CICLO DE VIDA E DESENVOLVIMENTO DE SOFTWARE

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Análise e Desenvolvimento de Sistemas

Prof. *Evandro Zatti,* *M. Eng.*

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.

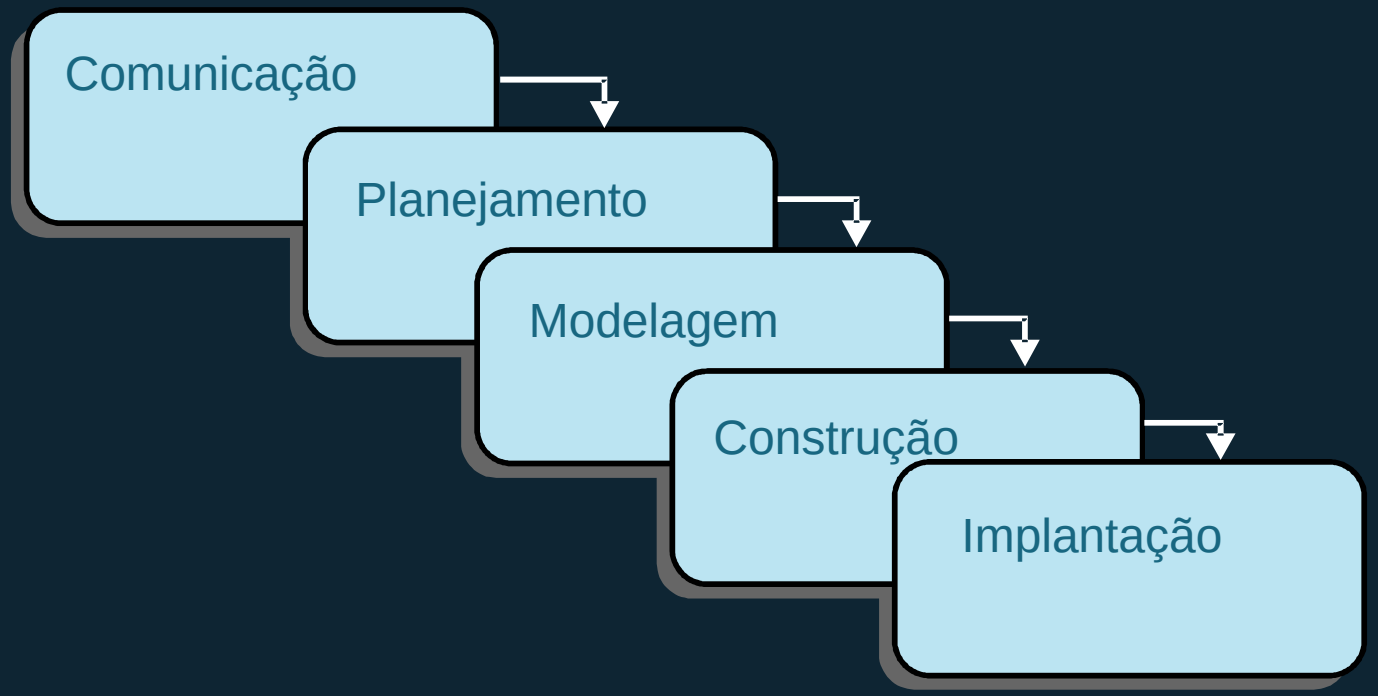
MODELOS PRESCRITIVOS

- 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)

