PDLC/SDLC com IA

Dia 2 - Projeto Greenfield

Da verificação à especificação

Tradicional

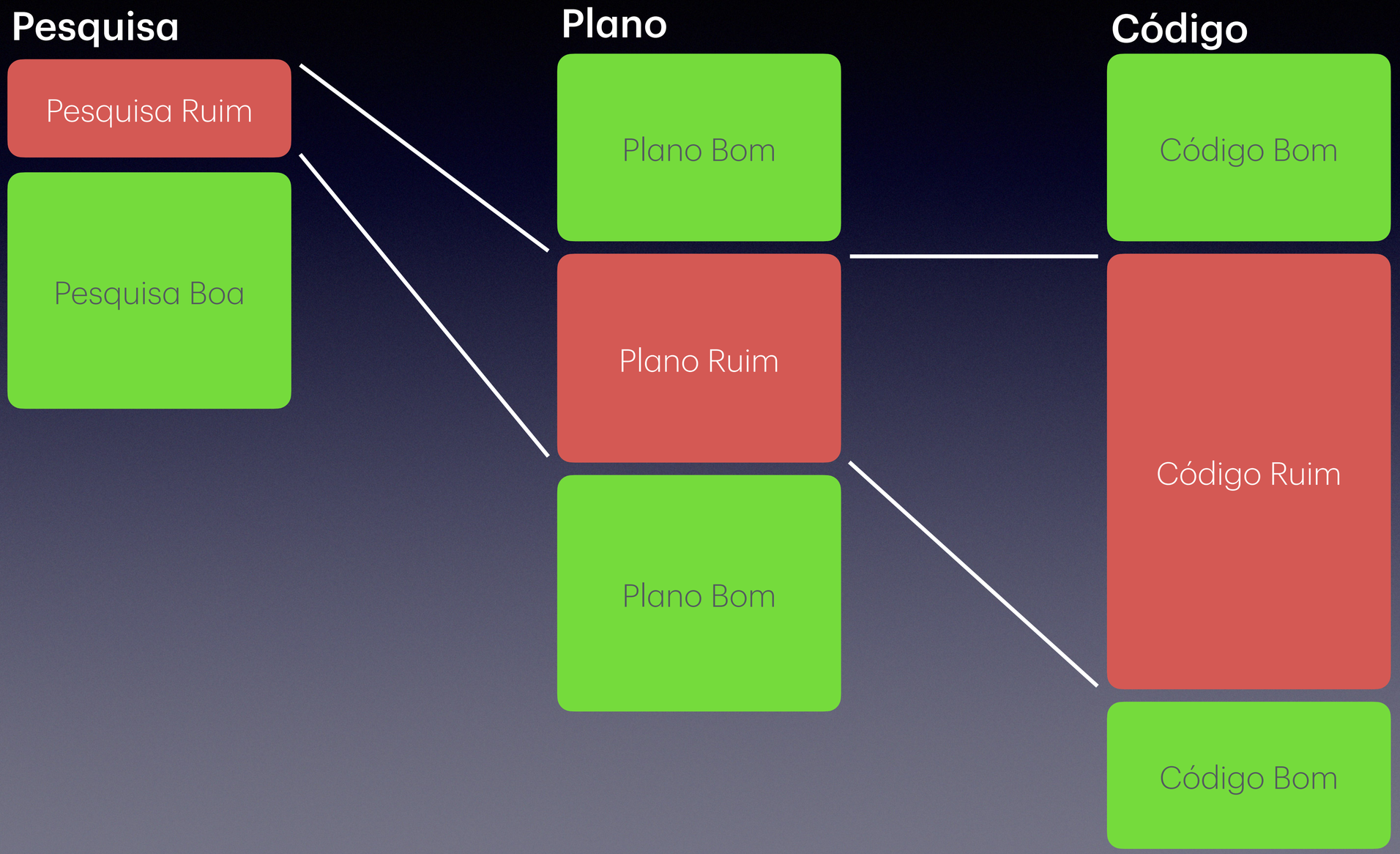
Entender → Design → Codificar → Testar

Orientado por especificações

Especificar → Gerar → Verificar → Iterar

Com uma verificação robusta, você pode buscar soluções em vez de criá-las manualmente.

