

FUNDAMENTOS DA ENGENHARIA DE SOFTWARE

ANÁLISE E PROJETO DE SISTEMAS

Prof. Evandro Zatti, M. Eng.

CICLO DE VIDA E DESENVOLVIMENTO DE SOFTWARE

SOFTWARE

“O software de computador é o produto que profissionais de software constroem e dão suporte ao longo do tempo. Ele engloba os programas que executam em computador de qualquer tamanho e arquitetura, o conteúdo que é exibido quando o programa executa, e a informação descritiva tanto na forma física quanto virtual que abrange praticamente qualquer meio eletrônico.”

(PRESSMAN e MAXIM, 2015, p. 1)

CICLO DE VIDA DO SOFTWARE

- O **ciclo de vida** de um software considera suas fases desde a concepção até a morte.
- É muito comum a confusão entre **ciclo de vida** e **ciclo de desenvolvimento** de software, este último também chamado de **processo de software**.

CICLO DE VIDA DO SOFTWARE

- O **ciclo de vida** considera as seguintes fases:
 - ✓ Definição;
 - ✓ Desenvolvimento;
 - ✓ Operação;
 - ✓ Retirada.

DEFINIÇÃO

- A fase de **definição** do software está relacionada às atividades iniciais da concepção:
 - ✓ Modelagem de Processos de Negócios;
 - ✓ Estudo de viabilidade (inclusive custo-benefício);
 - ✓ Análise de sistemas.

DESENVOLVIMENTO

- A fase de **desenvolvimento** do software considera:
 - ✓ Projeto (conceitual; arquitetura; interface; algoritmos e estruturas de dados);
 - ✓ Implementação (codificação, compilação, testes);
 - ✓ Verificação e Validação (garantia de qualidade).

OPERAÇÃO

- A fase de **operação** está relacionada a instalação e uso:
 - ✓ Distribuição;
 - ✓ Instalação e configuração;
 - ✓ Treinamento;
 - ✓ Utilização;
 - ✓ Manutenção.

RETIRADA

- A fase de **retirada** considera que o software não tem mais condição de continuar operando. Não é um processo simples, sendo que o estudo deve considerar processos de reengenharia, substituição gradual até o abandono.

CICLO DE VIDA DE DESENVOLVIMENTO / PROCESSO DE SOFTWARE

“Quando você trabalha para construir um produto ou sistema, é importante seguir uma série de passos pré-definidos – um roteiro que ajuda você a criar um resultado oportuno e de alta qualidade. O roteiro que você segue é chamado de **processo de software**.”

(PRESSMAN e MAXIM, 2015, p. 30)

PROCESSO DE SOFTWARE

- De uma forma genérica, o processo de desenvolvimento de software poderia considerar as seguintes etapas:
 - ✓ Comunicação;
 - ✓ Planejamento;
 - ✓ Modelagem;
 - ✓ Construção;
 - ✓ Implantação.

PROCESSO DE SOFTWARE

- Considerando as práticas de gerenciamento de projetos, existem basicamente dois tipos de modelos de desenvolvimento de software:
 - ✓ Modelos prescritivos;
 - ✓ Modelagem ágil.
- Atualmente, utilizam-se os dois tipos em um mesmo projeto de software, uma vez que eles são complementares.

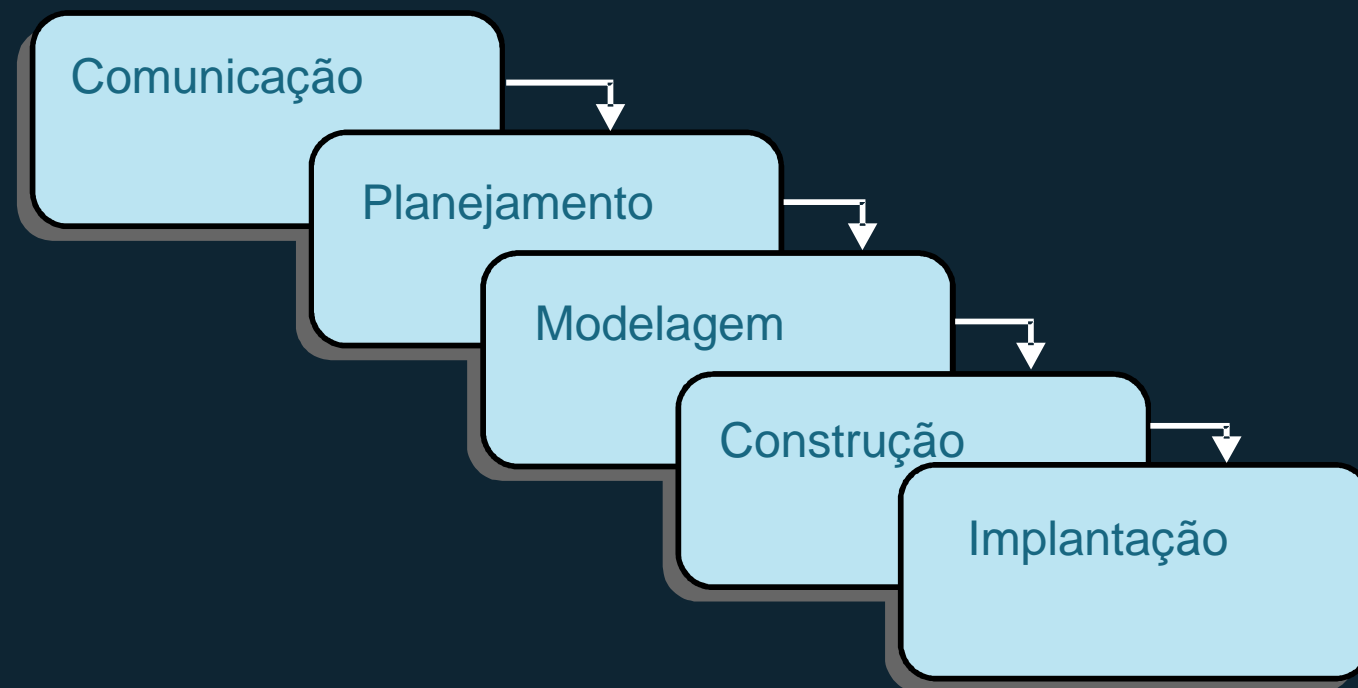
MODELOS PRESCRITIVOS

- Um modelo prescritivo de processos predefine um conjunto de atividades e marcos com geração e produtos para criação de *software*.
- São características dos modelos prescritivos:
 - ✓ Focam em procedimentos prescritivos e os produtos que devem ser criados;
 - ✓ Baseados no paradigma de comando e controle (interessante para gerências);
 - ✓ Minimização da presença do usuário/cliente.

