

Planejamento do DEV

Planejamento do DEV

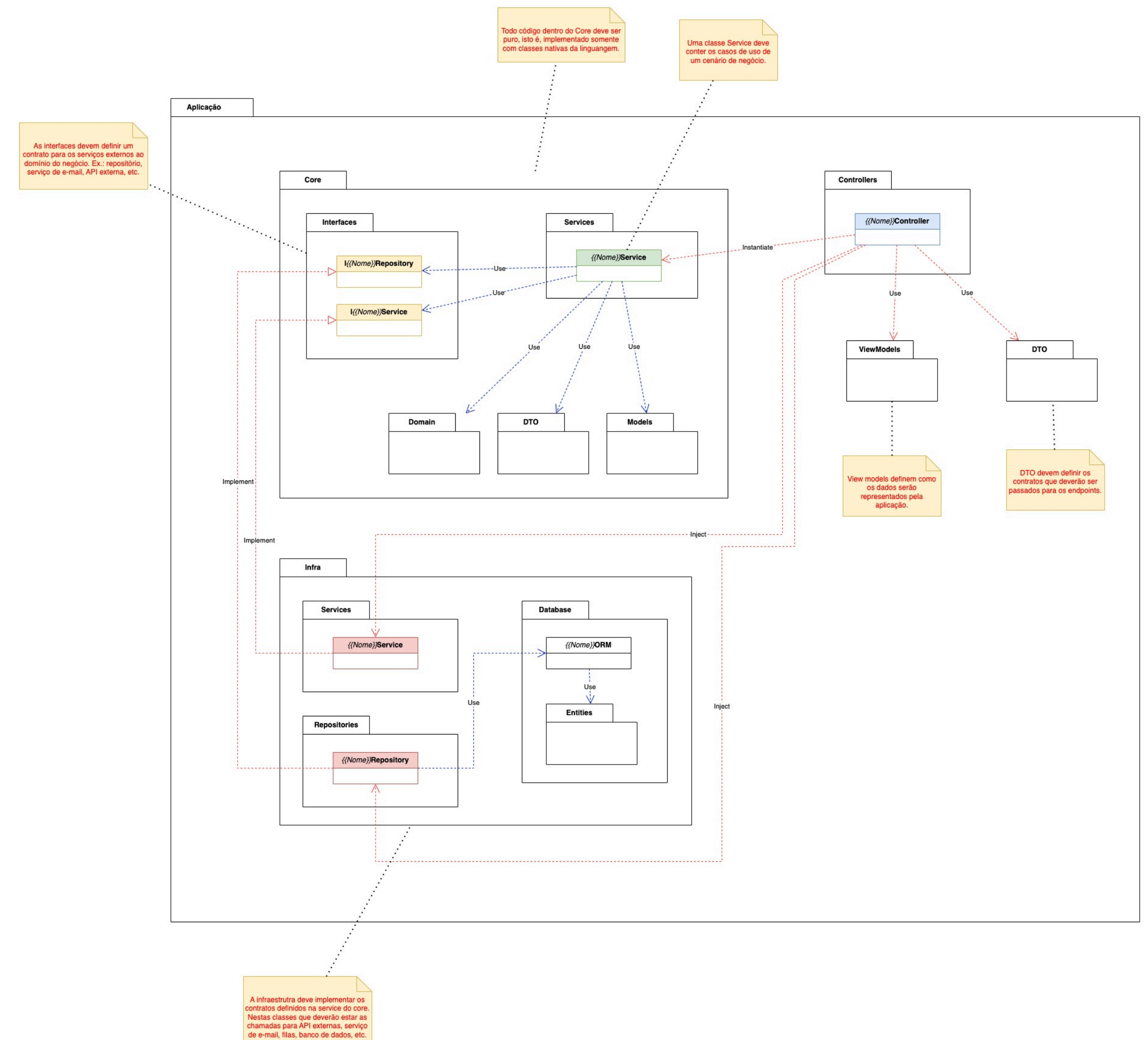
Padronização da arquitetura

- Deverá ser definida a arquitetura no início do projeto
- Deverá ter a **definição de pronto** para **qualidade de código ANTES** de iniciar o desenvolvimento
- Desenho **macro** da solução já deverá ser fornecida e **estar claro** para todo o time
- A arquitetura deve permitir a entrega do projeto em **incrementos funcionais** para que os **milestones** sejam alcançados