Qualidade das Etapas



Desenvolvimento com IA

O “Gargalo do Contexto”

Programação tradicional com IA (Vibe Coding)

Iteração Inicial
(Boilerplate, Basics)

Progresso Rápido 😄
(Messages 1-20)

Ponto da Dor
(Depois de ~20 Mensagens)

Alucinações
(Variáveis excluídas)

Esquece Restrições
(TS → JS)

Sugestões Redundantes
(Instala biblioteca existente)

Causa Principal: Memória de Curto Prazo
(Janela de Contexto Temporária/ Paradigma da Conversa Informal”)

Método “Desenvolvimento baseado em Contexto”

Método + Agentes

Fase de Pré-Código
(Brief, PRD, Architecture)

Gestão com Agentes Inteligentes
(Entendimento de Longo Prazo do Projeto)

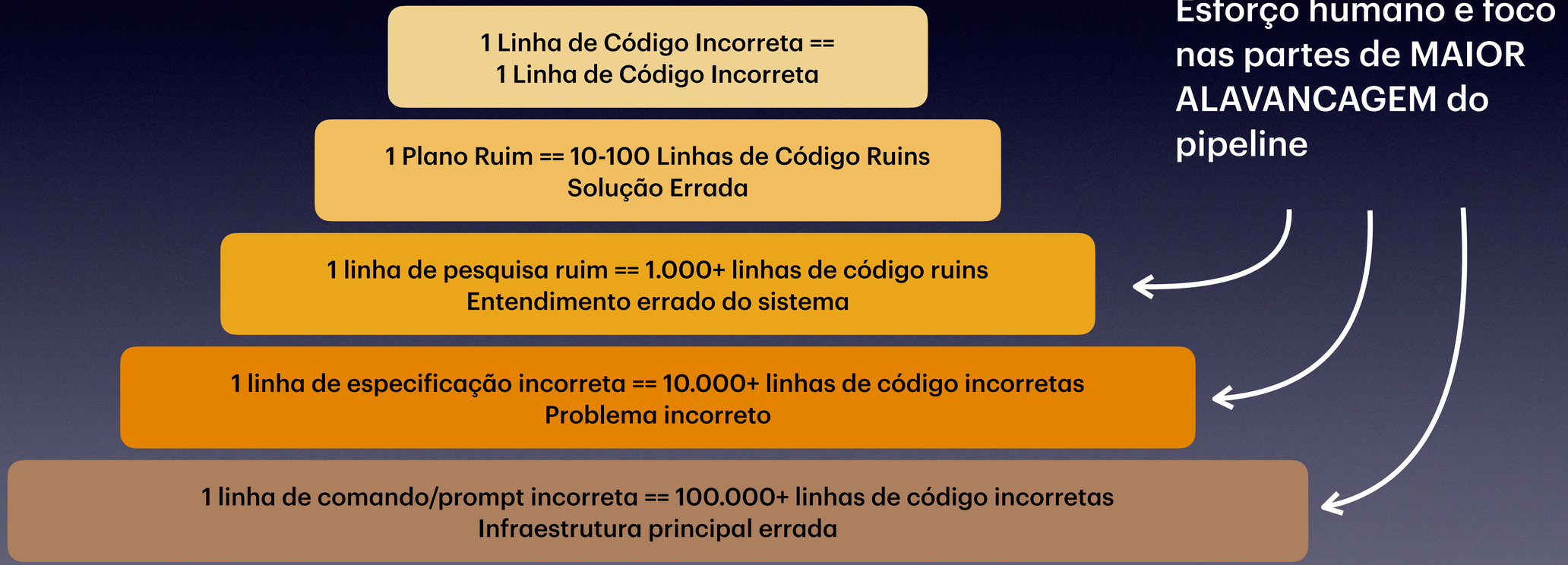
Refatoração Precisa

Respeita as Restrições

Sugestões Ligadas ao Contexto

Mudança de Paradigma: Gerenciando o Agente
(Impõe planejamento antes da codificação)

Hierarquia de Alavancagem



Fases

Fase de Planejamento

Requsitos de projeto, arquitetura e detalhamento do trabalho

1. **Analista de Negócios** (Analyst)
2. **Gerente de Produto** (PM)
3. **Especialista UI/UX** (UI/UX)
4. **Arquiteto** (Architect)
5. **Gerente de Projeto** (PO)

Fase de Desenvolvimento

Criar histórias, escrever código, validar a qualidade

1. **Scrum Master** (SM)
2. **Desenvolvedor** (DEV)
3. **Garantia de Qualidade** (QA)

Analista de Negócios

Analyst

Saídas:
Briefing do projeto (brief.md)

Transferência:
Início → Gerente de Projeto (PM)

Comando de inicialização:
/agents/analyst

Obs:
Primeiro agente de fluxo Greenfield. Coleta requisitos e cria o briefing do projeto.

Gerente de Projeto

PM

Saídas:
Documento de Requisitos do Projeto (PRD)

Transferência:
Analista → PM ←→ PO

Comando de inicialização:
/agents/pm

Obs:
Cria o doc de requisitos do projeto (PRD). Retorna depois para alinhar o PRD com a Arquitetura.

