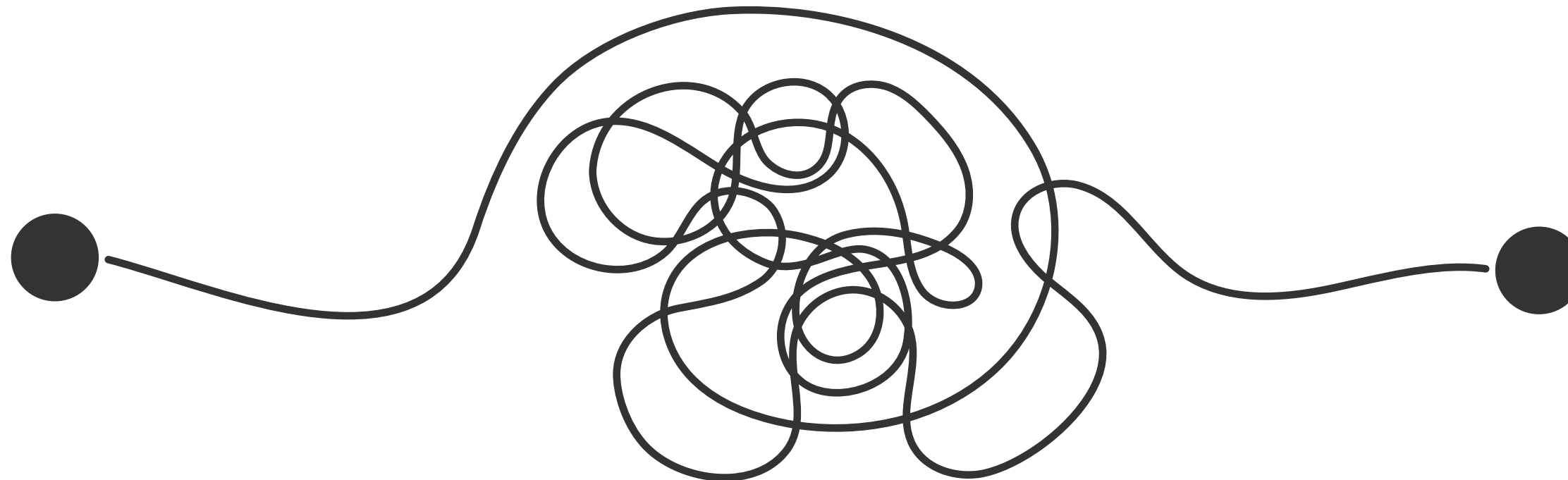


METODOLOGIA ÁGIL

	Introdução
	O que tínhamos antes do ágil
	Metodologia ágil
	SCRUM
	Kanban
	Ferramentas



— INTRODUÇÃO

Engenharia de Software

O que tentamos resolver?

Porque é tão complexo?

Demora tanto assim pra sair do outro lado?

Custa muito caro?

Mas não é só perguntar e fazer?

Triângulo de projetos:

custo x tempo x escopo


O QUE TÍNHAMOS ANTES DO ÁGIL —

PMBOK - Project Management Body of Knowledge
Conhecimento em Gerenciamento de Projetos

