



MODELOS ÁGEIS II

LEAN, KANBAN, SCRUMBAN E XP

Prof. Raul Sidnei Wazlawick

UFSC-CTC-INE

CONTEÚDO



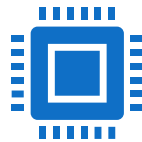
Lean



Kanban



Scrumban



XP – eXtreme
Programming



FILOSOFIA LEAN



Eliminar o desperdício.



Amplificar a aprendizagem.



Decidir o mais tarde possível.



Entregar o mais rápido possível.



Empoderar a equipe.



Construir integridade.



Ver o todo.



KANBAN



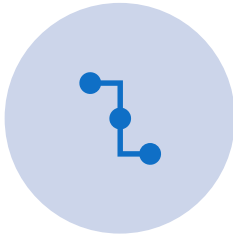
PROCESSO CONTÍNUO,
SEM SPRINTS



NÚMERO DE TAREFAS
LIMITADO POR COLUNA



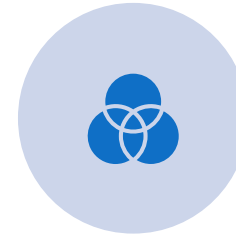
PRINCÍPIOS FUNDAMENTAIS DE KANBAN



COMECE COM O QUE
VOCÊ FAZ AGORA.



CONCORDAR EM
BUSCAR MUDANÇAS
EVOLUCIONÁRIAS.



INICIALMENTE,
RESPEITE OS PAPEIS,
RESPONSABILIDADES
E CARGOS ATUAIS.



INCENTIVAR ATOS DE
LIDERANÇA EM
TODOS OS NÍVEIS.



PRÁTICAS KANBAN



Visualizar o fluxo de trabalho.



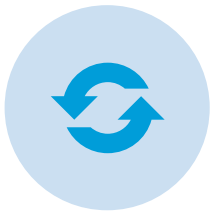
Limitar a quantidade de trabalho em andamento.



Gerenciar e medir o fluxo.



Tornar as políticas do processo públicas.



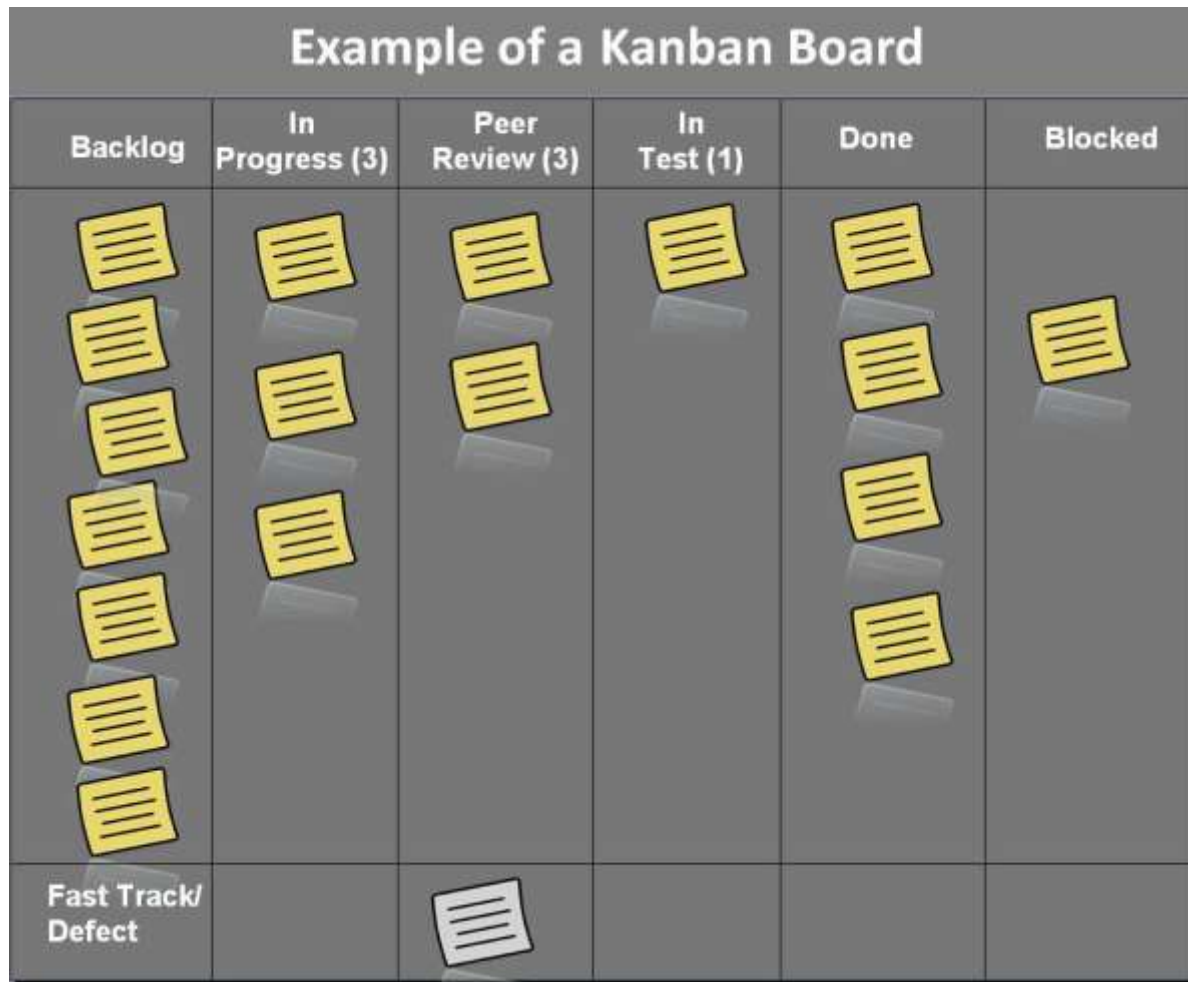
Implementar loops de feedback.