Arquiteto

Architect

Saídas:
Doc de Arquitetura, Stack de Tecnologia, Modelos de Dados, Especificações de API, Padrões de Código

Transferência:
Architect → PM (alinhar) → PO

Comando de inicialização:
/agents/architect

Obs:
Cria uma arquitetura abrangente. Devolve ao PM para alinhamento, depois ao PO para fragmentação.

Gerente de Produto

PO

Saídas:
Documentos fragmentados (PRD/Architecture)

Transferência:
PM + Architect → SM

Comando de inicialização:
/agents/po

Obs:
Fragmenta documentos em partes implementáveis.
Ponte entre planejamento e desenvolvimento.

Scrum Master

Scrum Master

Saídas:
Histórias de usuário

Transferência:
PO → Dev

Comando de inicialização:
/agents/sm

Comando principal:
*draft

Obs:
Cria histórias de usuário detalhadas dos épicos.
Primeiro agente na fase de desenvolvimento.

Desenvolvedor

Developer

Saídas:
Código fonte, testes, implementação

Transferência:
SM → QA

Comando de inicialização:
/agents/dev

Comando principal:
*develop-story

Obs:
Implementa as histórias. Somente atualiza os itens e a seção de Agente de Desenvolvimento.

Garantia de Qualidade

QA

Saídas:
Revisões de Qualidade, Design de Testes, Portões de Qualidade

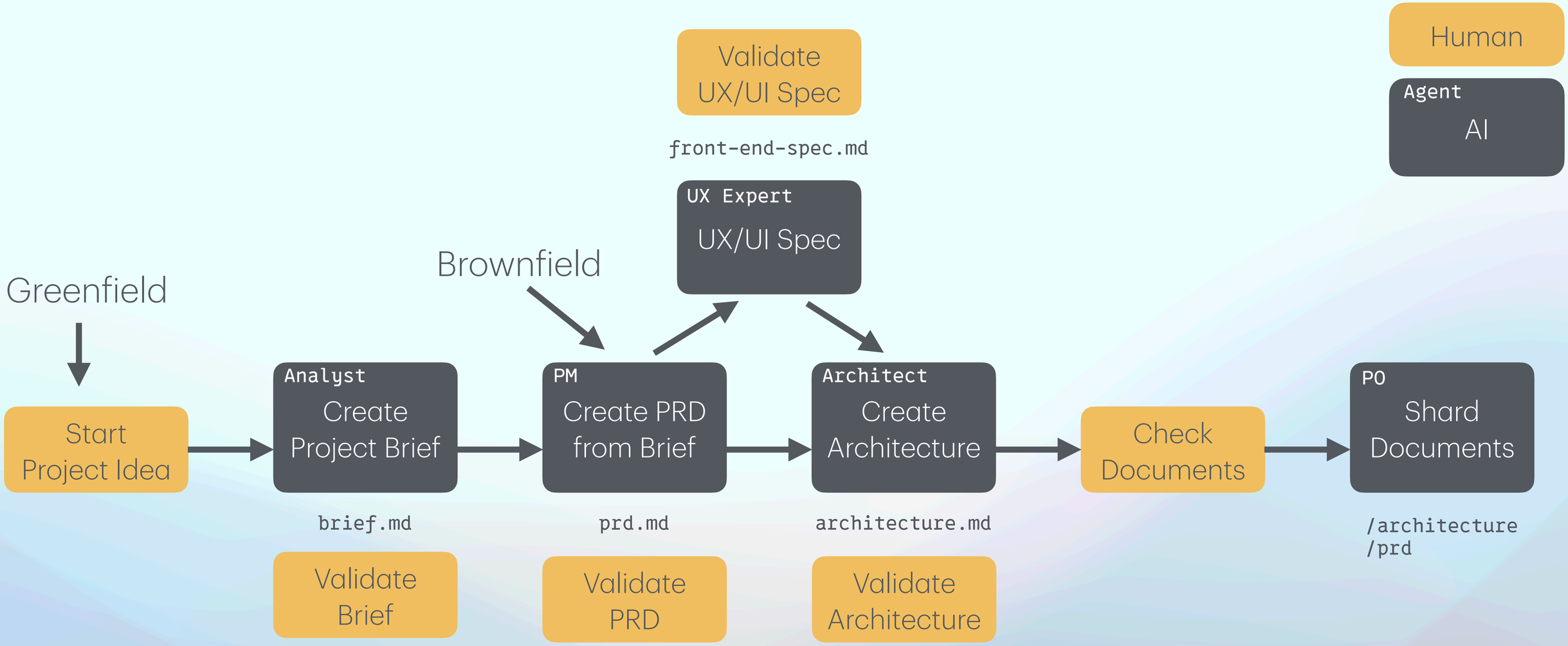
Transferência:
DEV ←→ DEV (iteração)