MODELOS PRESCRITIVOS

- São modelos prescritivos:
 - ✓ Cascata (ciclo de vida clássico);
 - ✓ Modelos Incrementais:
 - Incremental;
 - RAD;
 - ✓ Modelos Evolucionários:
 - Prototipagem;
 - Espiral;
 - ✓ Processo Unificado (*Unified Process – UP*):
 - *Rational Unified Process (RUP) → IBM Rational Unified Process (IRUP)*

CASCATA (CICLO DE VIDA CLÁSSICO)

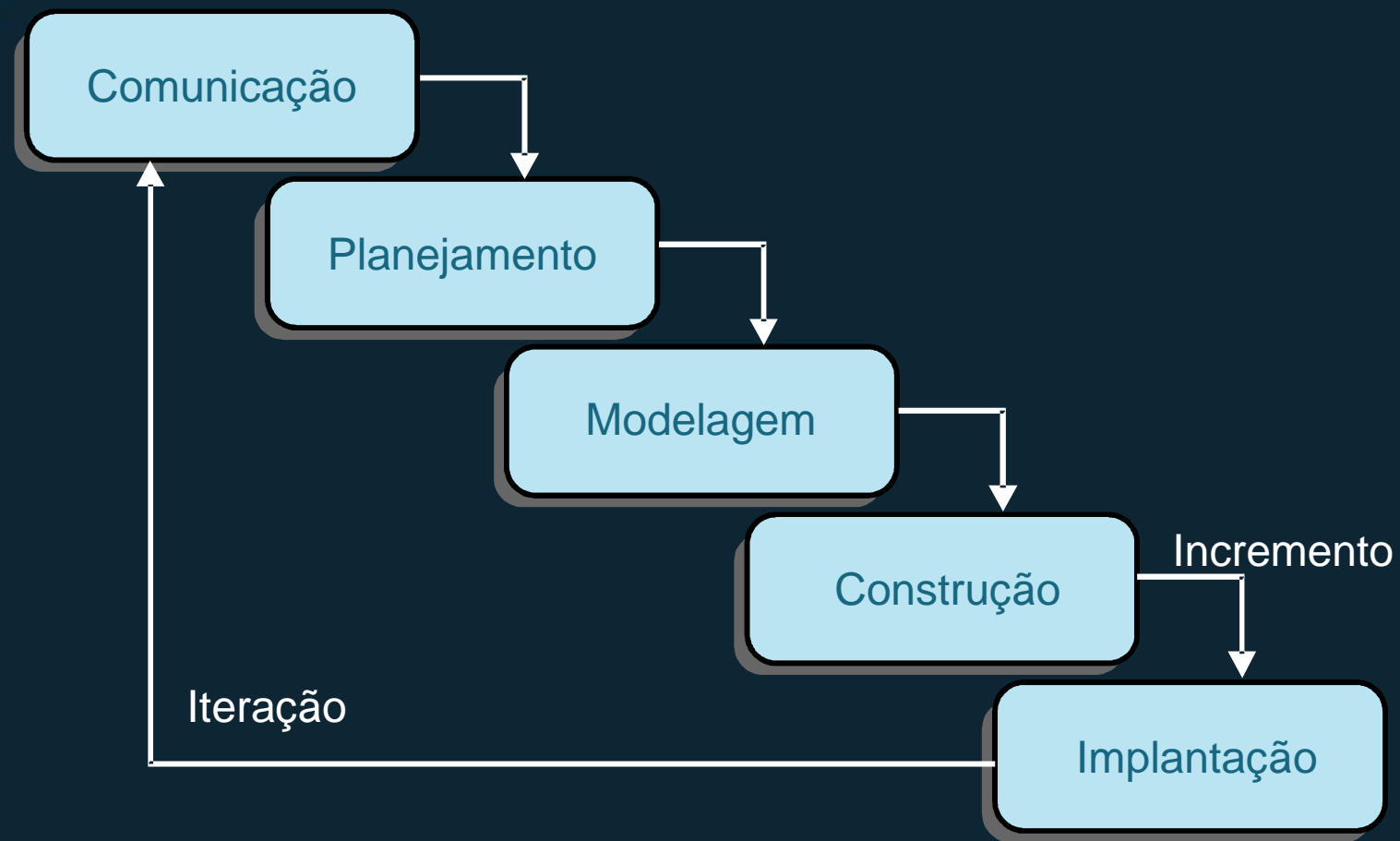


fonte: PRESSMAN e MAXIM, 2015, p. 42

CASCATA (CICLO DE VIDA CLÁSSICO)

- Comunicação
 - ✓ iniciação do projeto e levantamento de requisitos;
- Planejamento
 - ✓ estimativas, cronograma e monitoramento;
- Modelagem
 - ✓ análise e projeto;
- Construção
 - ✓ codificação e testes;
- Implantação
 - ✓ entrega, manutenção e *feedback*.