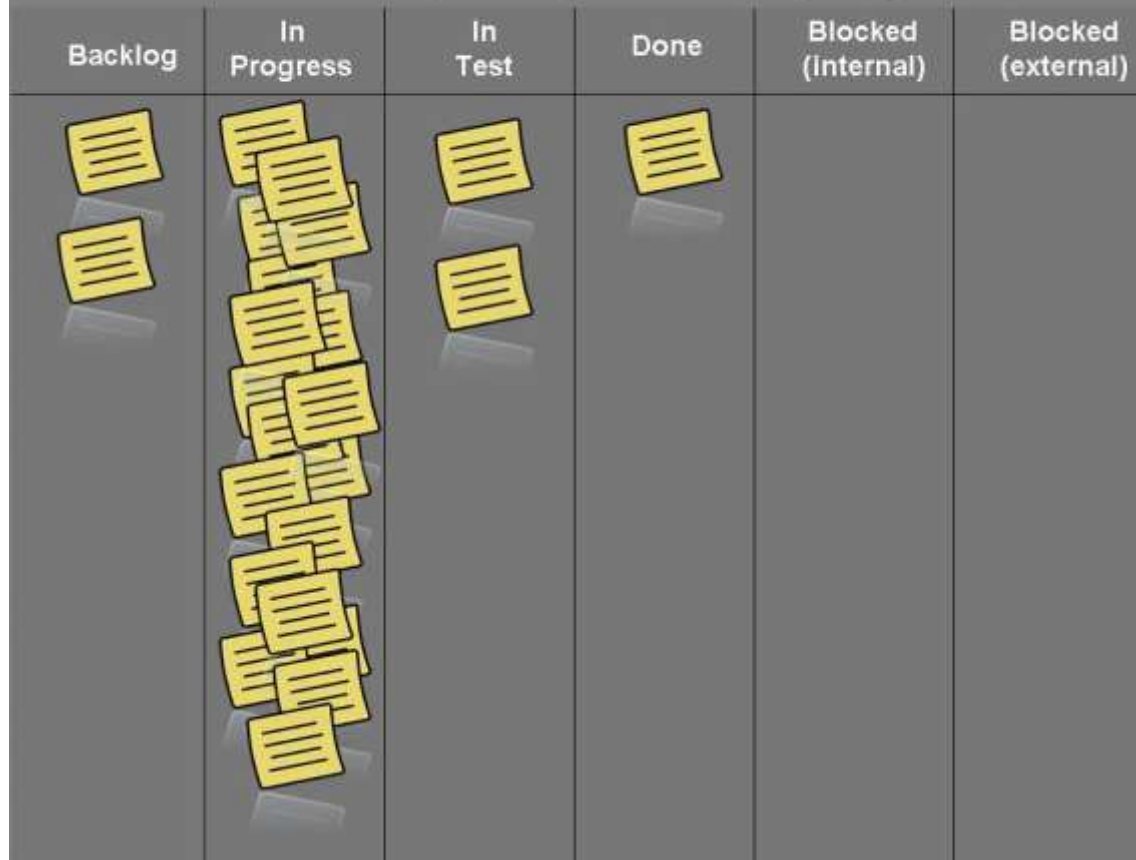
Usar modelos para reconhecer oportunidades de melhoria.