Planejamento do DEV

Padronização da arquitetura

- O desenho deve ser atualizado ao longo do desenvolvimento conforme se identificam as necessidades de mudança



Planejamento do DEV

Análise e entendimento da história

- O dev precisa minimamente se planejar antes de escrever código. Caso a história não disponibilize, deverá ser criado pelo menos:
 - **Diagrama de caso de uso:** ajuda no entendimento das regras e as interações com atores (aplicativos, arquétipos, api externas, etc)
 - **Diagrama de classes** para atender a história. Deverá complementar o digrama de classes já fornecido no início do projeto
 - Se a regra se mostrar complexa, lance mão da criação de **diagramas de interação** ou de **diagrama de sequência**. Esse dará respaldo para refinar a história junto ao product owner ou analista responsável pela definição das regras de negócio
 - Essa análise guiará o planejamento de esforço/prazo para implementação

Planejamento do DEV

Definição de datas e milestones

- Após realizar o setup do projeto dá para definir pelo menos **3 milestones** para o desenvolvimento da história:
 1. Definição dos contratos identificados:

Projeto rodando com a(s) controller(s) anotada(s) para gerar o Swagger
 2. Repositories definidas e script para criação/alteração do banco de dados
 3. Finalização da história e realização de teste integrado para liberar como concluída