MODELO INCREMENTAL

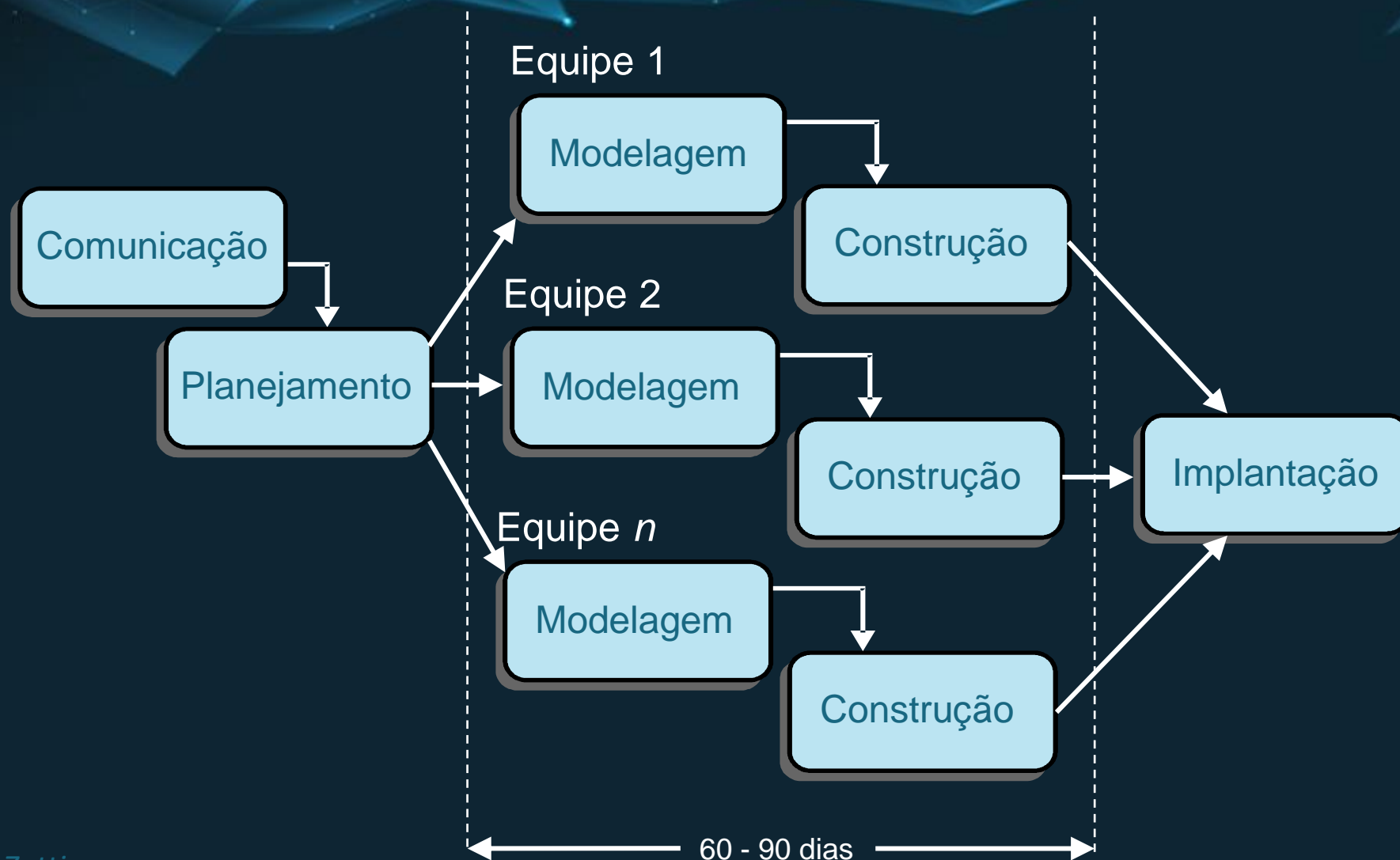


fonte: PRESSMAN e MAXIM, 2015, p. 44

MODELO INCREMENTAL

- Quando um Modelo Incremental é usado, o primeiro incremento frequentemente é chamado de núcleo do produto.
- Isto é, os requisitos básicos são satisfeitos, mas muitas características suplementares deixam de ser elaboradas.
- O núcleo do produto é usado pelo cliente e um plano é desenvolvido para o próximo incremento como resultado do uso e/ou avaliação.