QUADRO KANBAN



Example of a first attempt at a Kanban board where too many things are "in progress"



ITENS DOS CARTÕES



ÉPICOS



HISTÓRIAS
DE USUÁRIO



TAREFAS



Pool of Ideas	Feature Preparation		Feature Selected	User Story Identified	User Story Preparation		User Story Development		Feature Acceptance		Deployment	Delivered
	3 - 10		2 - 5	30	15		15		8		5	
Epic 431	In Progress	Ready			In Progress	Ready	In Progress	Ready (Done)	In Progress	Ready		Epic 294
Epic 478	Epic 444	Epic 662	Epic 602			Story 602-02 Story 602-03	Story 602-06 Story 602-04	Story 602-05 Story 602-01	Epic 401	Epic 609	Epic 694	Epic 386
Epic 562	Epic 589		Epic 302	Story 302-03 Story 302-02	Story 302-01 Story 302-06	Story 302-07 Story 302-08	Story 302-09	Story 303-05 Story 302-04	Epic 468	Epic 577	Epic 276	Epic 419
Epic 439	Epic 651		Epic 335	Story 335-09 Story 335-08	Story 335-10 Story 335-01 Story 335-03	Story 335-05 Story 335-02	Story 335-06 Story 335-07		Epic 362		Epic 339	Epic 388
Epic 329			Epic 512	Story 512-04 Story 512-05	Story 512-07 Story 512-06 Story 512-03	Story 512-01					Epic 521	Epic 287
Epic 287											Epic 582	Epic 274
Epic 606	Discarded											
	Epic 511	Epic 213										
	Epic 221											

Policy

Business case showing value, cost of delay, size estimate and design outline.

Policy

Selection at Replenishment meeting chaired by Product Director.

Policy

Small, well-understood, testable, agreed with PD & Team

Policy

As per "Definition of Done" (see...)

Policy

Risk assessed per Continuous Deployment policy (see...)

SCRUMBAN

Aspecto	Kanban	Scrum	Scrumban
Papeis	Não há	Product owner, Scrum master e development team	A equipe e quaisquer papeis que se façam necessários
Reuniões diárias	Não há	Daily meeting	Usadas para garantir o trabalho contínuo e reduzir o tempo de ociosidade dos membros da equipe
Review e retrospective	Não há	As duas reuniões são prescritas no final das sprints	Podem ser feitas na medida do necessário para aprimorar o conhecimento e o processo
Fluxo de trabalho	Contínuo	Sprints	Contínuo, mas iterações regidas por timeboxing podem ser usadas eventualmente
Artefatos	Quadro Kanban	Product backlog e sprint backlog	Quadro Kanban
Equipes	Não menciona	Equipes multifuncionais focadas em requisitos	Equipes podem ser especializadas tecnicamente
WIP	Controlada pelo estado do quadro Kanban	Controlada pelo conteúdo da sprint	Controlada pelo estado do quadro Kanban



XP – EXtREME PROGRAMMING

- Kent Beck



- Valores

- Simplicidade
- Respeito
- Comunicação
- Feedback
- Coragem



PRINCÍPIOS XP



FEEDBACK
RÁPIDO.



