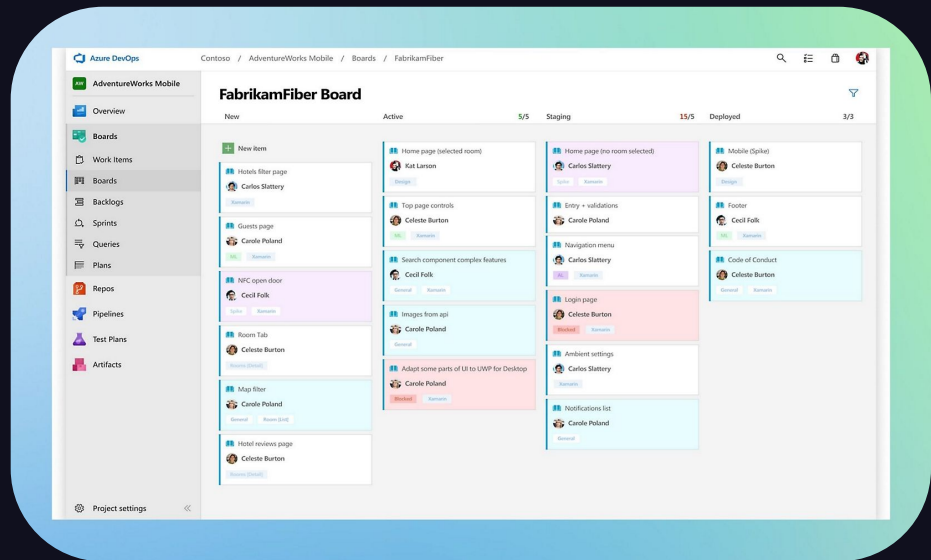
- Equipes de TI, **desenvolvimento e produtos digitais**.
- Projetos que exigem **ritmo contínuo de entregas (Sprints)**.
- Times que usam **Confluence** para documentação.

Limitações:

- ❌ Pode ser complexo para iniciantes.
- ❌ Relatórios avançados exigem plugins (como Jira Advanced Roadmaps).



Azure DevOps (Microsoft)



Melhor para: equipes que usam o ecossistema Microsoft e CI/CD.

Principais Recursos:

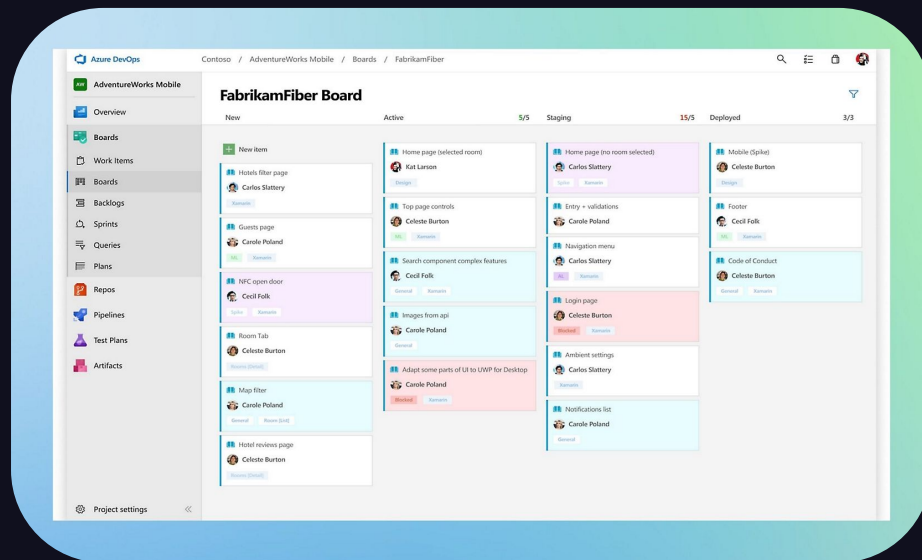
- ✓ **Repositórios Git** – Gestão de código-fonte integrada.
- ✓ **Pipelines de CI/CD** – Automação de builds e deploys.
- ✓ **Quadros Ágeis** – Scrum, Kanban e SAFe.
- ✓ **Test Plans** – Gestão de testes automatizados e manuais.
- ✓ **Integração com Azure, GitHub e Teams.**

Quando usar?

- Projetos que exigem **integração contínua e entrega (CI/CD)**.
- Times que já usam **Microsoft 365 (Teams, Power BI)**.
- Empresas com infraestrutura em **nuvem Azure**.

Limitações:

- ✗ Menos flexível para metodologias não técnicas (ex.: marketing).
- ✗ Interface menos intuitiva que o Jira.