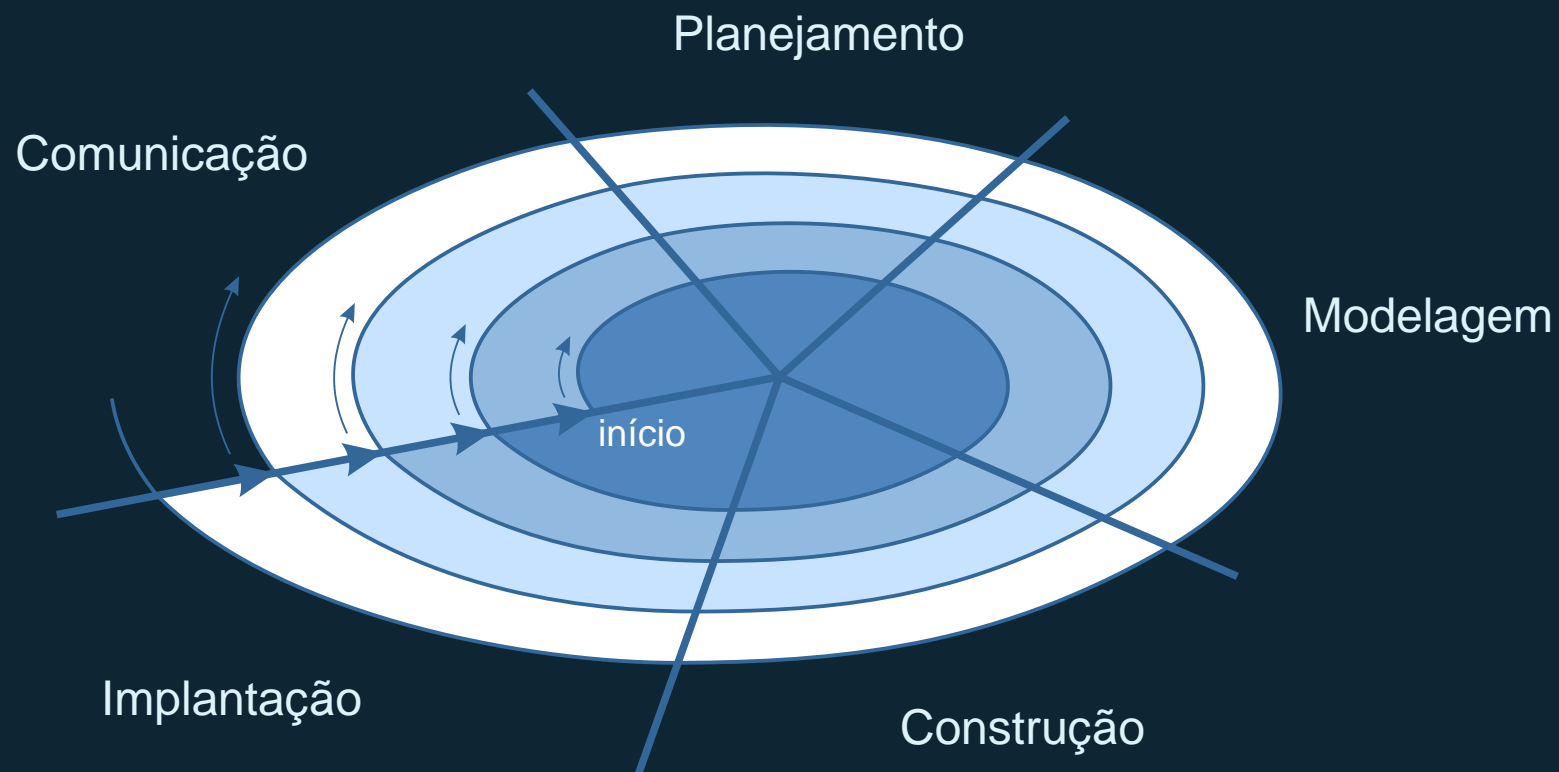
MODELO RAD



MODELO RAD

- O *Rapid Application Development* (RAD) é um modelo de processo de software incremental que enfatiza um ciclo de desenvolvimento curto.
- O Modelo RAD é uma adaptação, de alta velocidade, do modelo em cascata, no qual a agilidade é conseguida com o uso de uma abordagem de construção baseada em componentes.

ESPIRAL



fonte: PRESSMAN e MAXIM, 2015, p. 48

ESPIRAL

- Usando o Modelo Espiral, o software é desenvolvido em uma série de versões evolucionárias;
- Durante as primeiras iterações, as versões podem ser um modelo de papel ou protótipo;
- Durante as últimas iterações, são produzidas versões cada vez mais completas do sistema submetido à engenharia.

PROTOTIPAGEM



fonte: PRESSMAN e MAXIM, 2015, p. 46

PROTOTIPAGEM

- A prototipagem é recomendada quando o cliente define um conjunto de objetivos gerais para o software, mas não identifica detalhadamente requisitos de entrada, processamento ou saída;
- Também sugere-se utilizar quando o desenvolvedor não está seguro sobre a eficiência de um algoritmo, da adaptabilidade de um sistema operacional ou da forma que a interação humano-computador deve assumir.

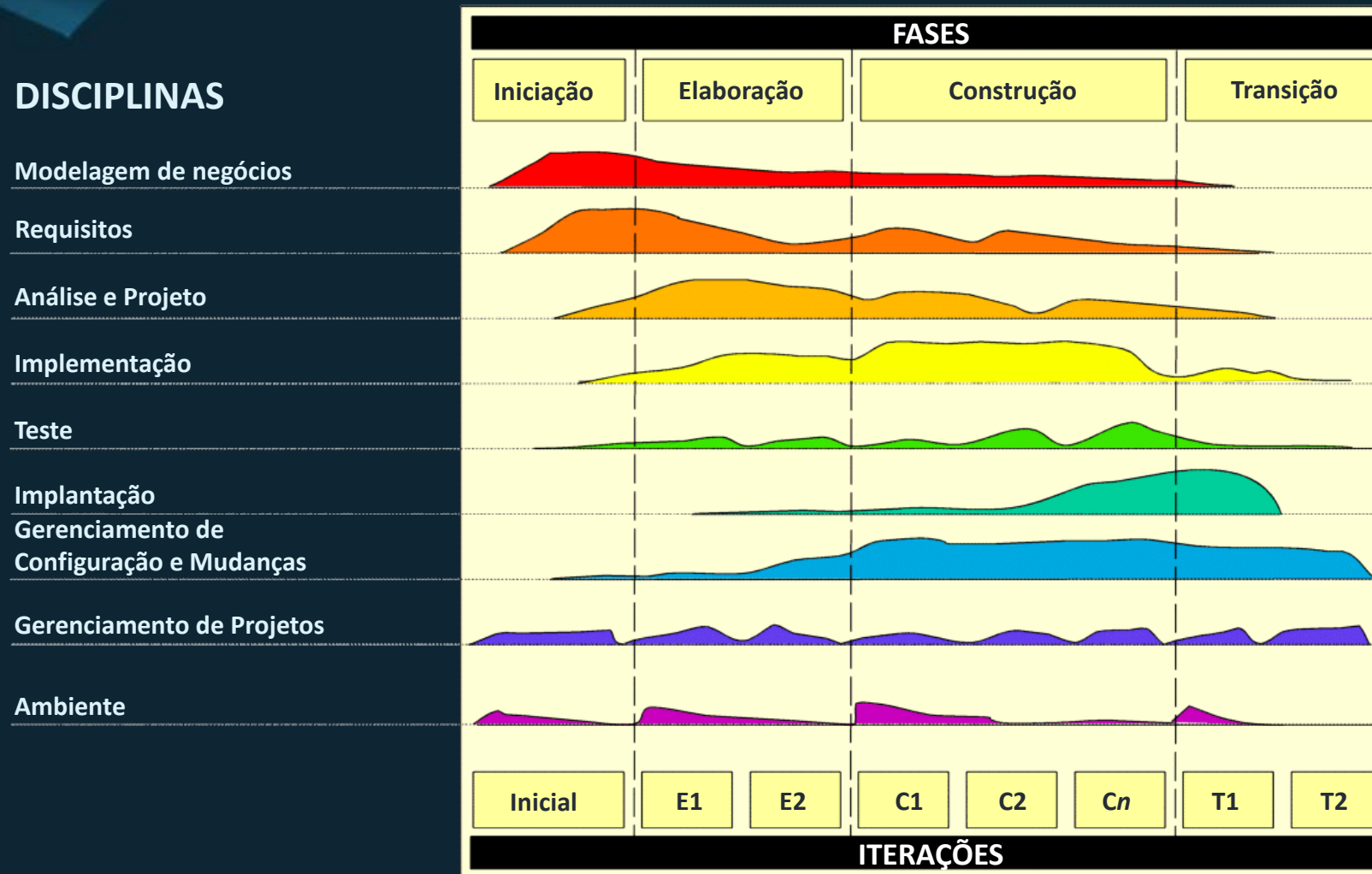
PROCESSO UNIFICADO

- É um processo de software orientado por casos de uso, centrado na arquitetura, iterativo e incremental;
- É uma tentativa de apoiar-se nos melhores recursos e características dos modelos convencionais de processo de software, incluindo muitos dos melhores princípios de desenvolvimento ágil;
- Reconhece a importância da comunicação com o cliente e dos métodos diretos para descrever a visão do cliente de um sistema;
- Sugere um fluxo de processo iterativo e incremental.

