



INSTITUTO FEDERAL

Espírito Santo
Campus Serra

Gestão de Projetos de Software

Prof.: Elton Siqueira Moura

(elton@ifes.edu.br)

GPS | Campus Serra | Serra, ES

Lean Development

O material do curso estará disponível no ambiente
virtual de aprendizagem Moodle
moodle.ifes.edu.br – Gestão de Projetos de Software

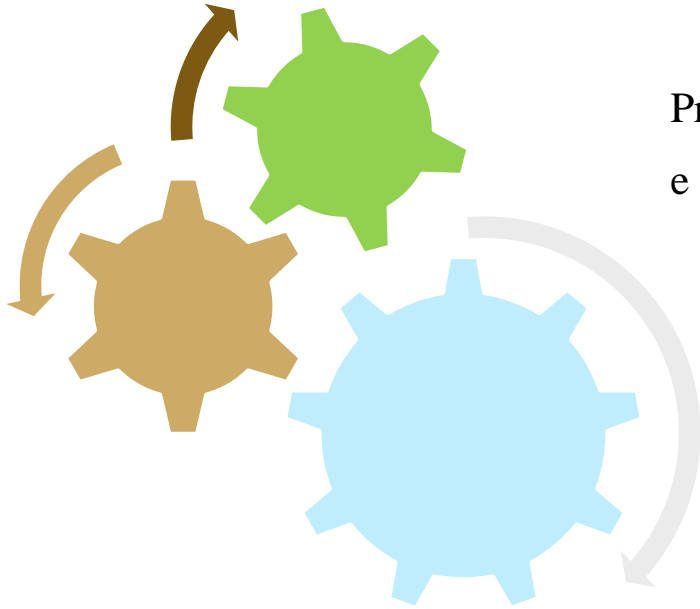
ifes.edu.br/

Objetivos de aprendizagem

Ao final desta aula espera-se que o aluno seja capaz de:

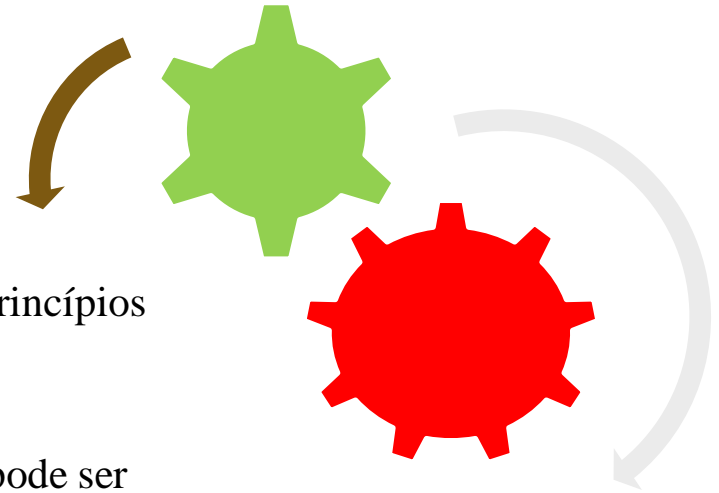
1. Reconhecer os princípios do desenvolvimento Lean.
2. Identificar o propósito das ferramentas que são utilizadas no método Lean de desenvolvimento de software.

Princípios chaves



Processos de desenvolvimento devem ser simplificados, eficientes e voltados para a produção de resultados de alta qualidade.

Foco na simplificação do desenvolvimento.