Comparativo

Critério	Jira	Azure DevOps
Foco	Gestão ágil (Scrum/Kanban)	Agile + CI/CD + Repositórios
Integrações	Marketplace extenso	Melhor para stack Microsoft
UI/UX	Customizável, mas complexo	Mais técnico, menos intuitivo
Preço	Caro para grandes times	Mais acessível em planos básicos
CI/CD	Via plugins (ex: Bamboo)	Pipelines nativos
Melhor para	Times ágeis multidisciplinares	Devs em stack Microsoft/Azure

Alternativas populares

- **Trello** (simples, baseado em Kanban)
- **ClickUp** (híbrido, com docs e gestão de tarefas)
- **Asana** (foco em projetos não-técnicos)
- **GitLab** (integra DevOps completo)

Qual escolher?

- **Se sua equipe já usa Microsoft** → Azure DevOps.
- **Se precisa de customização e plugins** → Jira.
- **Se quer CI/CD embutido** → Azure DevOps ou GitLab.
- **Se busca simplicidade** → Trello ou ClickUp.

Ambos são excelentes, mas a escolha depende do fluxo de trabalho e stack tecnológica da sua equipe.

OBRIGADO! :)

Fundamentos de Gerenciamento de Projetos

Uso de dashboards e relatórios para tomada de decisão



Plataforma completa de aprendizado
contínuo em programação.

#BoostingPeople

rocketseat.com.br

Todos os direitos reservados © Rocketseat S.A.

Aula 3

Uso de dashboards e relatórios para tomada de decisão

Os **dashboards e relatórios** em ferramentas de gestão ágil (como Jira e Azure DevOps) são fundamentais para visualizar o progresso do time, identificar gargalos e tomar decisões baseadas em dados.



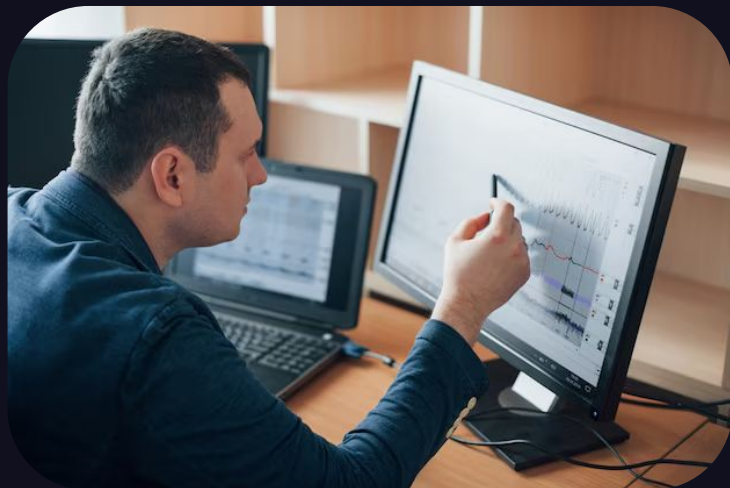
Dashboards em Ferramentas Ágeis

São painéis visuais que consolidam métricas, gráficos e indicadores em tempo real.

Jira

- **Tipos de widgets:**
 - **Burndown Chart:** Mostra o progresso da sprint (tarefas concluídas vs. restantes).
 - **Velocity Chart:** Média de pontos entregues por sprint (para planejamento futuro).
 - **Cumulative Flow Diagram (CFD):** Identifica gargalos (ex.: muitas tarefas em "Em andamento").
 - **Filter Results:** Lista personalizada de tarefas (ex.: bugs críticos).
- **Como usar para decisões:**
 - Se o **Burndown** estiver "achatado", o time pode estar bloqueado.
 - Se o **Velocity** variar muito, as sprints estão mal planejadas.





Azure DevOps

- **Widgets úteis:**
 - **Sprint Burndown:** Similar ao Jira.
 - **Cycle Time:** Tempo médio para concluir uma tarefa (otimiza fluxo).
 - **Lead Time:** Tempo desde a criação até a entrega (mede eficiência).
 - **Test Analytics:** Taxa de sucesso/falha de testes automatizados.
- **Como usar para decisões:**
 - **Cycle Time alto** → Processo precisa de ajustes (ex.: revisões longas).
 - **Lead Time crescente** → Possível sobrecarga do time.
 -

Relatórios ágeis para tomada de decisão

Relatório	O que mede?	Decisão Associada
Burndown Chart	Progresso diário da sprint	Ajustar escopo ou alocar mais recursos.
Velocity Report	Capacidade do time por sprint	Planejar sprints realistas.
CFD	Fluxo de trabalho (Kanban)	Reduzir WIP (trabalho em progresso).
Control Chart	Tempo médio por etapa (ex.: "Code Review")	Melhorar etapas lentas.
Bug Trend	Frequência de bugs reportados	Investigar causas raízes (ex.: testes frágeis).