PROCESSO UNIFICADO

- Fases do Processo Unificado:
 - ✓ **Iniciação:** abrange atividades de comunicação com o cliente e de planejamento;
 - ✓ **Elaboração:** inclui a comunicação com o cliente e atividades de modelagem do processo genérico;
 - ✓ **Construção:** usando o modelo arquitetural como entrada, desenvolve ou adquire os componentes de software que vão tornar cada caso de uso operacional;
 - ✓ **Transição:** abrange os últimos estágios da atividade genérica de construção e a primeira parte da atividade genérica de implantação.

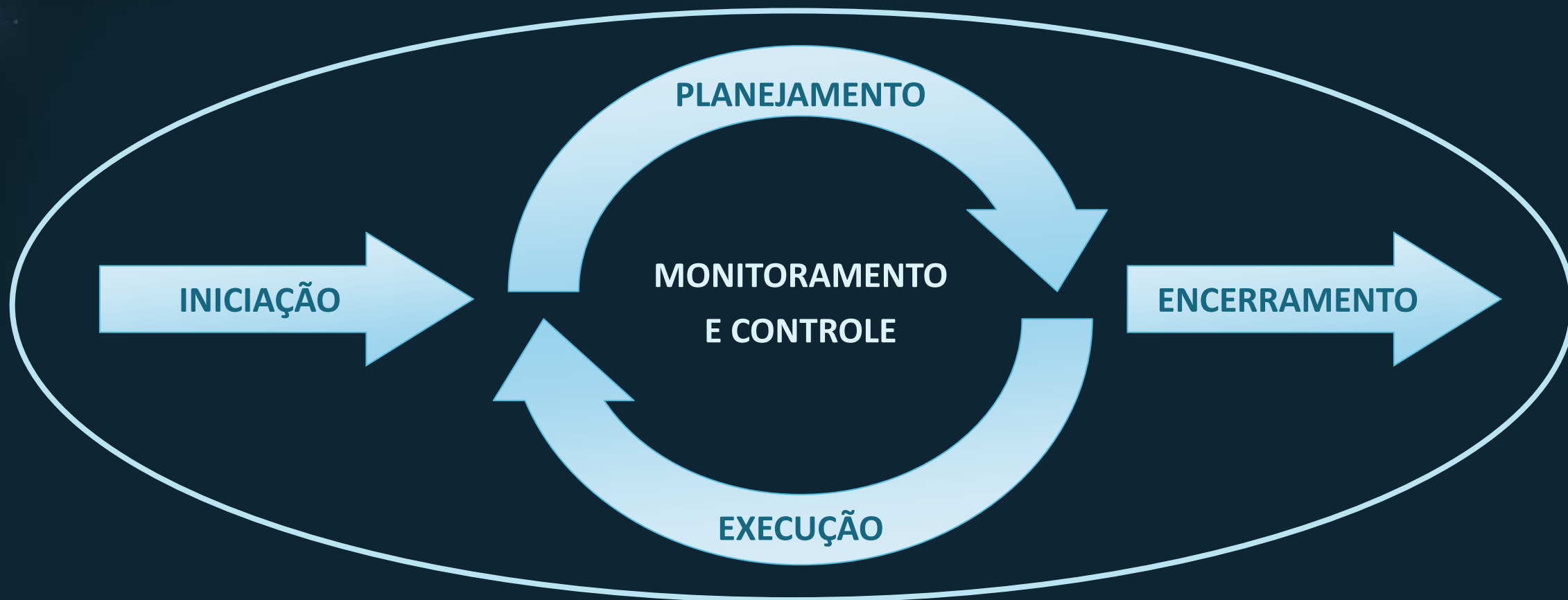
PROCESSO UNIFICADO



fonte:
KRUCHTEN, 2003
(adaptado)