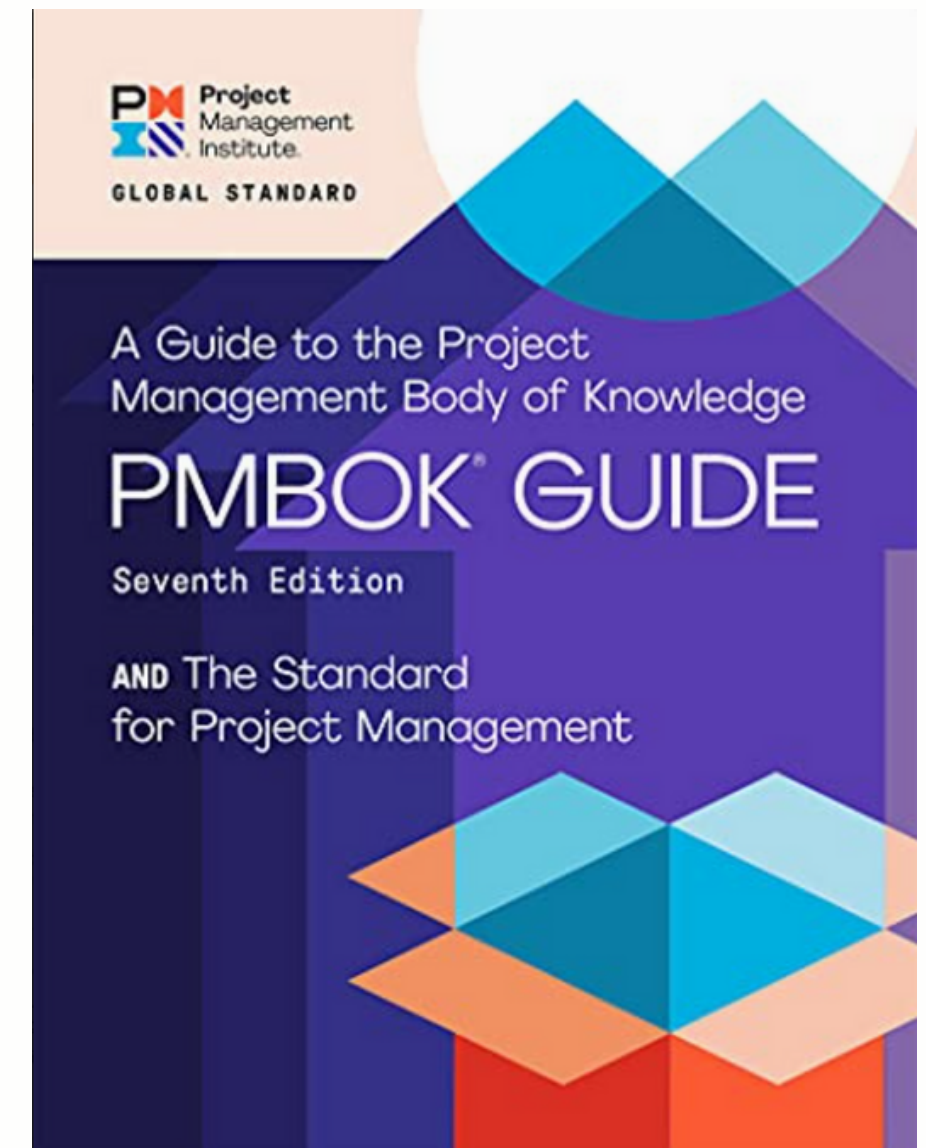
756 Páginas de puro aprendizado
49 Processos

E a bagatela de

Kindle
R\$ 157,90

Capa Comum
R\$ 404,60 

É moroso, trabalhoso, envolve muito conhecimento teórico e muitas pessoas pra fazer acontecer