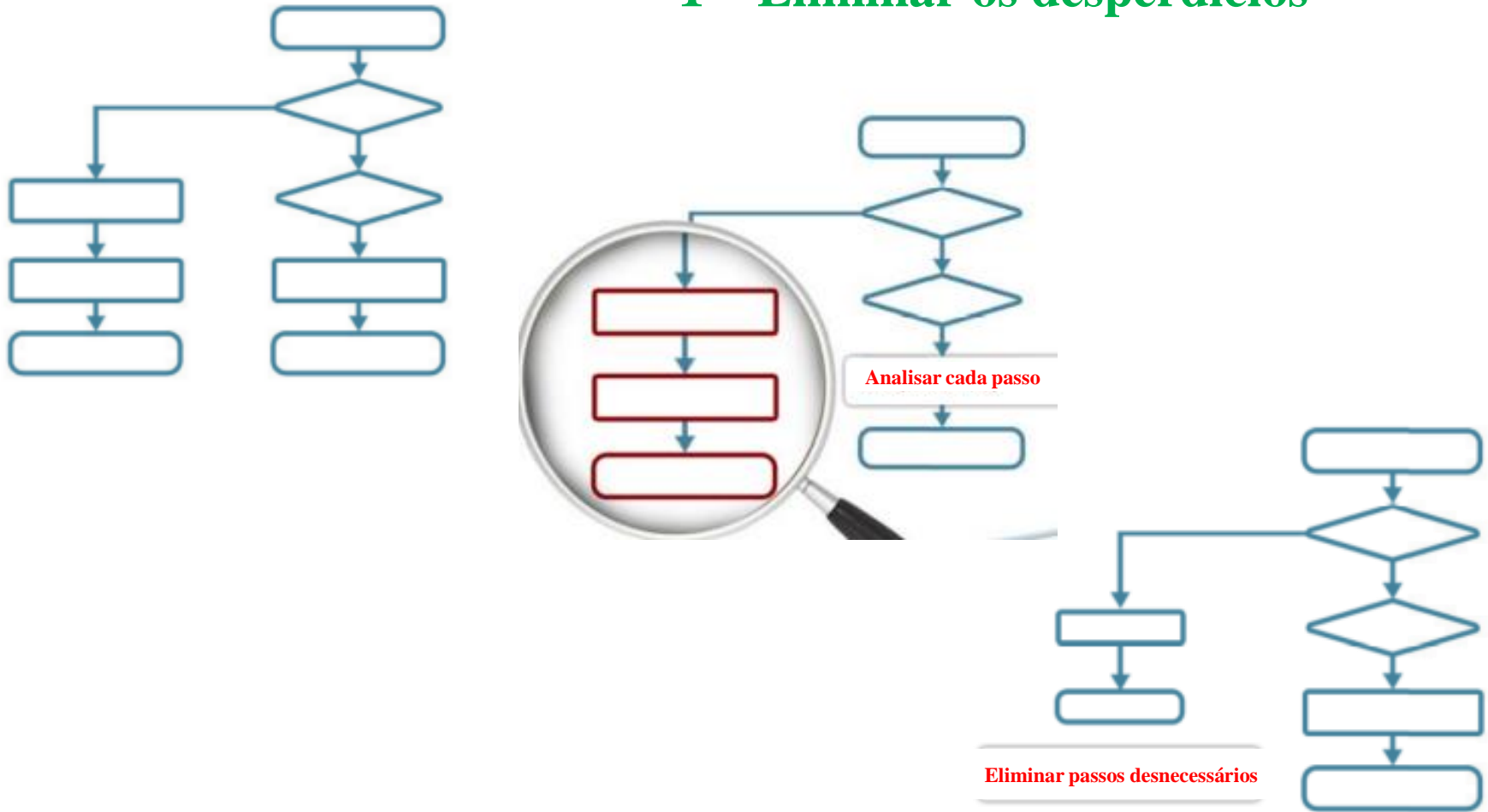


Não há prescrições para a realização do trabalho e, sim, princípios a serem seguidos.

Tem como base a Produção Enxuta (Lean Production) e pode ser utilizada em conjunto com outras metodologias ágeis.

Princípios chaves

1 – Eliminar os desperdícios



Princípios chaves

Desperdícios em projetos de TI

Processos desnecessários:

Procedimentos administrativos pesados, desperdícios de tempo que poderiam ser revertidos em tempo de desenvolvimento. Ex: Uso de formulários de assinaturas e preenchimento de quadro de horários.

Atrasos na produção:

Qualquer tempo gasto aguardando – esperando trabalho – é considerado um desperdício. Desperdícios de tempo privam o cliente de receber mais rapidamente o valor proveniente do projeto. Ex: Aguardar código ser desenvolvido e aguardar a gestão aprovar as tarefas.

Multitarefa:

Realizar tarefas simultâneas, alternando entre tarefas novas e diferentes requer uma mudança de foco e tempo para ajustes (*setup*).

Princípios chaves

Desperdícios em projetos de TI

Trabalho incompleto:

Qualquer trabalho que ainda não foi liberado ao cliente ou que permanece inacabado é considerado como desperdício. Mesmo o trabalho que está incompleto porque ainda está sendo realizado representa um potencial desperdício, pois nenhum valor foi entregue ainda ao cliente.

Ex: código que ainda não foi integrado e código que necessita ser testado, bem como trabalho iniciado e abandonado devido a novos requisitos.

Princípios chaves

2 –Concentrar na integridade do processo de desenvolvimento

Percebida

Depende da percepção do cliente. Será alta se o cliente estiver satisfeito com o produto, o qual atende os requisitos, é altamente utilizável e possui boa qualidade e valor.

Ex: um cliente pode estar satisfeito com um sistema de workflow por estar ajustado ao exato fluxo de documento, é muito amigável, é comparável com os melhores produtos no mercado e foi entregue a um preço competitivo.

Conceitual

É julgada pelos desenvolvedores. É alta se os diferentes componentes de um produto se integram bem uns aos outros, produzindo um produto confiável e que atende os seus propósitos.

Ex: um sistema de gestão de pessoal tem integridade conceitual se a folha de pagamento, recursos humanos e módulos de acompanhamento de desempenho se integram transparente e eficientemente.

Princípios chaves

3 – Olhar o todo

Testar

Os testes precisam ocorrer em todo o sistema ao invés de apenas no nível de unidade ou componente. O objetivo é assegurar que os desenvolvedores se concentrem na funcionalidade total do sistema, ao invés de apenas em algumas partes.