GESTÃO DE PROJETOS

CICLO DE VIDA

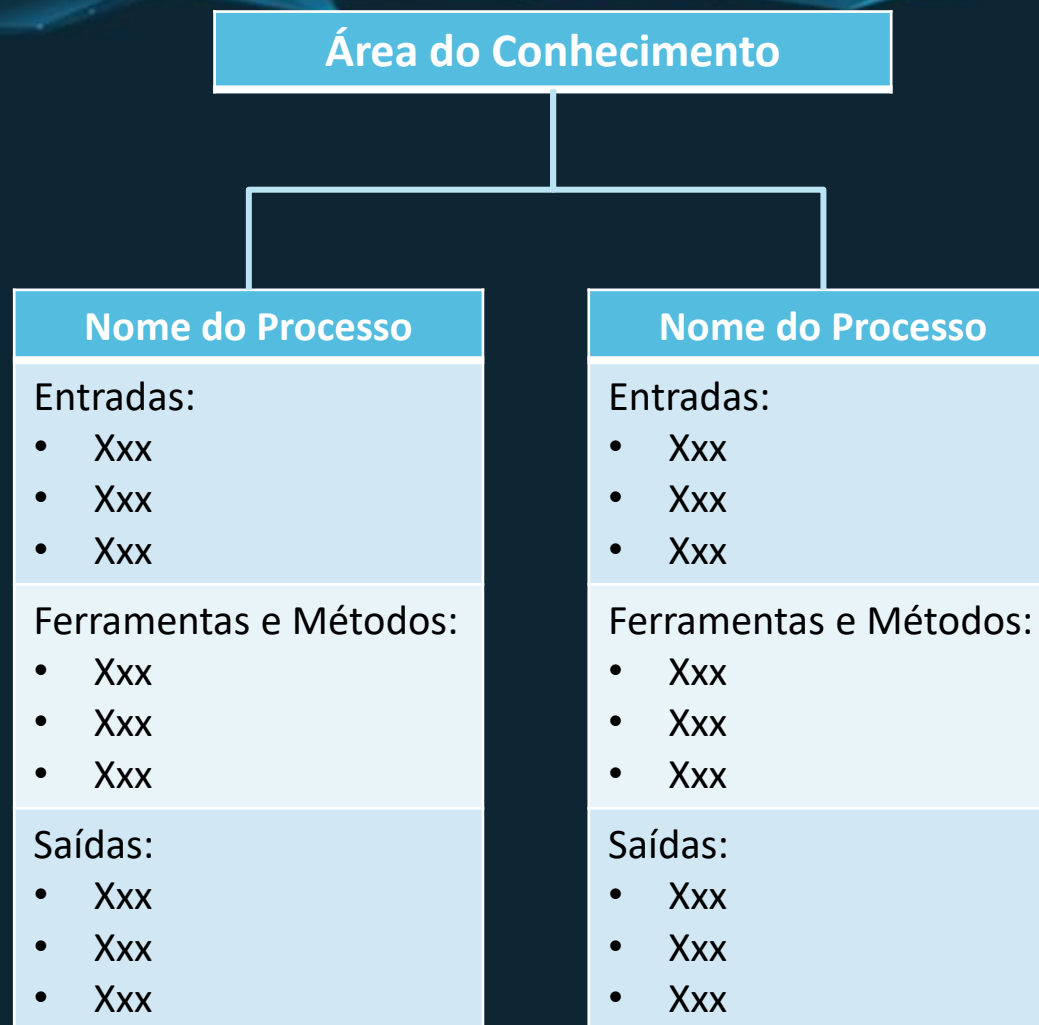


ÁREAS DE CONHECIMENTO



fonte: PMI, 2008

PROCESSOS



fonte: PMI, 2008

ÁREAS DE CONHECIMENTO X PROCESSOS

Área de Conhecimento	Iniciação	Planejamento	Execução	Monitoramento e Controle	Encerramento
Integração	1.1. Desenvolver o termo de abertura do projeto	1.2. Desenvolver o plano de gerenciamento do projeto	1.3. Orientar e gerenciar o trabalho do projeto	1.4. Monitorar e controlar o trabalho do projeto 1.5. Realizar o controle integrado de mudanças	1.6. Encerrar o projeto ou fase
Escopo		2.1. Planejar o Gerenciamento do Escopo 2.2. Coletar os requisitos 2.3. Definir o escopo 2.4. Criar a EAP		2.5. Validar o escopo 2.6. Controlar o escopo	
Tempo		3.1. Planejar o gerenciamento do Cronograma 3.2. Definir as atividades 3.3. Sequenciar atividades 3.4. Estimar os recursos das atividades 3.5. Estimar as durações das atividades 3.6. Desenvolver o cronograma		3.7. Controlar o cronograma	
Custos		4.1. Planejar o gerenciamento dos Custos 4.2. Estimar custos 4.3. Determinar o orçamento		4.4. Controlar os custos	
Qualidade		5.1. Planejar o gerenciamento da qualidade	5.2. Realizar a garantia de qualidade	5.3. Controlar a qualidade	

ÁREAS DE CONHECIMENTO X PROCESSOS

Área de Conhecimento	Iniciação	Planejamento	Execução	Monitoramento e Controle	Encerramento
Recursos Humanos		6.1. Planejar o gerenciamento dos recursos humanos	6.2. Mobilizar a equipe do projeto 6.3. Desenvolver a equipe do projeto 6.4. Gerenciar a equipe do projeto		
Comunicações		7.1 Planejar o gerenciamento das comunicações	7.2. Gerenciar as comunicações	7.3. Controlar as comunicações	
Riscos		8.1. Planejar o gerenciamento dos riscos 8.2. Identificar os riscos 8.3. Realizar a análise qualitativa dos riscos 8.4. Realizar a análise quantitativa dos riscos 8.5. Planejar as respostas aos riscos		8.6. Controlar os riscos	
Aquisição		9.1. Planejar o gerenciamento das aquisições	9.2. Conduzir as aquisições	9.3. Controlar as aquisições	9.4. Encerrar as aquisições
Partes interessadas	10.1. Identificar partes interessadas	10.2. Planejar o gerenciamento das partes interessadas	10.3. Gerenciar o envolvimento das partes interessadas	10.4. Controlar o envolvimento das partes interessadas	

fonte: PMI, 2008

RESTRIÇÃO TRIPLA



CUSTO

TEMPO



QUALIDADE

ESCOPO



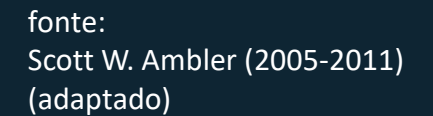
FRAMEWORKS ÁGEIS

MANIFESTO ÁGIL

- Um grupo inicial de 17 metodologistas formou a *Agile Software Development Alliance* (www.agilealliance.org) em fevereiro de 2001.
- Este grupo definiu o que se chama hoje de Manifesto Ágil, que possui um conjunto de valores, princípios e práticas que definem critérios para os processos de desenvolvimento ágil de software: Modelagem Ágil.