O QUE TÍNHAMOS ANTES DO ÁGIL —

— MODELOS DE CICLO DE VIDA DO SOFTWARE

MODELOS DE CICLO DE VIDA DO SOFTWARE

WATERFALL

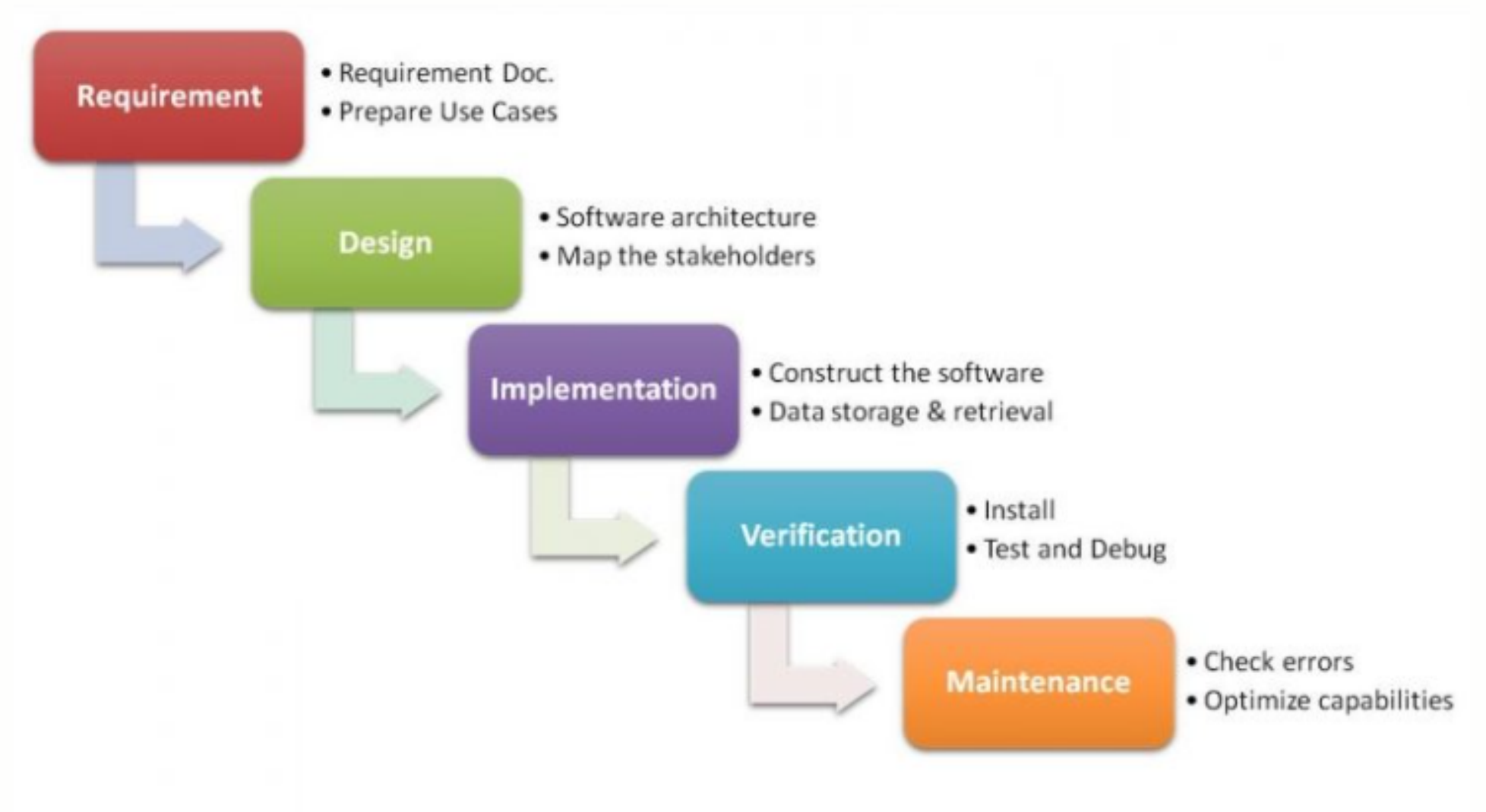
Pouco dinâmico

Entregáveis para finalização de cada etapa

Alto custo

Entrega final

Pelo menos é simples!

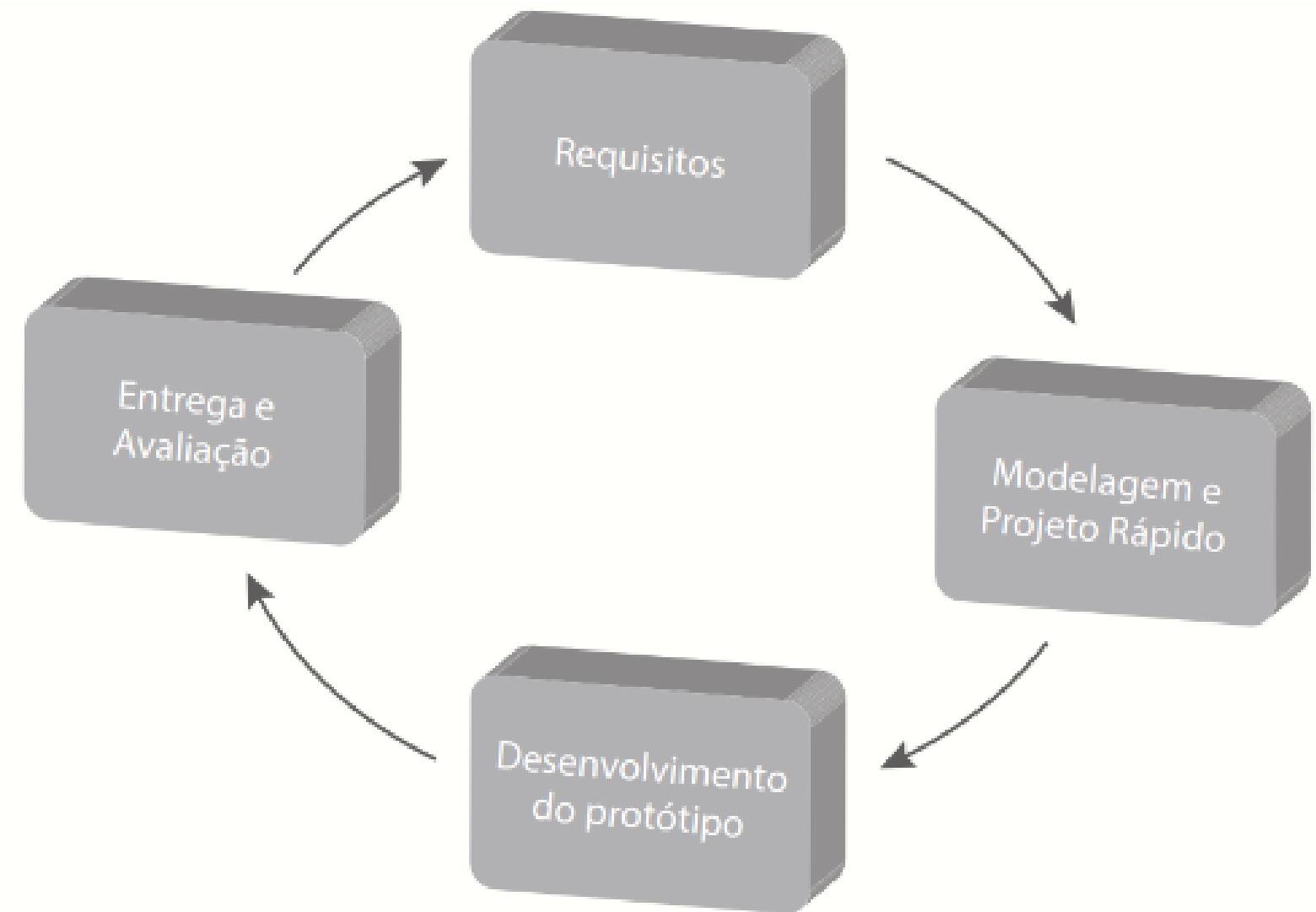


MODELOS DE CICLO DE VIDA DO SOFTWARE

PROTOTYPING

Não leva em consideração requisitos não funcionais
Zero documentação
Peso para manutenção

É um modelo mais real!