PRESUMIR
SIMPLICIDADE.



MUDANÇAS
INCREMENTAIS.



ABRAÇAR
MUDANÇAS.



TRABALHO DE
ALTA
QUALIDADE.



PRÁTICAS XP



Feedback:

jogo de planejamento
programação em pares
desenvolvimento dirigido por teste
equipe coesa



Processo contínuo:

integração contínua
entregas pequenas
excelência técnica



Compreensão compartilhada:

metáfora de sistema
padrões de codificação
design simples
posse coletiva do código



Bem-estar dos desenvolvedores:

ritmo sustentável



JOGO DE PLANEJAMENTO

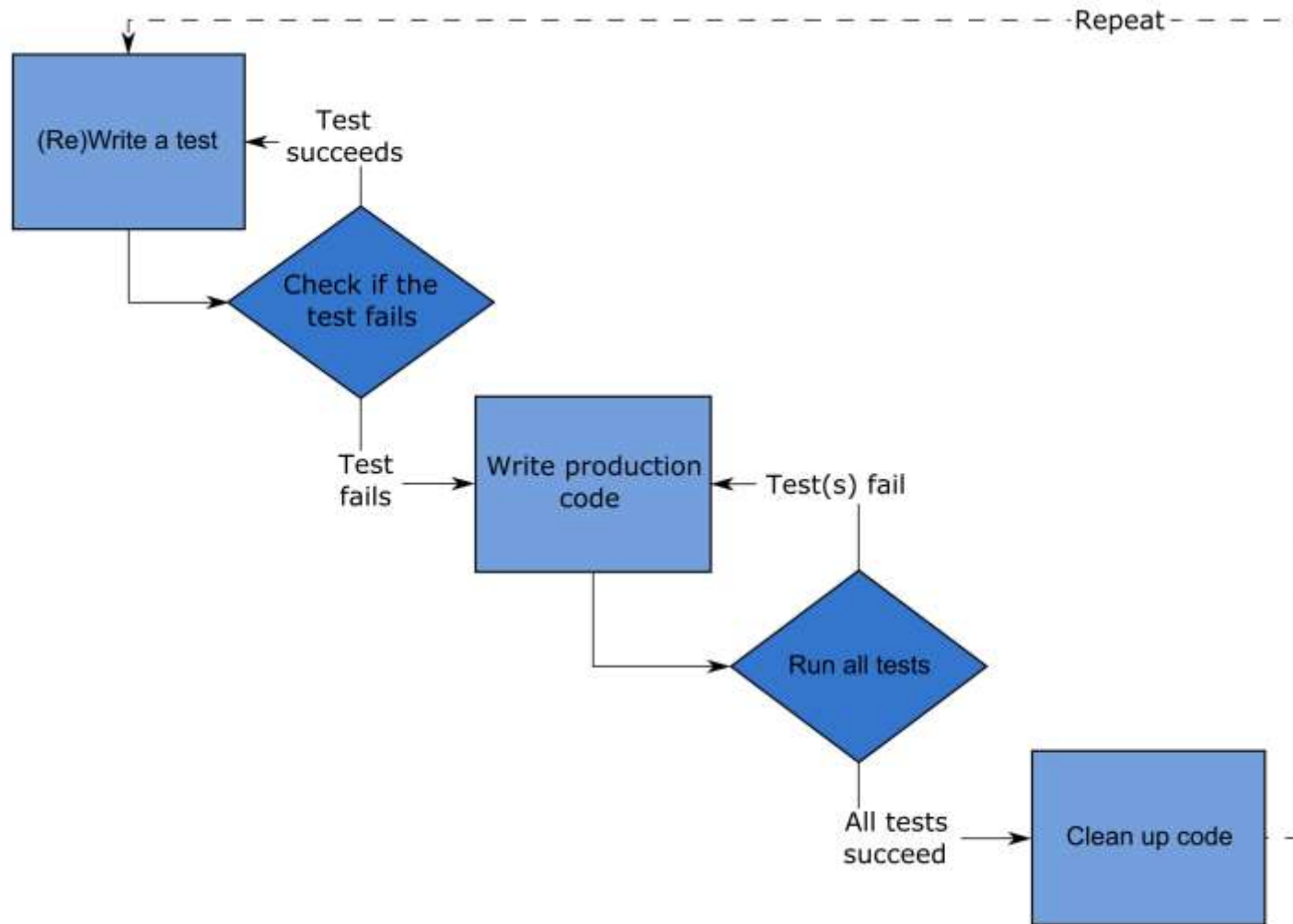
- Planejamento de Entregas
- Planejamento de Iteração
- Exploração
- Concordância
- Direção