MODELAGEM ÁGIL

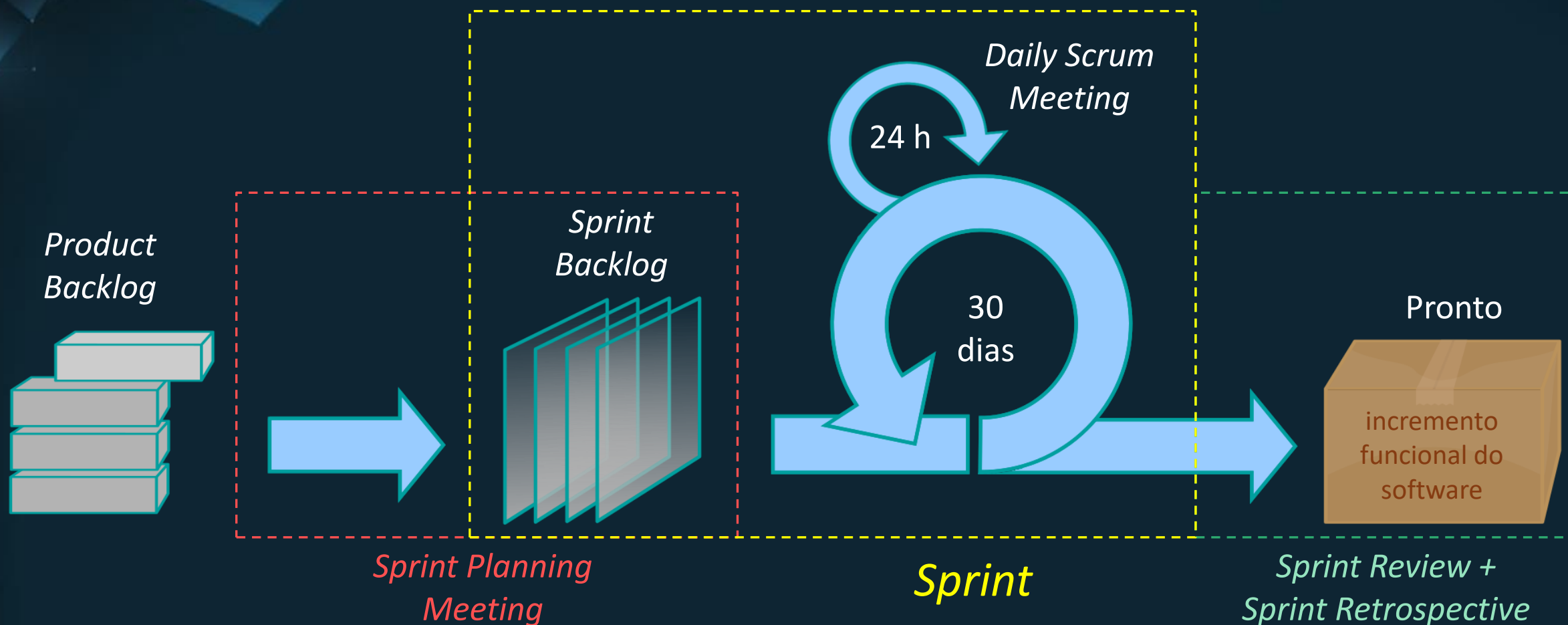
- A Modelagem Ágil (MA) é utilizada para modelagem e documentação eficazes de sistemas baseados em software;
- É baseada em um conjunto de valores, princípios e práticas;
- Não é um processo prescritivo;
- Não define procedimentos detalhados de como criar um determinado tipo de modelo, e sim fornece conselhos sobre como ser um modelador eficiente.



SCRUM

- O Scrum é considerado um framework objetivo, com **papéis**, **eventos** e **artefatos** bem definidos, apoiado por algumas ferramentas.
- O framework Scrum define basicamente três papéis:
 - ✓ *Product Owner*: dono do produto que está sendo desenvolvido;
 - ✓ *Development Team*: equipe responsável pela entrega do produto;
 - ✓ *Scrum Master*: responsável pela condução da equipe na atuação sob o framework.

VISÃO GERAL / CICLO DE VIDA DO SCRUM



KANBAN

- Kanban é uma palavra japonesa que significa **registro** ou **sinalização**;
- Ligado aos conceitos de produção *lean* e *just in time*, nos quais a produção acontece sob demanda, sem geração de estoque.
- Nos cartões são colocados indicativos do estágio que se encontra cada processo. Ex.: “para executar”, “em andamento”, “finalizado”;
- Os cartões são fixados no *kanban board* (quadro kanban).

KANBAN E SOFTWARE

Backlog	Análise		Desenvolvimento		Teste		Pronto
<div>Imprimir Recibo</div> <div>Reembolso</div>	<div>Esperando análise</div> <div>Adicionar taxa de vendas local</div>	<div>Em análise</div> <div>Fornecer desconto para compradores recorrentes</div>	<div>Esperando desenv.</div> <div>Comprar item extra opcional</div>	<div>Em desenv.</div> <div>Remover item do carrinho</div>	<div>Esperando teste</div> <div>Se cadastrar no website</div> <div>Pagar por acesso ao website</div>	<div>Em teste</div> <div>Gerar recibo</div>	<div>Leitura de código</div> <div>Validação de preço</div>

fonte: IIBA, 2010 (adaptado)

KANBAN E SOFTWARE

- Identificação de gargalo:

Backlog	Análise		Desenvolvimento		Teste		Pronto
<div>Imprimir Recibo</div> <div>Reembolso</div>	<div>Esperando análise</div> <div>Adicionar taxa de vendas local</div> <div>Fornecer desconto para compradores recorrentes</div>	<div>Em análise</div> <div>Comprar item extra opcional</div> <div>Remover item do carrinho</div> <div>Se cadastrar no website</div>	<div>Esperando desenv.</div>	<div>Em desenv.</div>	<div>Esperando teste</div> <div>Pagar por acesso ao website</div>	<div>Em teste</div> <div>Gerar recibo</div>	<div>Leitura de código</div> <div>Validação de preço</div>

fonte: IIBA, 2010 (adaptado)

DESIGN THINKING

“O Design Thinking é o equilíbrio entre negócio e arte, estrutura e caos, intuição e lógica, conceito e execução, ludicidade e formalidade, controle e empoderamento.”

Idris Mootee

- ✓ Decisões centradas no ser humano;
- ✓ Questionar as questões;
- ✓ Construir para pensar;
- ✓ Iterar.

PROCESSO DO DESIGN THINKING

- 1. Entendimento:** momento de divergência do pensamento
 - ✓ Ferramenta 2x2 de entendimento; desconstrução do desafio.
- 2. Observação:** momento de convergência de olhar e empatia
 - ✓ Triangulação das pesquisas: entrevistas, olhar etnográfico, design participativo.
- 3. Ponto de vista:** momento de convergência de pensamento;
 - ✓ Mapa de empatia ou persona.