MODELOS DE CICLO DE VIDA DO SOFTWARE

ITERATIVE AND INCREMENTAL

Mini cascata, também conhecido como
Pequenas entregas que compõem o todo
Trás ainda os problemas, porém de forma bem mais amena

Estamos no mais ágil possível!



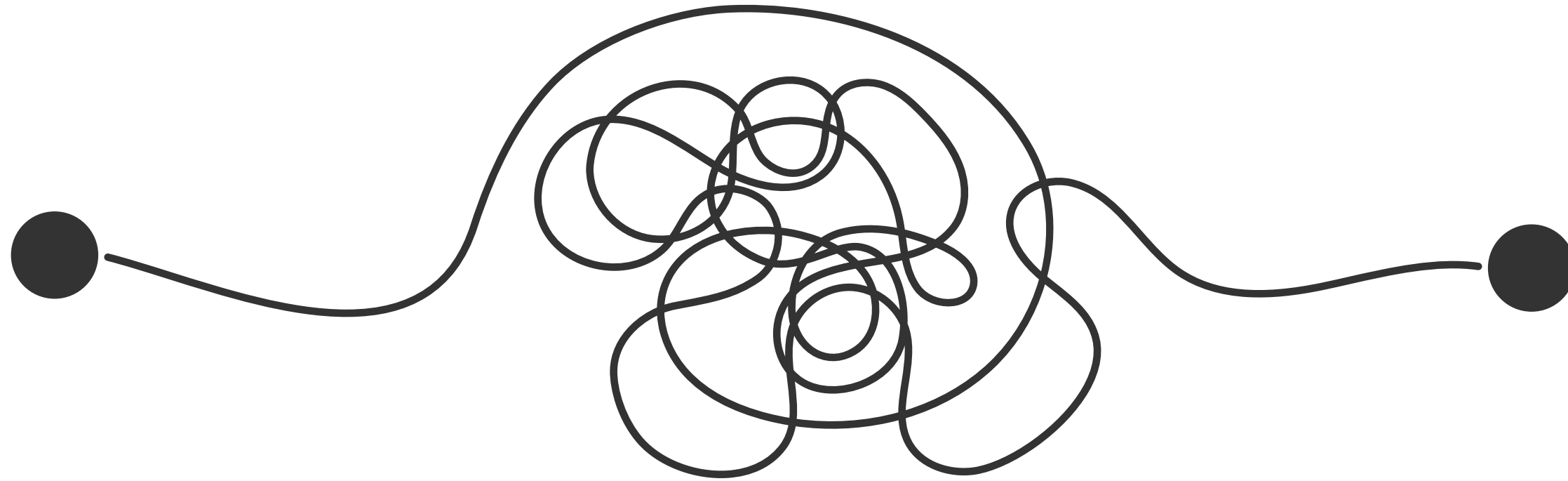
MODELOS DE CICLO DE VIDA DO SOFTWARE

SPIRAL

Analise de risco é acrescentada ao já conhecido
Evolução de entregas antigas
Modelo complexo

Talvez uma das melhores formas!





— METODOLOGIA ÁGIL

Para resolver o problema de não entregas

Para ser ágil e não rápido

Documentação, validação e processos

Simplicidade

Jeff Sutherland

Ken Schwaber



Valores ágeis

Indivíduos e interações	mais que	processos e ferramentas
Software em funcionamento	mais que	documentação abrangente
Colaboração com o cliente	mais que	negociação de contratos
Responder a mudanças	mais que	seguir um plano

Princípios ágeis

1. “Nossa maior prioridade é satisfazer o cliente através da entrega contínua e adiantada de software com valor agregado”
2. “Mudanças nos requisitos são bem-vindas, mesmo tardiamente no desenvolvimento. Processos ágeis tiram vantagem das mudanças visando vantagem competitiva para o cliente”
3. “Entregar frequentemente software funcionando, de poucas semanas a poucos meses, com preferência à menor escala de tempo”
4. “Pessoas de negócio e desenvolvedores devem trabalhar em conjunto diariamente por todo o projeto”
5. “Construa projetos em torno de indivíduos motivados. Dê a eles o ambiente e o suporte necessário e confie neles para fazer o trabalho”
6. “O método mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento é por meio de conversa face a face”
7. “Software funcionando é a medida primária de progresso”
8. “Os processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente”
9. “Contínua atenção à excelência técnica e bom design aumenta a agilidade”
10. “Simplicidade – a arte de maximizar a quantidade de trabalho não realizado – é essencial”
11. “As melhores arquiteturas, requisitos e designs emergem de equipes auto-organizáveis”
12. “Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e então refina e ajusta seu comportamento de acordo.”