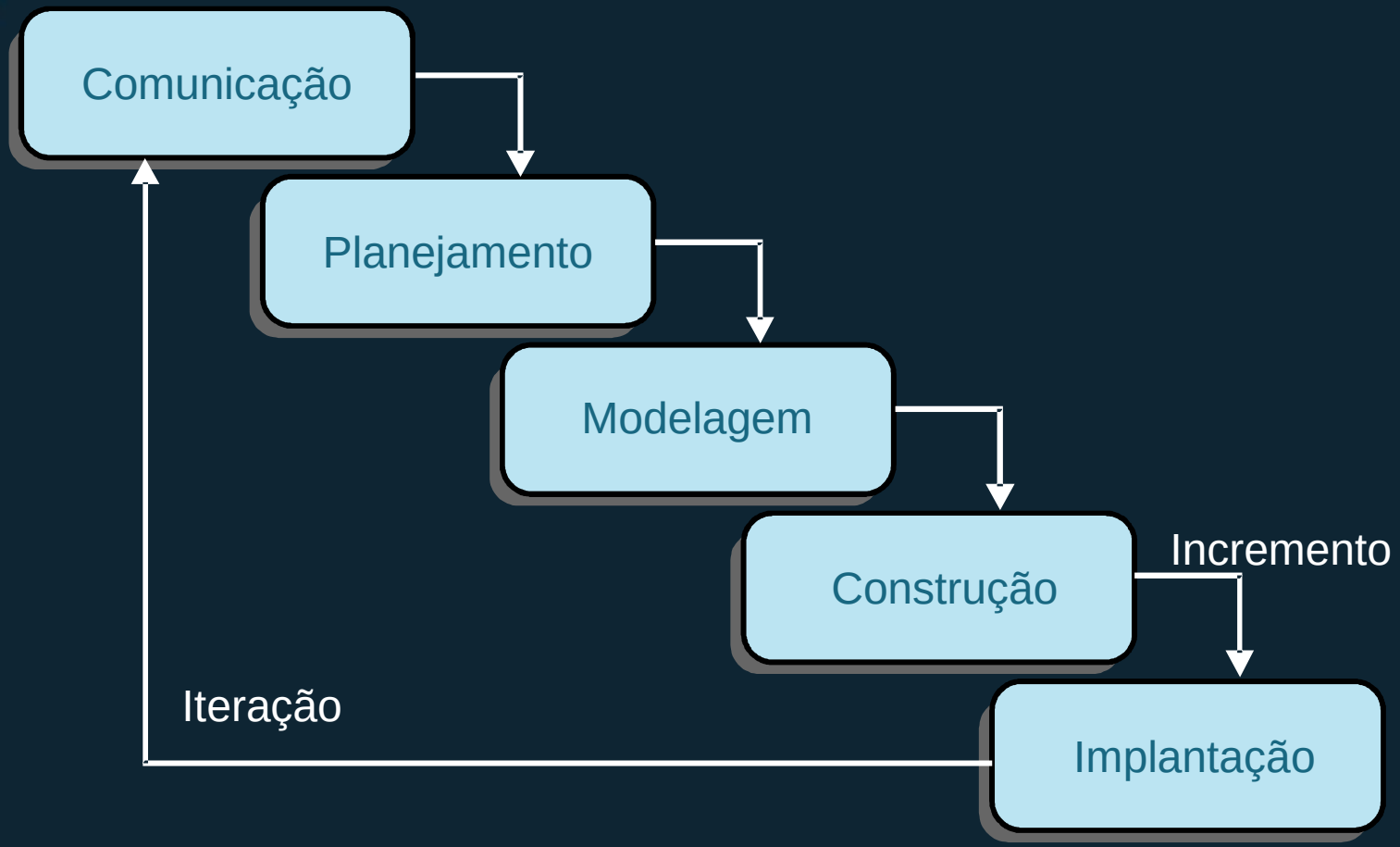


fonte: PRESSMAN e MAXIM, 2015, p. 42

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

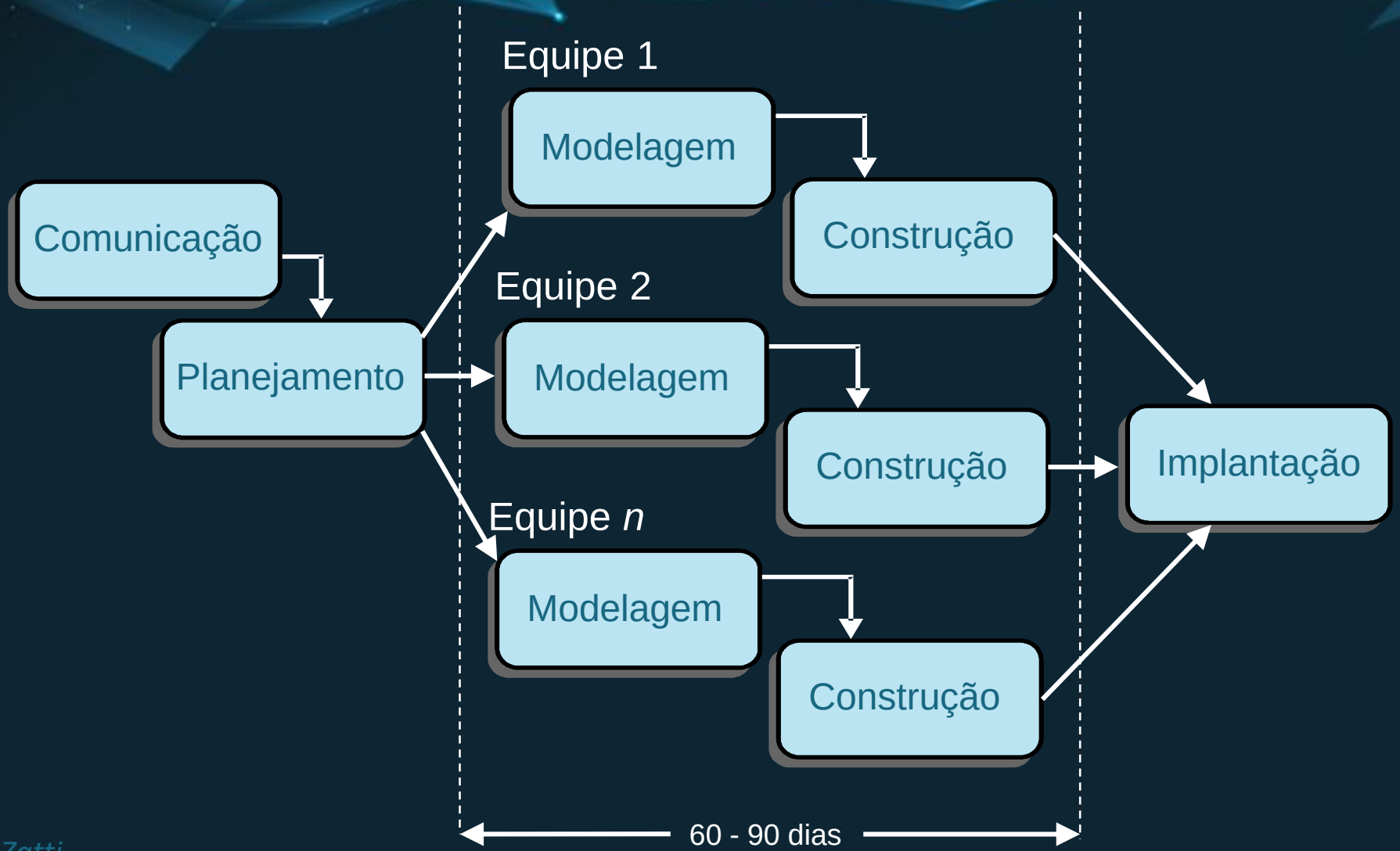


fonte: PRESSMAN e MAXIM, 2015, p. 44

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


Equipe 1
Modelagem
Comunicação
Construção
Equipe 2
Planejamento
Modelagem
Construção
Implantação
Equipe *n*
Modelagem
Construção
60 - 90 dias

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.