Exemplos práticos de tomada de decisão

Cenário 1: Atrasos na Sprint

- **Dado:** Burndown Chart mostra tarefas não concluídas.
- **Ação:**
 - Verificar se há dependências externas bloqueando o time.
 - Reduzir escopo ou adicionar membros em tarefas críticas.

Cenário 2: Baixa Velocidade

- **Dado:** Velocity Report com queda por 2 sprints.
- **Ação:**
 - Revisar estimativas (pontos muito altos?).
 - Investigar motivos (ex.: falta de capacitação, interrupções).

Cenário 3: Gargalo em "Testes"

- **Dado:** CFD mostra acúmulo de tarefas em "Testing".
- **Ação:**
 - Automatizar testes ou realocar recursos para a área.

Melhores Práticas

- ✓ **Padronize métricas:** Defina quais KPIs importam (ex.: Velocity, Cycle Time).
- ✓ **Compartilhe com o time:** Dashboards devem ser visíveis a todos.
- ✓ **Combine dados qualitativos:** Reuniões de retrospectiva + dados.
- ✓ **Use filtros:** Ex.: separar bugs por criticidade.

Controle das Aquisições

- **Objetivo:** Garantir que o fornecedor cumpra o acordado.
- **Ações:**
 - Monitorar entregas (qualidade e prazos).
 - Gerenciar alterações contratuais.
 - Resolver conflitos (ex.: atrasos, não conformidades).

Ferramentas Adicionais para Análise

- **Power BI** (para integrar dados do Jira/Azure DevOps em visualizações avançadas).
- **Grafana** (monitoramento contínuo de métricas de DevOps).
- **Custom Reports** (via Jira Query Language – JQL).

Dashboards e relatórios transformam dados brutos em **insights acionáveis**. O segredo é:

1. **Monitorar** os indicadores certos.
2. **Agir** rapidamente com base nos dados.
3. **Ajustar** processos continuamente.

OBRIGADO! :)

Fundamentos de Gerenciamento de Projetos

Automação de processos em gerenciamento de projetos



Plataforma completa de aprendizado
contínuo em programação.

#BoostingPeople

rocketseat.com.br

Todos os direitos reservados © Rocketseat S.A.

Aula 4

Automação de processos em
gerenciamento de projetos



Automação de processos em gerenciamento de projetos é uma abordagem que visa otimizar, padronizar e agilizar as atividades relacionadas à gestão de projetos por meio de ferramentas tecnológicas.

Essa prática reduz o trabalho manual, diminui erros, melhora a comunicação entre equipes e aumenta a eficiência na entrega dos projetos.

Processos que podem ser automatizados

Planejamento e Cronograma:

- Uso de ferramentas como **Microsoft Project**, **ClickUp** ou **Asana** para criar cronogramas automatizados com dependências e alertas.
- Integração com calendários (Google Calendar, Outlook) para atualizações automáticas.

Gestão de Tarefas e Atribuições:

- Ferramentas como **Trello**, **Jira** ou **Monday.com** permitem automatizar fluxos de trabalho (ex.: mover tarefas entre colunas quando concluídas).
- Notificações automáticas para prazos e responsáveis.

Comunicação e Relatórios:

- **Slack** ou **Microsoft Teams** integrados a bots para notificações de status.
- Geração automática de relatórios com **Power BI**, **Tableau** ou **Google Data Studio**.

Gestão de Riscos e Issues:

- Sistemas como **RiskWatch** ou **Jira** podem alertar sobre riscos baseados em métricas pré-definidas.

Aprovações e Fluxos de Trabalho:

- Plataformas como **Zapier** ou **Microsoft Power Automate** para automatizar aprovações via e-mail ou chat.

Gestão de Documentos:

- **Google Drive, SharePoint ou Notion** com versionamento automático e templates pré-definidos.

Ferramentas populares para automação

Ferramenta	Aplicação
Microsoft Power Automate	Automação de fluxos entre apps (Excel, Teams, ERP).
Zapier	Integração entre +3.000 apps (Gmail, Slack, Trello).
Jira + Automation	Gestão ágil com triggers automáticos.
ClickUp	Automação de tarefas e status.
Smartsheet	Planilhas inteligentes com gatilhos.

Benefícios da automação

- **Redução de tarefas repetitivas** (ex.: envio de relatórios).
- **Maior visibilidade** do projeto em tempo real.
- **Menos erros manuais** em cronogramas e orçamentos.
- **Comunicação mais eficiente** com partes interessadas.

Desafios

- **Custo inicial** de implementação.
- **Resistência da equipe** à mudança.
- **Necessidade de personalização** para processos específicos.

Primeiros Passos

1. **Mapeie processos repetitivos** (ex.: aprovações, atualizações de status).
2. **Escolha ferramentas alinhadas** ao seu orçamento e stack tecnológico.
3. **Teste em pequena escala** antes de expandir.
4. **Treine a equipe** para adoção eficaz.

OBRIGADO! :)