SCRUM_

Pilares do Scrum

Conhecimento baseado no empirismo, ou seja, baseado em evidências históricas do que deu certo.

O empirismo têm 3 pilares:

- Transparência -> o que é importante tem que ser de conhecimento geral.
- Inspeção -> Inspecionar constantemente o andamento sem atrapalhar o trabalho.
- Adaptação -> quando for identificado na inspeção algum desvio, deve-se adaptar para corrigir o processo. Tudo isso é o quanto antes, pra evitar gastos.

SCRUM_

Valores do Scrum

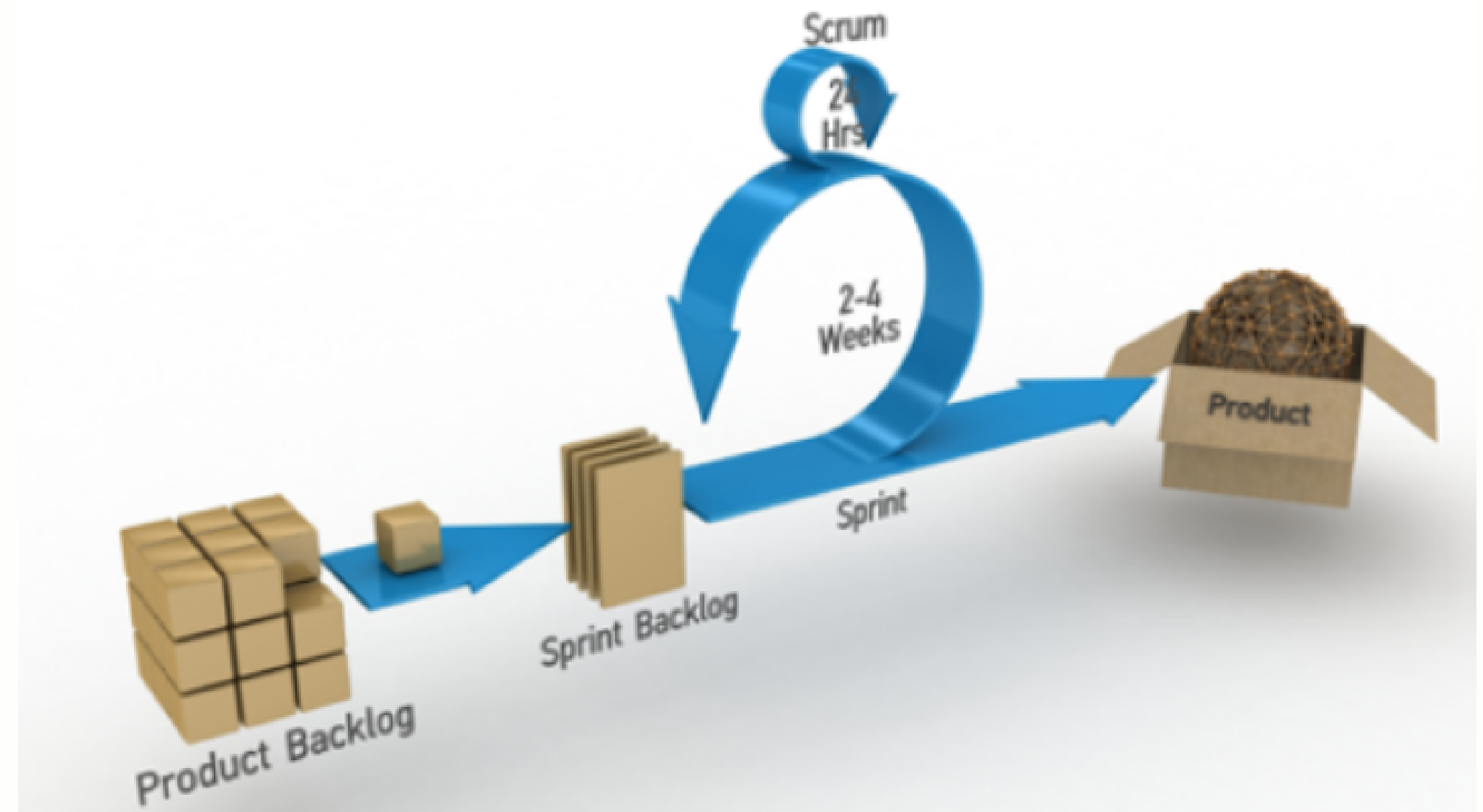
Equipes auto organizadas para maximizar a comunicação e reduzir a supervisão a partir dos eventos e papéis seguidos corretamente. O SCRUM é colaborativo, tão logo todos passam a desenvolver e praticar os valores de forma colaborativa, desenvolvendo naturalmente em todos do time, baseado nos pilares:

- Coragem -> enfrenar problemas e fazer o certo.
- Foco -> focar na sprint e nos objetivos do time.
- Comprometimento -> para alcançar os objetivos.
- Respeito -> respeitar uns aos outros, todos são capazes.
- Abertura -> estar aberto a todo o trabalho e dispostos a fazer acontecer.