PROCESSO DO DESIGN THINKING

4. Ideação: momento de divergência de pensamento

- ✓ Brainstorming.

5. Prototipagem: gerar mais ideias

- ✓ Protótipos; *storyboards*; maquetes; vídeos.

6. Teste: procurar erros

7. Iteração: feedback

- ✓ Pontos positivos; pontos negativos; ideias novas; questões.

LEAN INCEPTION

- O Lean Inception é uma capacitação de uma equipe feita para alinhar um grupo de pessoas, no qual colocam em prática um ciclo de atividades para que seja possível a construção de um Produto Mínimo Viável (MVP).

Paulo Caroli, 2018

PROCESSO DO LEAN INCEPTION

- **Segunda-Feira:** reunião inicial (*kick-off*) com os *stakeholders*; compreende-se os objetivos do produto;
- **Terça-Feira:** identificam-se as **personas**; realiza-se um *brainstorming*; documentam-se as ideias e os processos;
- **Quarta-Feira:** descobrir e solucionar problemas encontrados no *brainstorming*; **jornada** para descrever uma sequência para alcançar o objetivo;

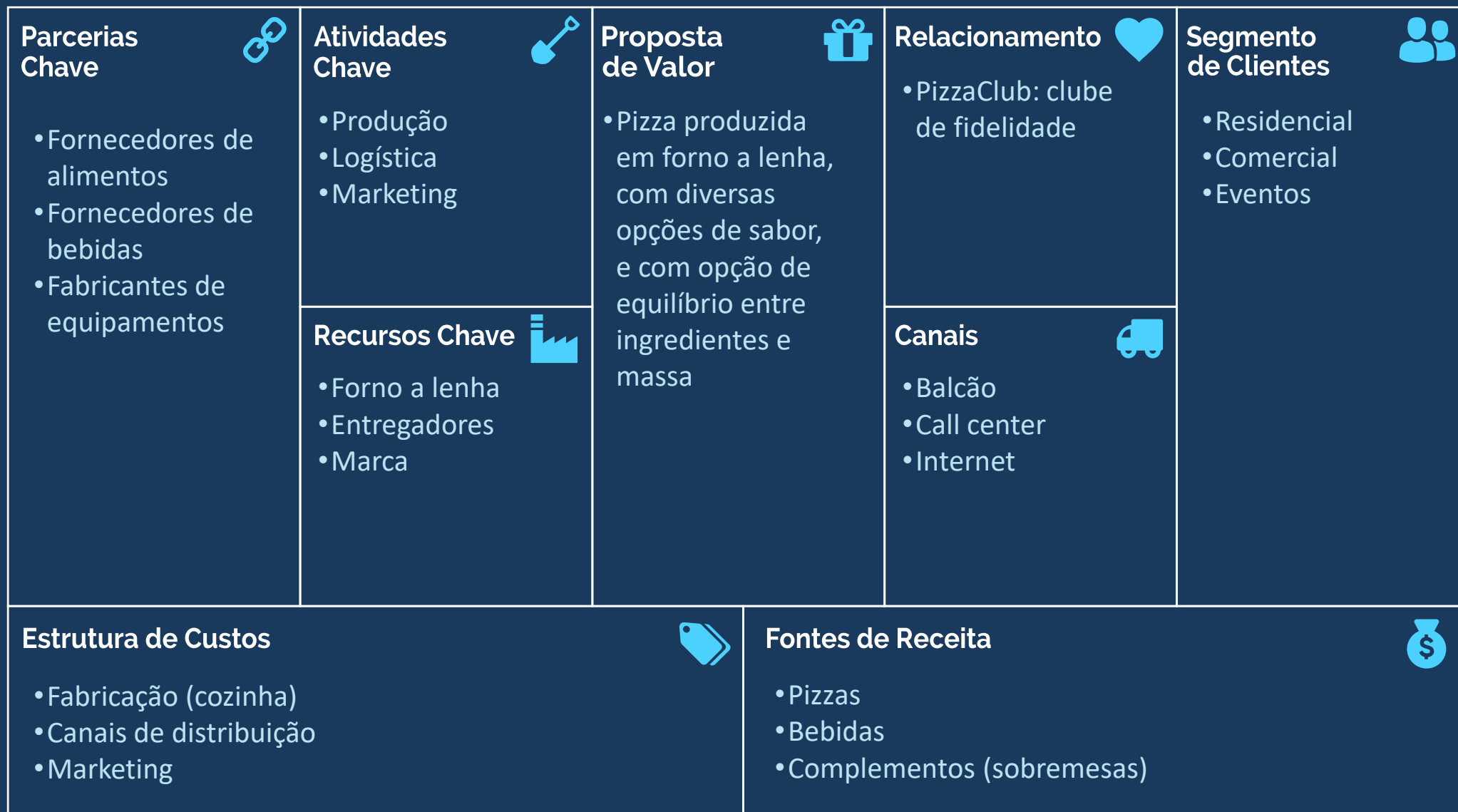
LEAN INCEPTION

- ✓ **Quinta-Feira:** elabora-se o **plano de entrega**; a proposta é feita baseada no MVP, por meio de um *canvas*;
- ✓ **Sexta-Feira:** apresenta-se o modelo *canvas* do MVP para os *stakeholders* e verifica-se se a proposta está alinhada com as ideias do cliente.

BUSINESS CANVAS



BUSINESS CANVAS – EXEMPLO



ATIVIDADE PRÁTICA

- Concepção do projeto prático avaliativo:
 - ✓ Formar equipes (sugerido: entre 3 e 5 integrantes);
 - ✓ Idealizar um sistema comercial (que envolva cadastros em banco de dados);
 - ✓ Registrar o processo de ideação.



REFERÊNCIAS

- PMI. *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*. 5th ed. 2008.
- PRESSMAN, R. W, MAXIM B. R. *Software Engineering - A Practitioner's Approach*. 8th ed. New York: McGraw-Hill, 2015.
- KOLB, J. **Compartilhando**.
 - ✓ Disponível em: <http://www.jkolb.com.br>. Acesso em 20/01/2018.
- KRUCHTEN, P. *Rational Unified Process Made Easy*. Boston: Addison-Wesley Professional, 2003.