Comando de inicialização:
/agents/qa

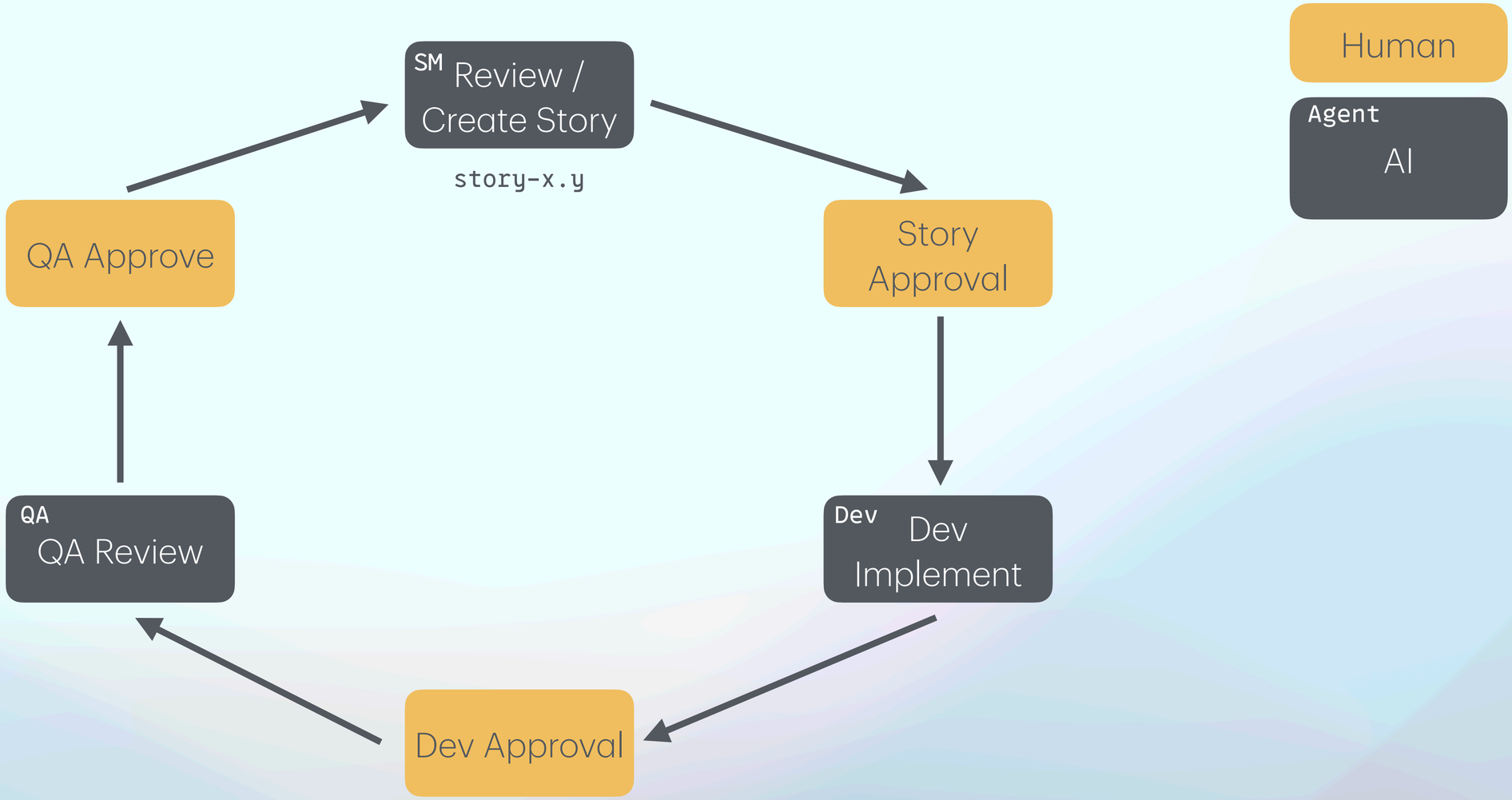
Comando principal:
*review, *design, *gate

Obs:
Revisa a qualidade do código.

Agentic Planning



Context-Engineered Development



Ferramentas de Desenvolvimento IA

- Github Copilot
- Cursor
- Claude Code
- Antigravity, Gemini Cli (Google)
- Kiro Code (Amazon)
- VS Code - RooCode, Kilo Code, Cline, etc

Frameworks de Desenvolvimento IA

- Spec Kit (Github)
- OpenSpec
- BMAD

Utilizando o BMAD

Breakthrough Method of Agile AI Driven Development

https://github.com/bmad-code-org/BMAD-METHOD