VISÃO GERAL

Modelo Interativo Incremental
Entrega contínua e evolutiva

Juntou o que tinha de melhor!



EVENTOS

Sprint Planning

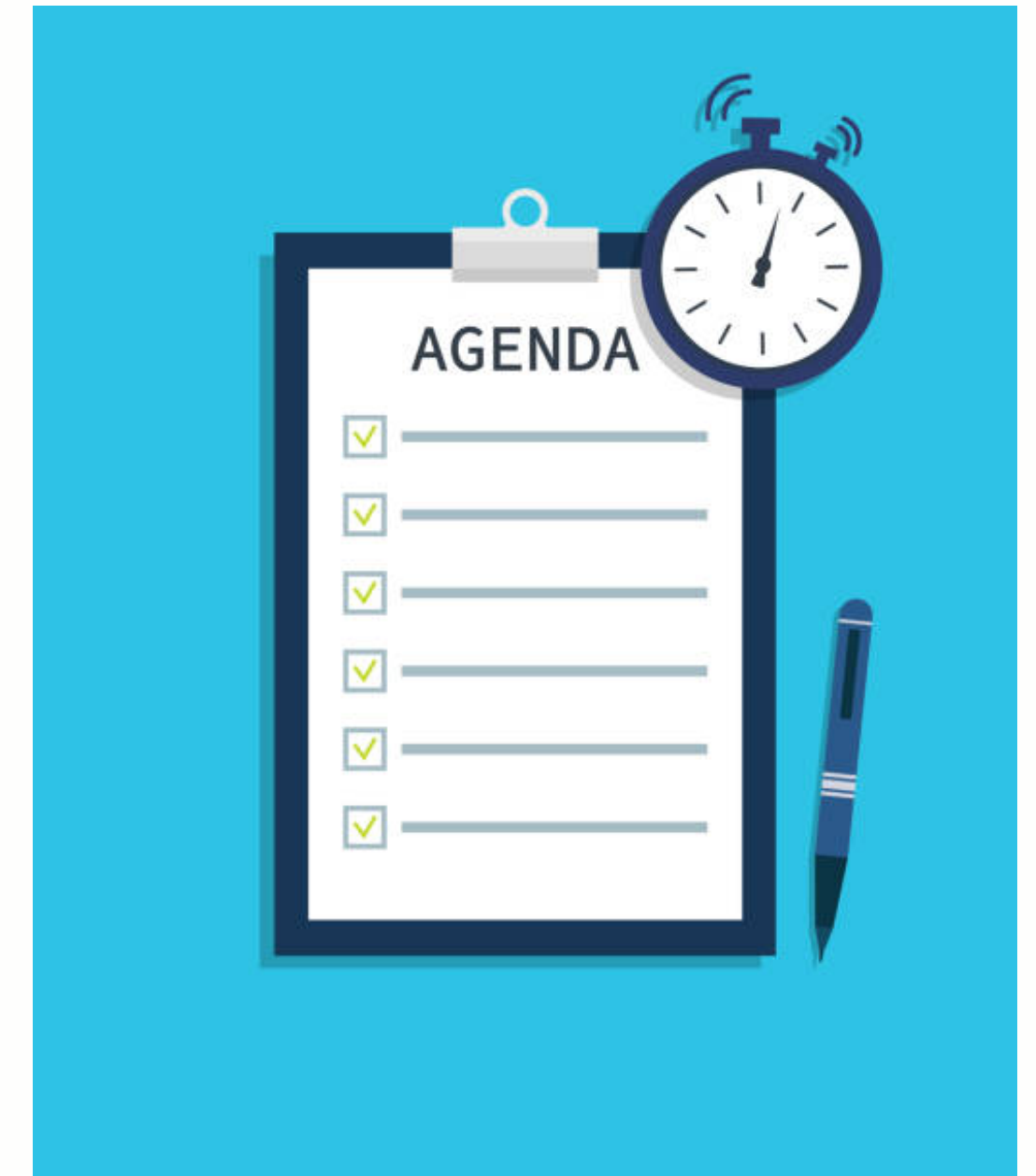
Daily

Sprint Review

Retrospective

Task Refine/Tech Refine

Reunião que poderia ser e-mail também é ser ágil!



ARTEFATOS

Product Backlog

Burndown

Sprint Backlog

Sprint

Documentação não tá, mas poderia!