PROGRAMAÇÃO EM PARES



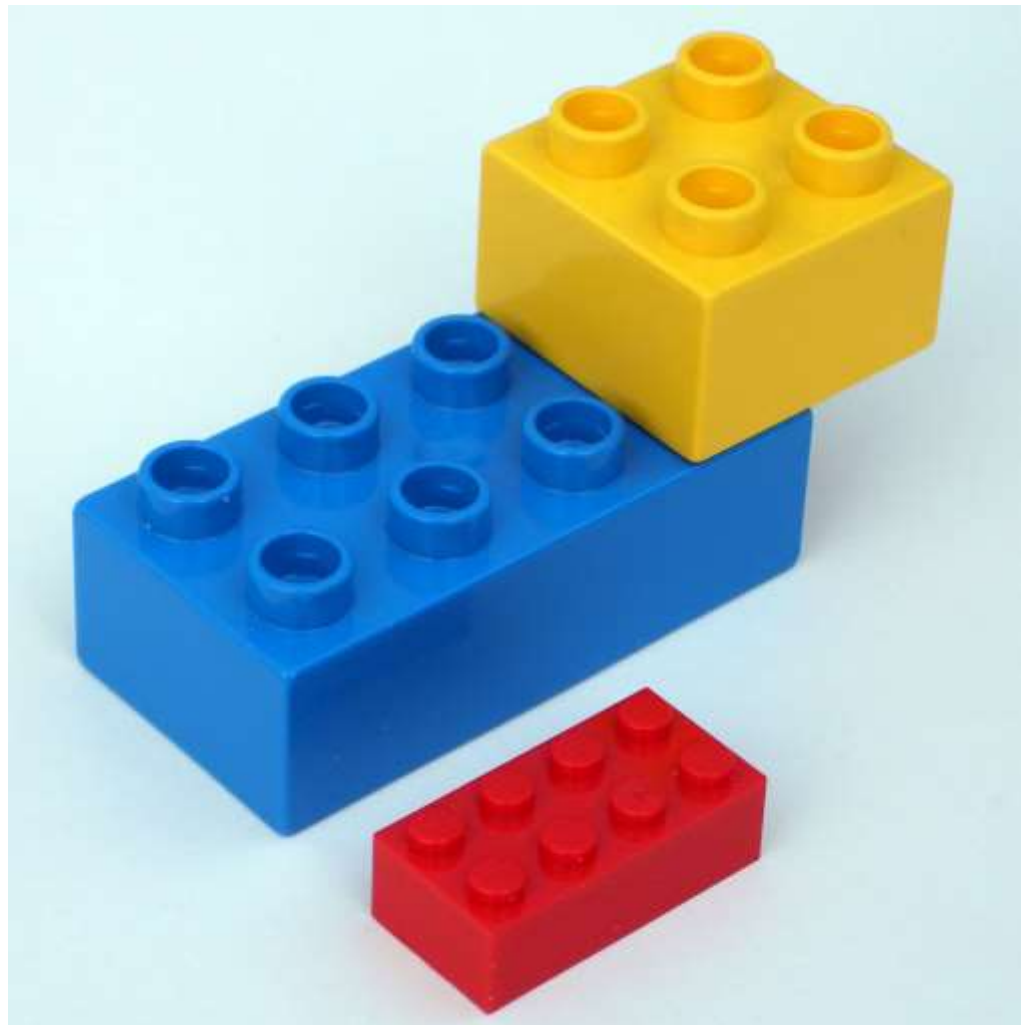
DESENVOLVIMENTO DIRIGIDO POR TESTE



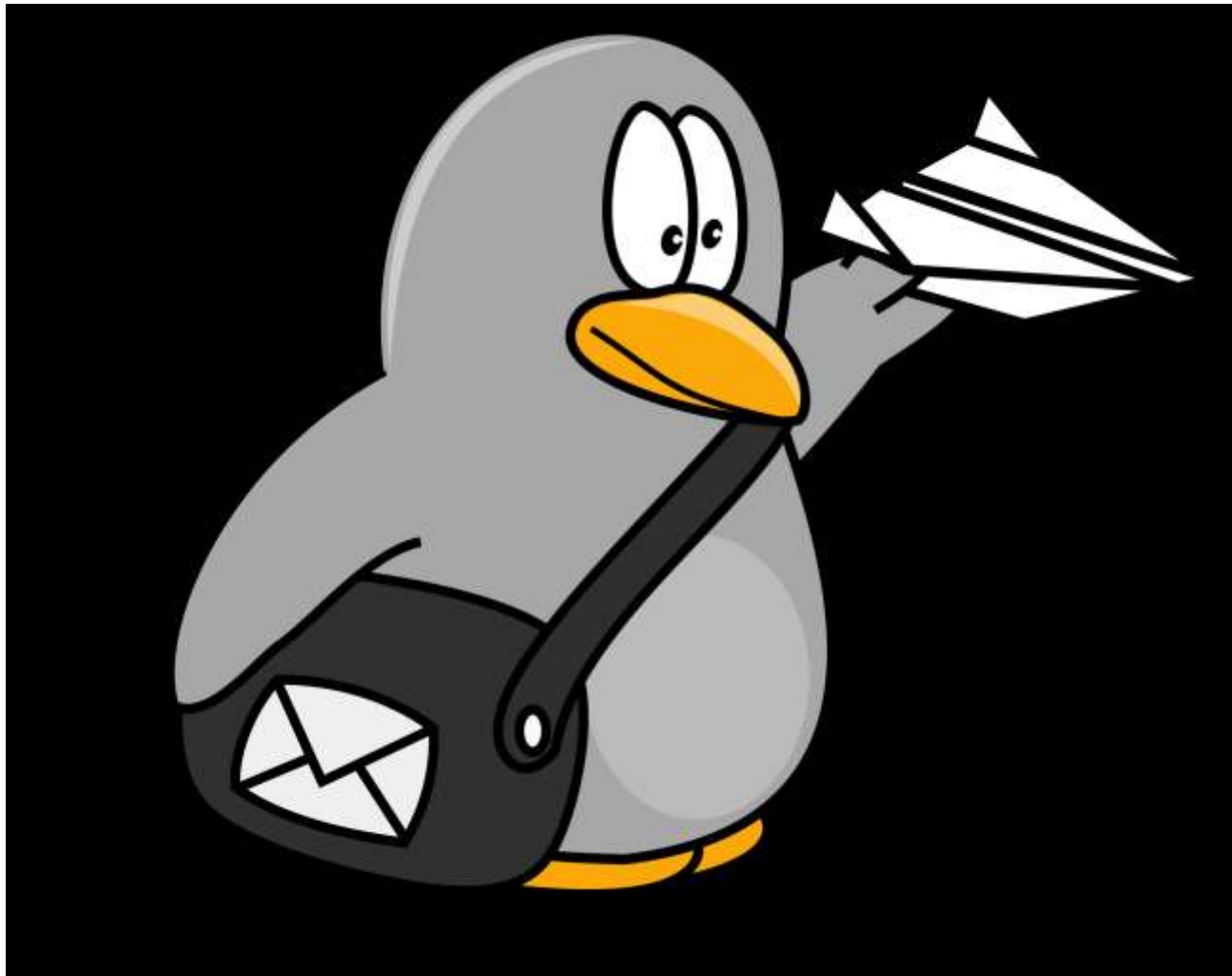
EQUIPE COESA



INTEGRAÇÃO CONTÍNUA



ENTREGAS PEQUENAS



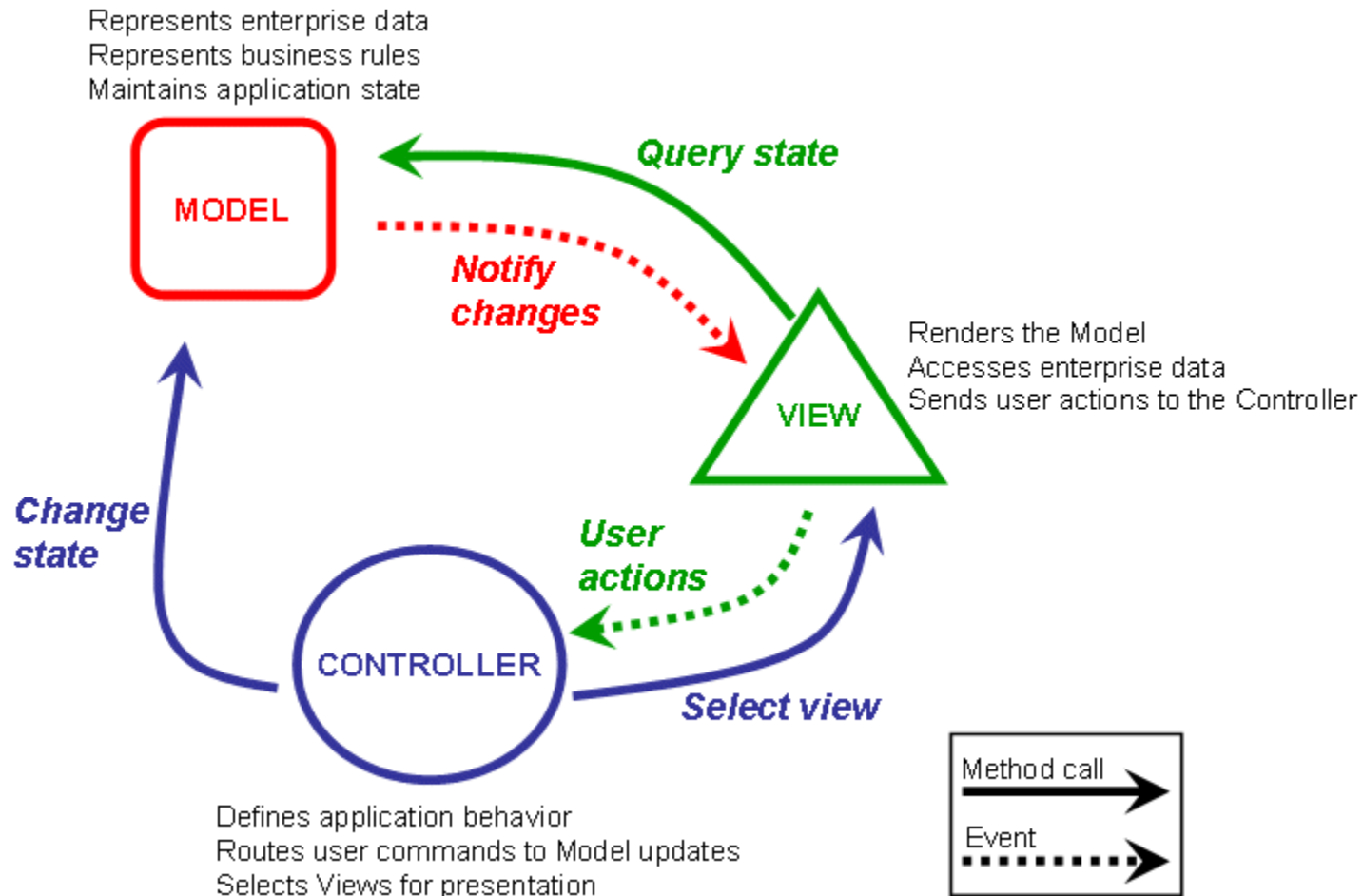
EXCELÊNCIA TÉCNICA



METÁFORA DE SISTEMA



PADRÕES DE CODIFICAÇÃO



DESIGN SIMPLES



POSSE COLETIVA DO CÓDIGO



RITMO SUSTENTÁVEL



REGRAS DE ENGAJAMENTO



Uma equipe XP consiste em um grupo de pessoas que se reúne para desenvolver um produto de software.



O cliente deve definir e continuamente ajustar os objetivos e prioridades, baseando-se em informações fornecidas pelos desenvolvedores e outros participantes.



O cliente está sempre disponível e fornece informações sob demanda para ajudar os desenvolvedores a formar estimativas ou fornecer os produtos necessários.