Planejamento do DEV

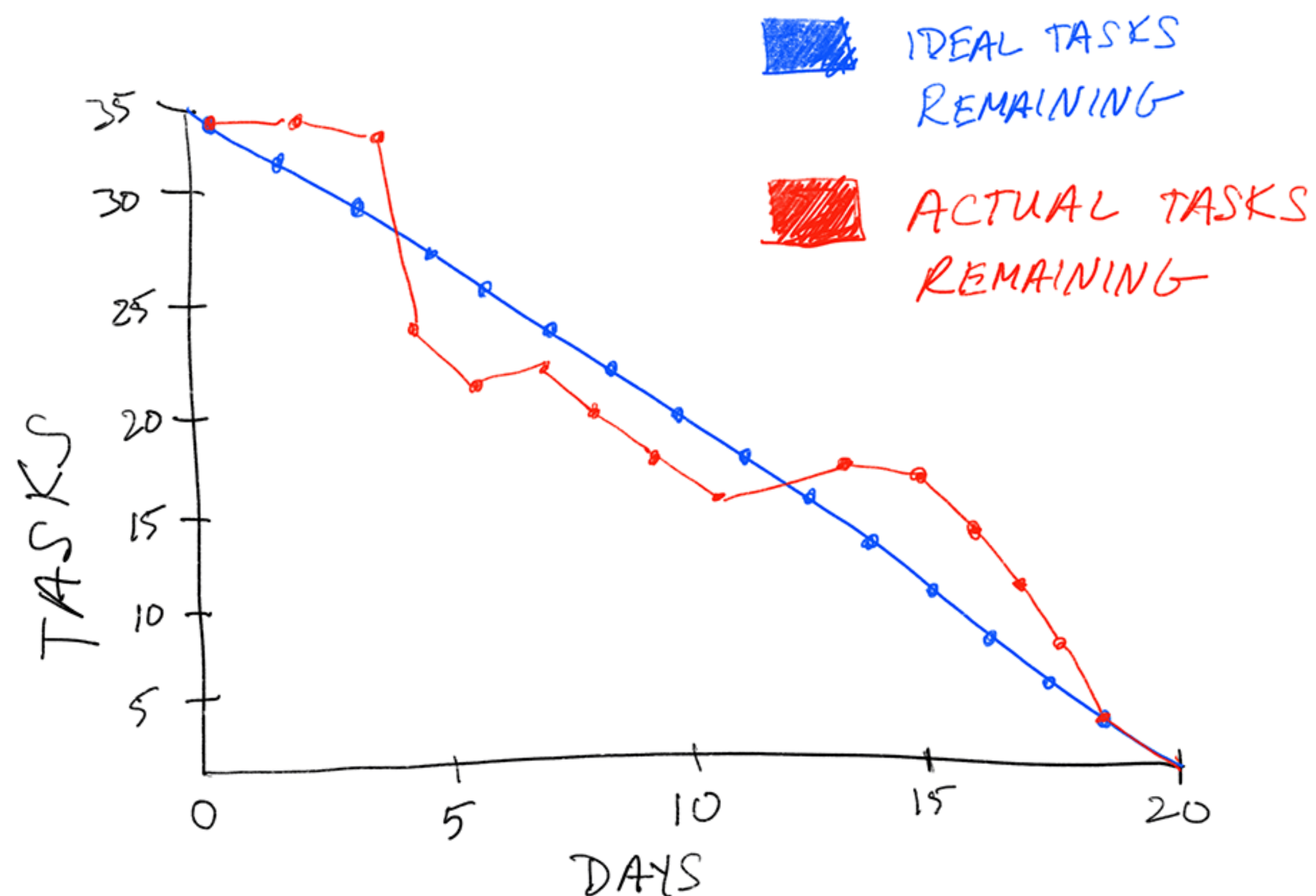
Acompanhamento diário

- Os **milestones** ajudarão **planejar** início de atividades **dependentes** dessa história junto ao time ou times externo
- O **planejamento** deverá ser entregue ao gerente do projeto ou scrum master para que esse possa **acompanhar** e integrar ao **planejamento do projeto** todo
- Se a ferramenta de board/kanban não tiver essa funcionalidade, não houver acesso ao MS Project e não quiser fazer planilha, **há alternativa** online: <https://www.onlinegantt.com>
- Para que seja possível o **acompanhamento**, deve-se **diariamente** reportar um percentual de progresso. Utilize a daily para reportar ou atualize o card na ferramenta de board/kanban (Jira, Trello, etc)

Controle da Gestão

Controle da Gestão

Gráfico de Burndown




- Um gráfico de burndown pode ser gerado à partir do progresso reportado pelo time
- Se a ferramenta utilizada para controlar o board/kanban não disponibilizar essa funcionalidade, pode ser usar templates como o disponibilizado pelo PMI [1]

Controle da Gestão

Mapeamento de riscos

- No **início** de cada ciclo de desenvolvimento deve-se tentar **mapear riscos e acompanhá-los** para que não afetem o atingimento da meta do ciclo
- Se a ferramenta utilizada para controlar o board/kanban não disponibilizar essa funcionalidade, pode ser usar templates como o disponibilizado pelo PMI [2]


kickoff.pmi.org

Accept	Green
Transfer	Yellow
Mitigate	Orange
Avoid	Red

elihood

Risk Worksheet

Project Name:
Project Manager:

	Potential Consequences				
	Not Significant	Minor	Moderate	Major	Severe
Almost Certain	Medium	High	Very High	Very High	Very High
Likely	Medium	High	High	Very High	Very High
Possible	Low	Medium	High	High	Very High
Unlikely	Low	Low	Medium	Medium	High
Rare	Low	Low	Low	Low	Medium

How To Use This Worksheet:

1. Enter each risk your team has identified in the Risk column.

Risk	Probability /Impact	Risk Response	Risk Response Plan

Referências

[1] Gráfico de burndown: <https://kickoff.pmi.org/app/getting-the-work-done/measure-project-success/3>

[2] Matriz de riscos: <https://kickoff.pmi.org/app/planning-your-project-work/know-your-risks-and-create-mitigation-plans/3>