SCRUM

PAPÉIS

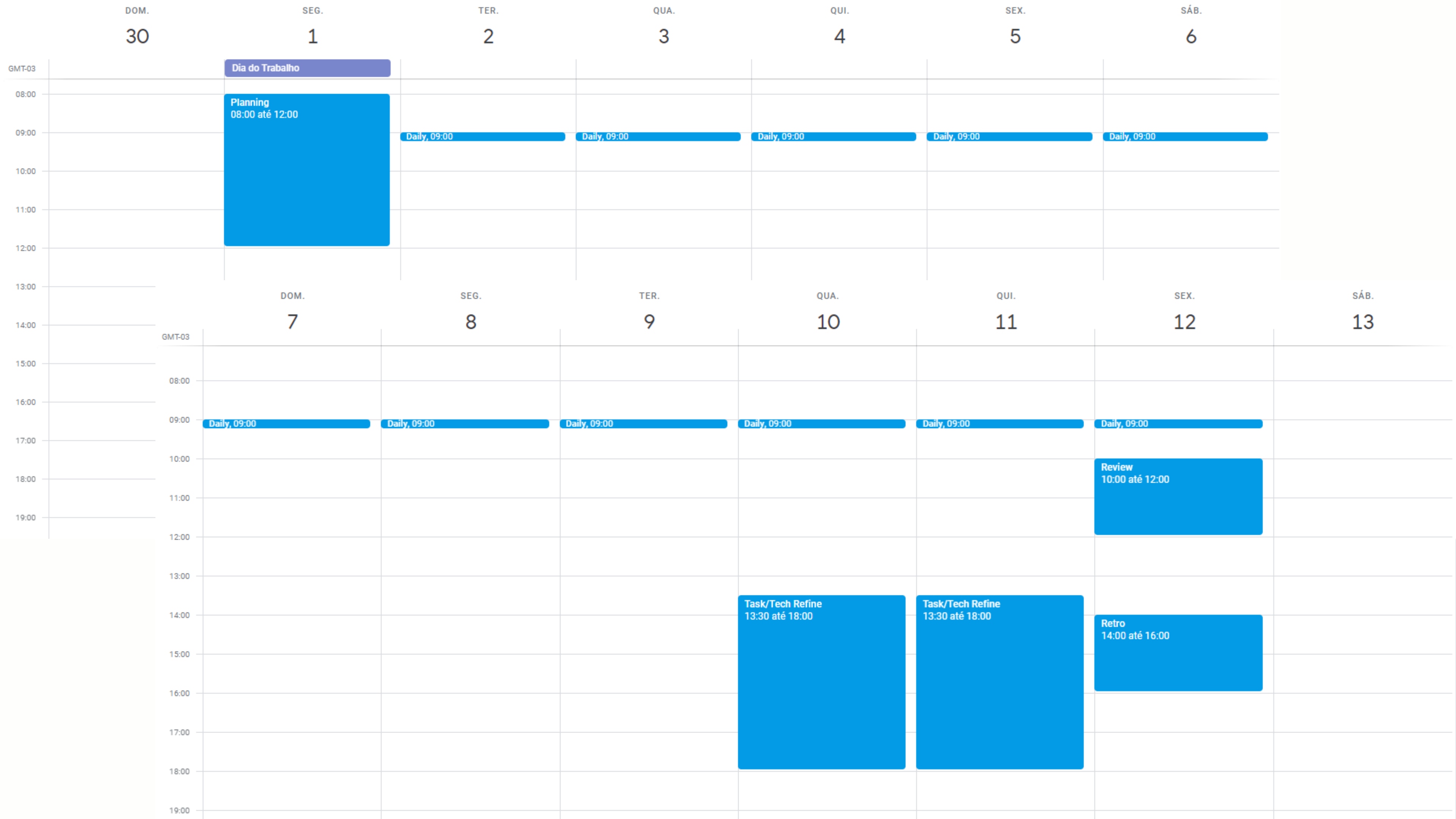
PO - Product Owner

DEV - Development Team

SM - Scrum Master

Super Man!