Visa, também, em assegurar que problemas de integração não sejam descobertos tardiamente no processo de desenvolvimento, podendo atrasar a entrega do sistema.

Otimizar o fluxo de trabalho

Os desenvolvedores precisam considerar todos os processos de trabalho do projeto, ao invés de apenas aqueles associados com as suas próprias tarefas.

Uma vez encerradas suas tarefas de codificação, por exemplo, os desenvolvedores são encorajados a apoiar outros desenvolvedores cujos trabalhos são mais longos ou se provando mais complexo.

Desenvolvedores que completam suas codificações podem auxiliar os testadores a fim de prevenir backlogs em acumular e causar atrasos.

Princípios chaves

4 – Decidir tardiamente

Lean Development encoraja o uso de decisões tardias, até o último momento possível. Isso se deve pelo fato dos clientes quase sempre mudarem os seus requisitos e aspectos de design e codificação.

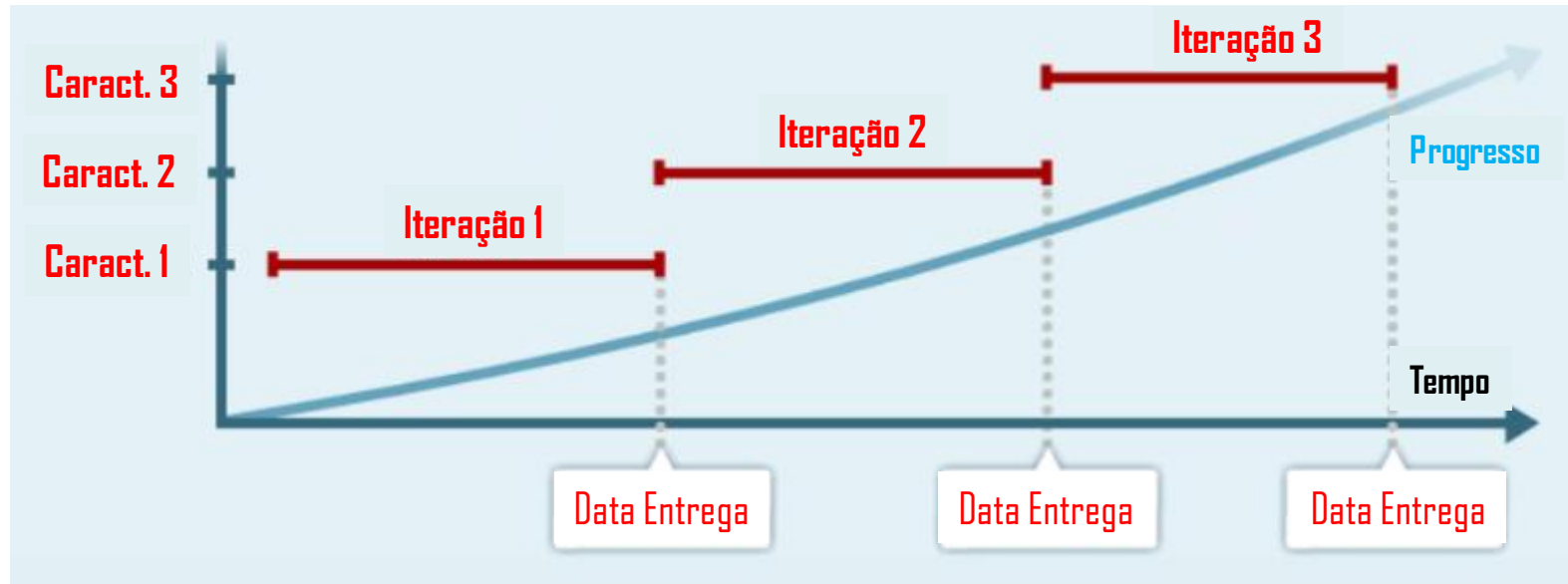
Vantagens das decisões tomadas tardiamente:

- Mantêm os desenvolvedores abertos às mudanças;*
- Assegura que decisões sejam feitas somente uma vez com o máximo de informações disponíveis;*
- Diminui incertezas e riscos, e ajuda a prevenir desperdícios.*

Princípios chaves

5 – Entregar rapidamente

Lean Development enfatiza entregas rápidas, com foco em reduzir tempo de mercado e atender as necessidades dos clientes tão rápido e eficientemente quanto possível.