Em qualquer momento, qualquer membro da equipe deve ser capaz de mensurar o progresso da equipe na direção dos objetivos do cliente.



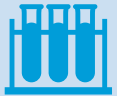
A equipe deve atuar como uma *rede social efetiva*, o que significa que deve haver comunicação honesta que leva à aprendizagem contínua, grau mínimo de separação entre o que é necessário para que a equipe faça progresso e as pessoas ou recursos que podem atender essas necessidades e alinhamento de autoridade e responsabilidade.



Timeboxing é usado para identificar segmentos de esforço de desenvolvimento e cada segmento não deve durar mais de um mês.



REGRAS DE FUNCIONAMENTO



O trabalho produzido deve ser continuamente validado pelo teste.



Escreva os testes de unidade antes do código, programe em pares e refatore o código para atender aos padrões de codificação.



Todo o código escrito para uso potencial no produto deve passar em todos os testes de unidade.



A posse do código é coletiva.



REGRAS DE PLANEJAMENTO



Escrever histórias de usuário.



O planejamento de entregas cria o cronograma de entregas.



Faça entregas pequenas frequentes.



O projeto é dividido em iterações.

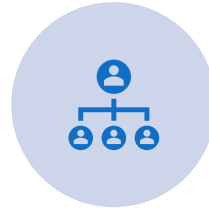


O planejamento da iteração inicia cada iteração.

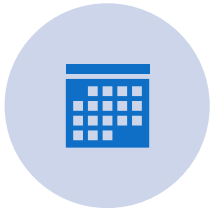
REGRAS DE GERENCIAMENTO



Dê à equipe um espaço de trabalho aberto e dedicado.



Defina uma jornada sustentável.



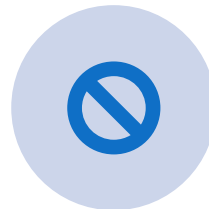
Inicie cada dia com uma reunião em pé.



A velocidade do projeto é medida.



Mova as pessoas.



Conserte XP quando for inadequado.



REGRAS DE DESIGN



Adote a simplicidade:

Testabilidade.
Browseabilidade.
Compreensibilidade e explicabilidade.



Escolha uma metáfora de sistema.



Use cartões CRC durante reuniões de projeto.



Crie spikes para reduzir riscos.



Nenhuma funcionalidade é adicionada antes da hora.



Use refatoração sempre e onde for possível.



REGRAS DE CODIFICAÇÃO



O cliente está sempre disponível.



O código deve ser escrito de acordo com padrões aceitos.



Escreva o teste de unidade primeiro.



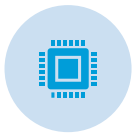
Todo código é produzido por pares.



Só um par faz integração de código de cada vez.



Integração deve ser frequente.



Defina um computador exclusivo para integração.



A posse do código deve ser coletiva.



REGRAS DE TESTE



Todo código deve ter testes de unidade.



Todo código deve passar pelos testes de unidade antes de ser entregue.



Quando um erro de funcionalidade é encontrado, testes de aceitação são criados.



Testes de aceitação são executados com frequência e os resultados são publicados.