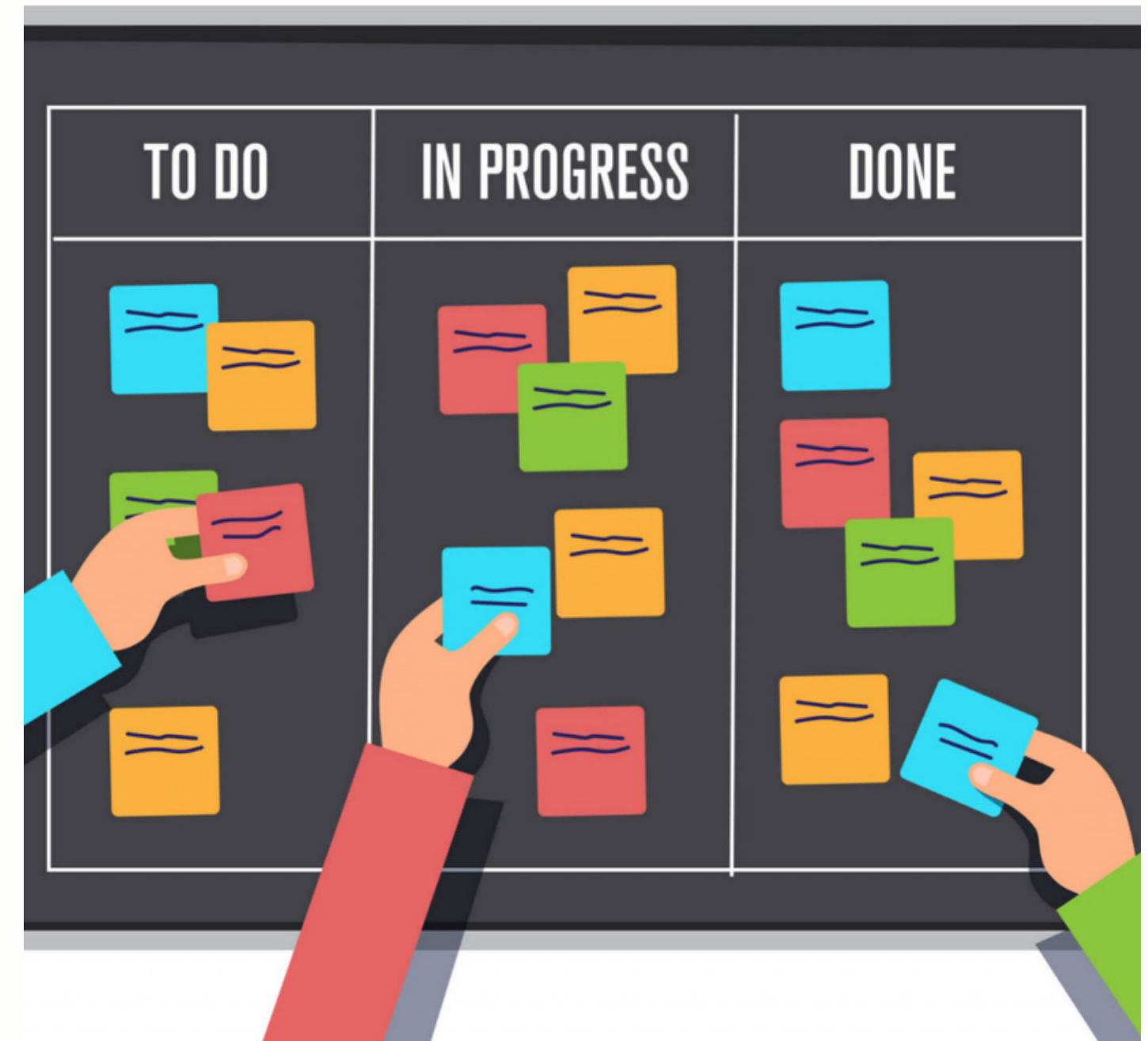


KANBAN_

Significa visual ou cartão

Visualização do fluxo de trabalho
Limitação da quantidade de trabalho
Medição de otimização do fluxo
Políticas de processos claras
Gerenciamento quantitativo

O plug-in perfeito pro SCRUM!!



PRÁTICAS

- **Foco na qualidade** - testes automatizados, revisão de código, análise colaborativa, Design Pattern ajudam a aumentar a produtividade e qualidade.
- **Limite de quantidade de trabalho em progresso** - reduzir a quantidade de trabalho, reduz a depreciação do conhecimento adquirido, aumentando a qualidade e confiança.
- **Entregas frequentes** - pequenas entregas frequentes agradam mais equipes parceiras e geram maior confiança.
- **Balanceamento de demanda com capacidade máxima** - necessidades não podem ser maior do que a capacidade de entrega.
- **Priorização das tarefas** - foco no que entrega valor.
- **Combate às fontes de variedade** - tudo o que pode, de alguma forma, prejudicar o desempenho.



COM TUDO ISSO —

— MÉTRICAS ÁGEIS

- **Througput** - quantidade de trabalho entregue por período
- **Lead Time** - Tempo do início à conclusão da tarefa
- **Flow Efficiency** - Tempo investido vs Tempo até a conclusão (Lead Time)
- **Velocity** - Tarefas concluídas vs estimativa por sprint
- **Cycle Time** - Tempo por etapa do trabalho
- **Burndown** - Estimativa de conclusão de pontos por tempo
- **CFD** - Diagrama de Fluxo Cumulativo, visualiza por período o status do trabalho em andamento
- **Scope Changes** - Incremento de novas tarefas na sprint após o início

Aqui surge uma famosa e importante frase "Pare de começar e comece a terminar"

Sem esquecer que o que realmente vale é software funcionando entregue ao cliente!