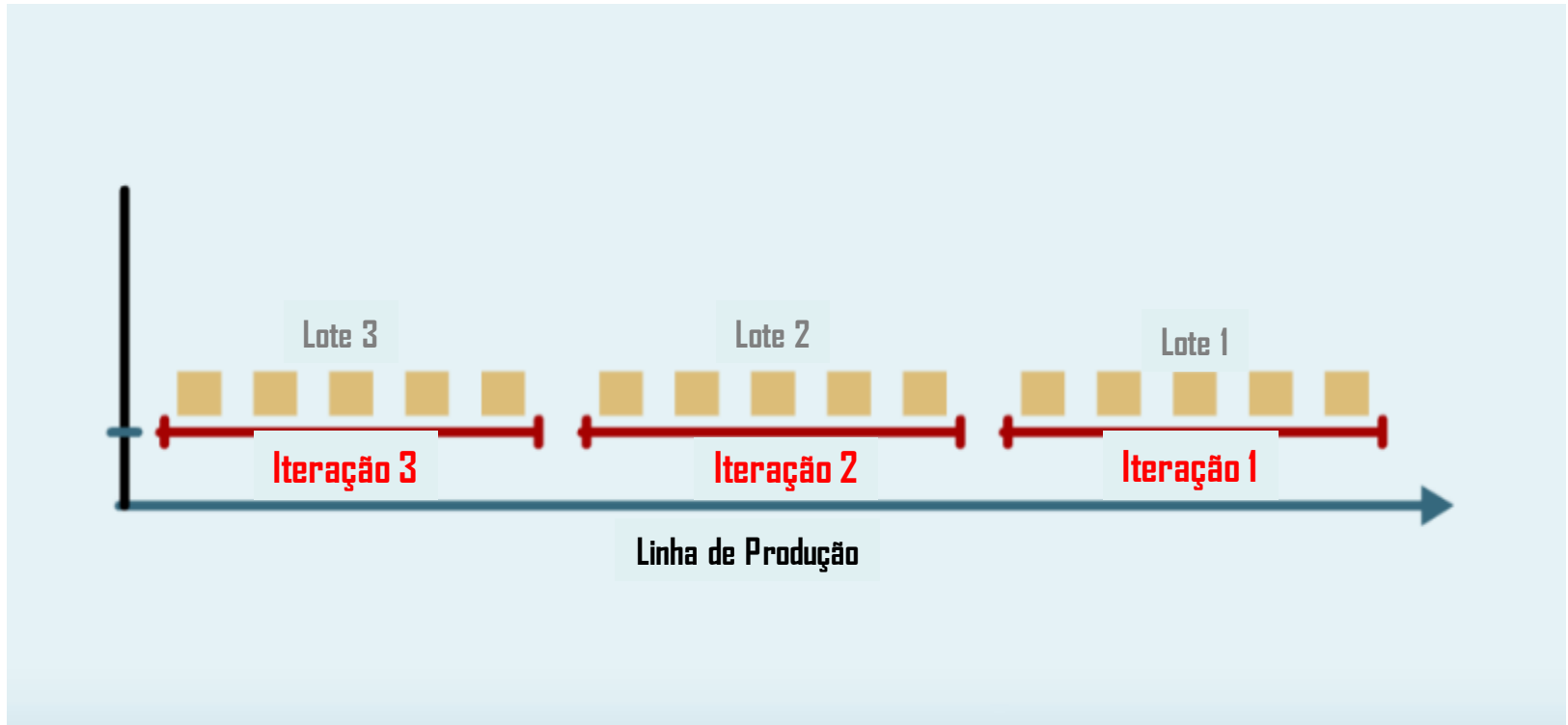
Identificar e remover gargalos e desperdícios no processo de desenvolvimento.

Liberar software após cada uma das múltiplas curtas iterações.

Feedback constante para o cliente.

Princípios chaves

5 – Entregar rapidamente



Lotes pequenos são mais eficientes quando pequenos lotes de trabalho se movem através de um sistema a uma taxa constante..

Princípios chaves

6 – Empoderar a equipe



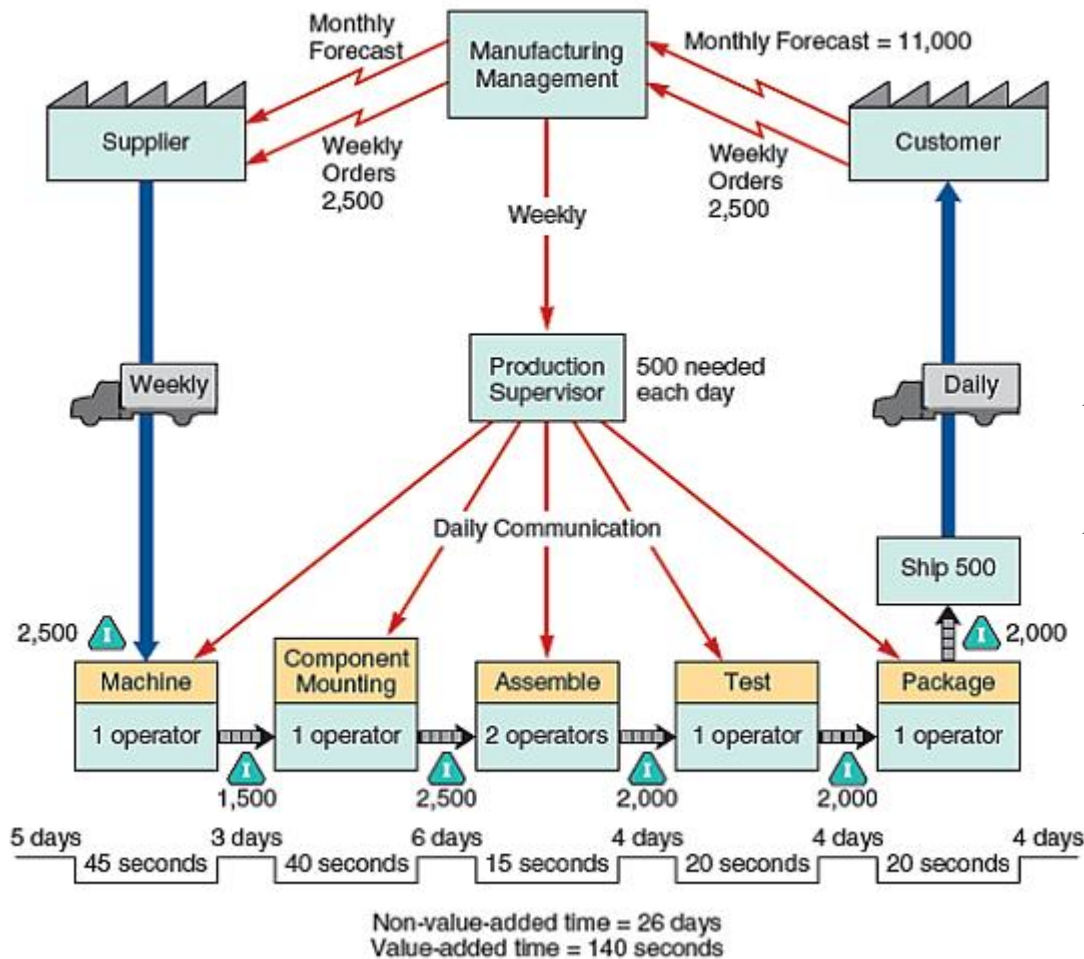
7 – Amplificar a aprendizagem



A aprendizagem é alcançada por meio de feedback progressivo que ocorre entre os membros da equipe e entre a equipe e o cliente.

Assegura que os desenvolvedores compreenderam totalmente os requisitos dos clientes. Isto amplifica a aprendizagem entre a equipe, com os desenvolvedores colaborando de forma mais estreita e compartilhando experiências.

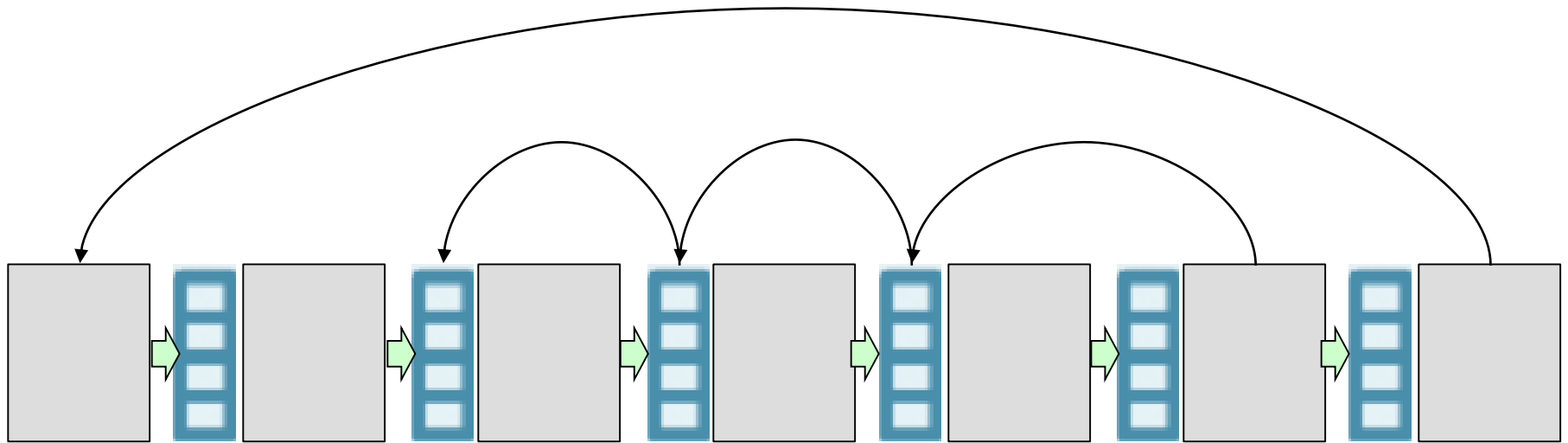
Ferramentas Lean



Value Stream Mapping – Para identificar e eliminar desperdícios, é necessário avaliar o processo de trabalho.

É preciso analisar cada passo do processo produtivo a fim de identificar quais produzem valor para o cliente e quais são considerados desperdícios.

Ferramentas Lean

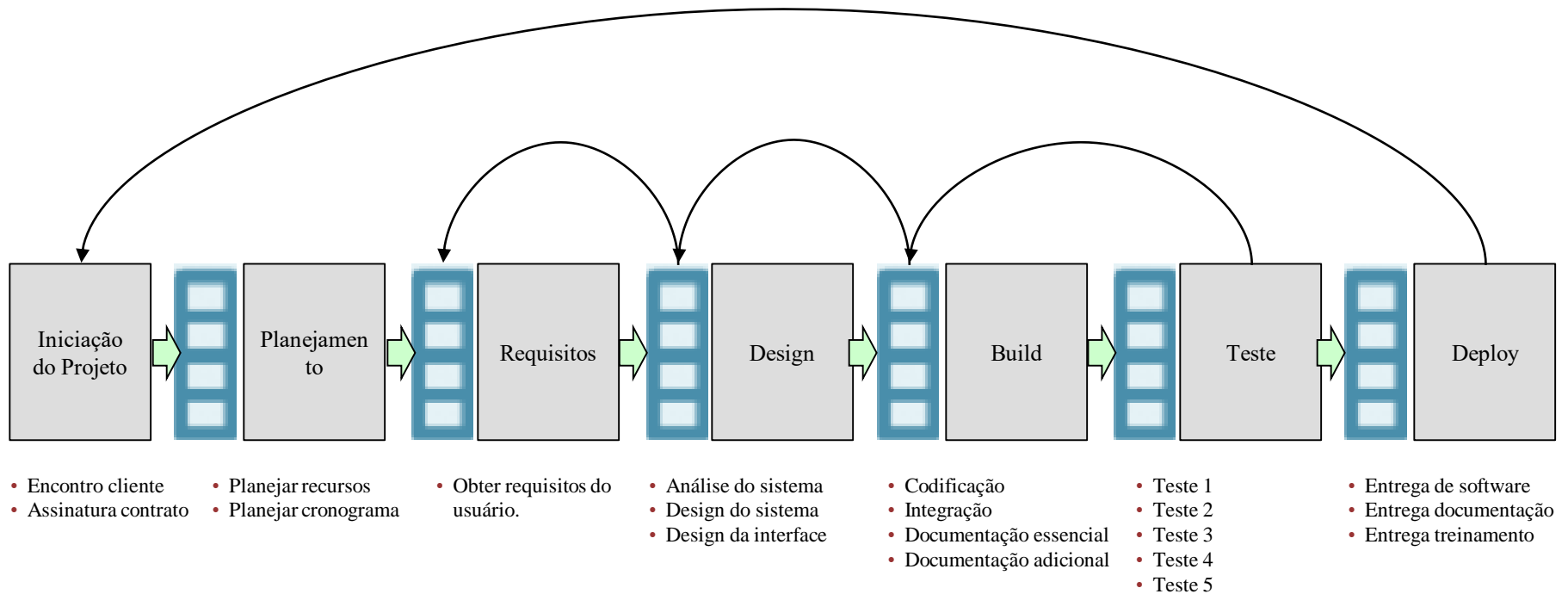


1 – Começar pelo final

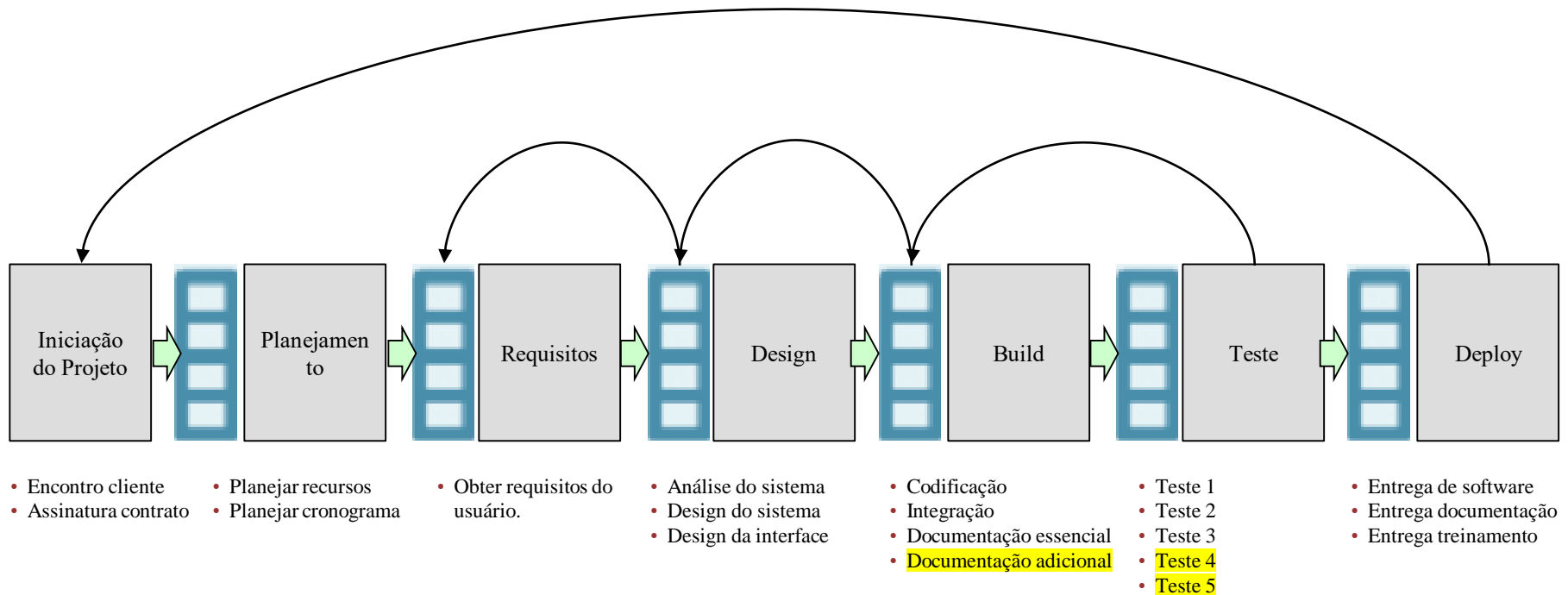
2 – Mapear as etapas do processo

3 – Analisar cada etapa

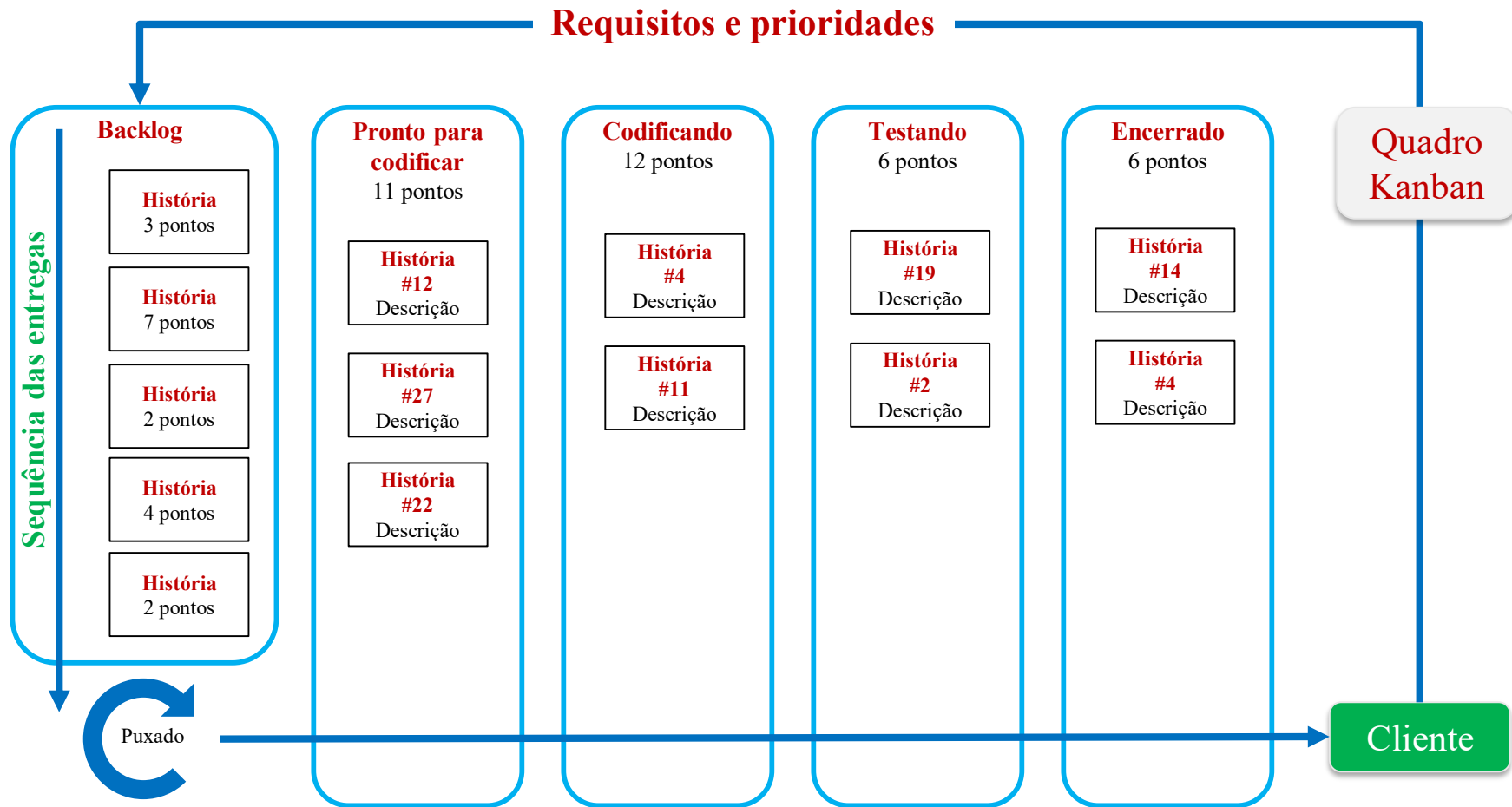
Ferramentas Lean



Ferramentas Lean



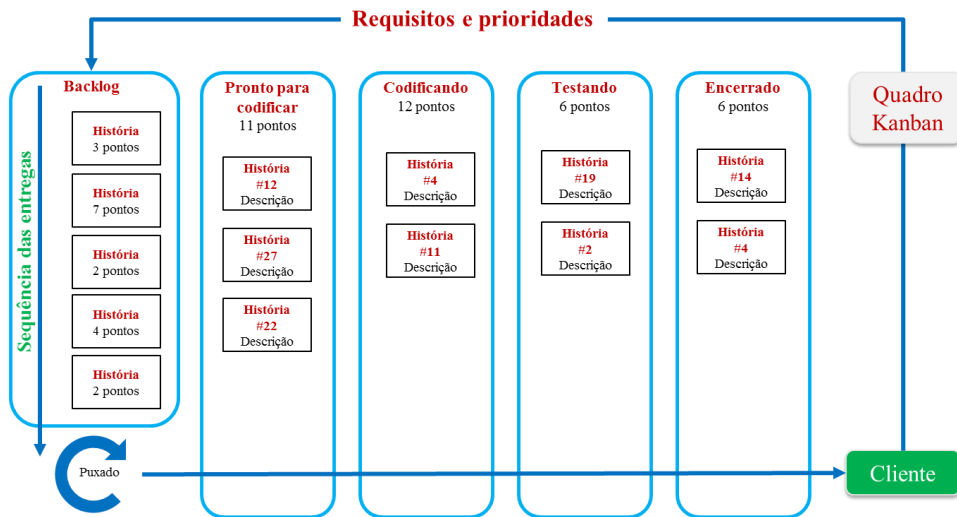
Ferramentas Lean



Ferramentas Lean

Kanban melhora vários aspectos do desenvolvimento.

Eficiência



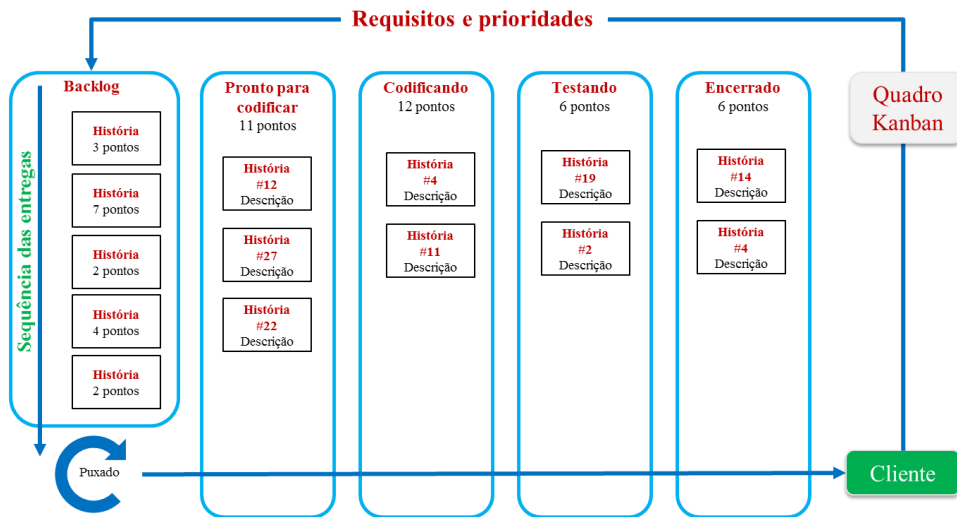
Devido ao sistema “puxado”, Kanban assegura que o trabalho flua suavemente entre diferentes departamentos e indivíduos, com o fluxo de trabalho nunca excedendo a capacidade em nenhum dos pontos.

Isto previne atrasos e otimiza a eficiência.

Ferramentas Lean

Kanban melhora vários aspectos do desenvolvimento.

Comunicação



O quadro Kanban é uma ferramenta altamente visual, traduzindo em facilidade em acompanhar o trabalho pela equipe do projeto.

As reuniões diárias se baseiam em torno do quadro, o que minimiza a necessidade de e-mails e outras ações de comunicação.

Kanban em combinação com a metodologia de gestão de projetos Scrum é conhecida como Sistema Scrumban.

Conclusões

- Lean Development simplifica o desenvolvimento ao encorajar equipes em desenvolver somente as funcionalidades que realmente são necessárias, desencorajando qualquer coisa que não adicione valor ao produto.
- A metodologia é baseada nos princípios de eliminar desperdícios, construir integridade, ver o todo, decidir tardiamente, entregar rapidamente, empoderar a equipe e amplificar a aprendizagem.
- Lean faz uso de das ferramentas: value stream mapping e Kanban.