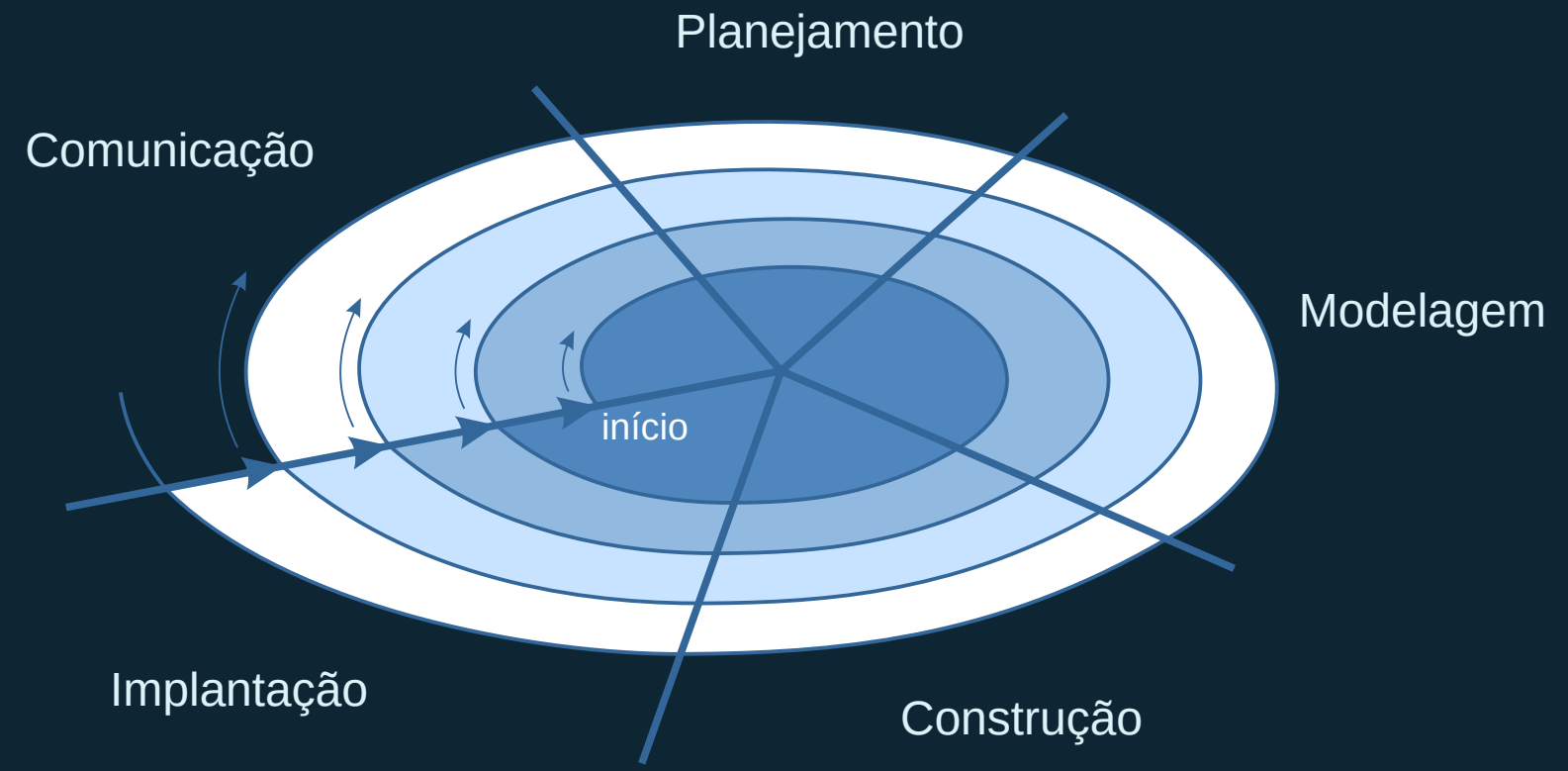
PROTOTIPAGEM

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.

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.

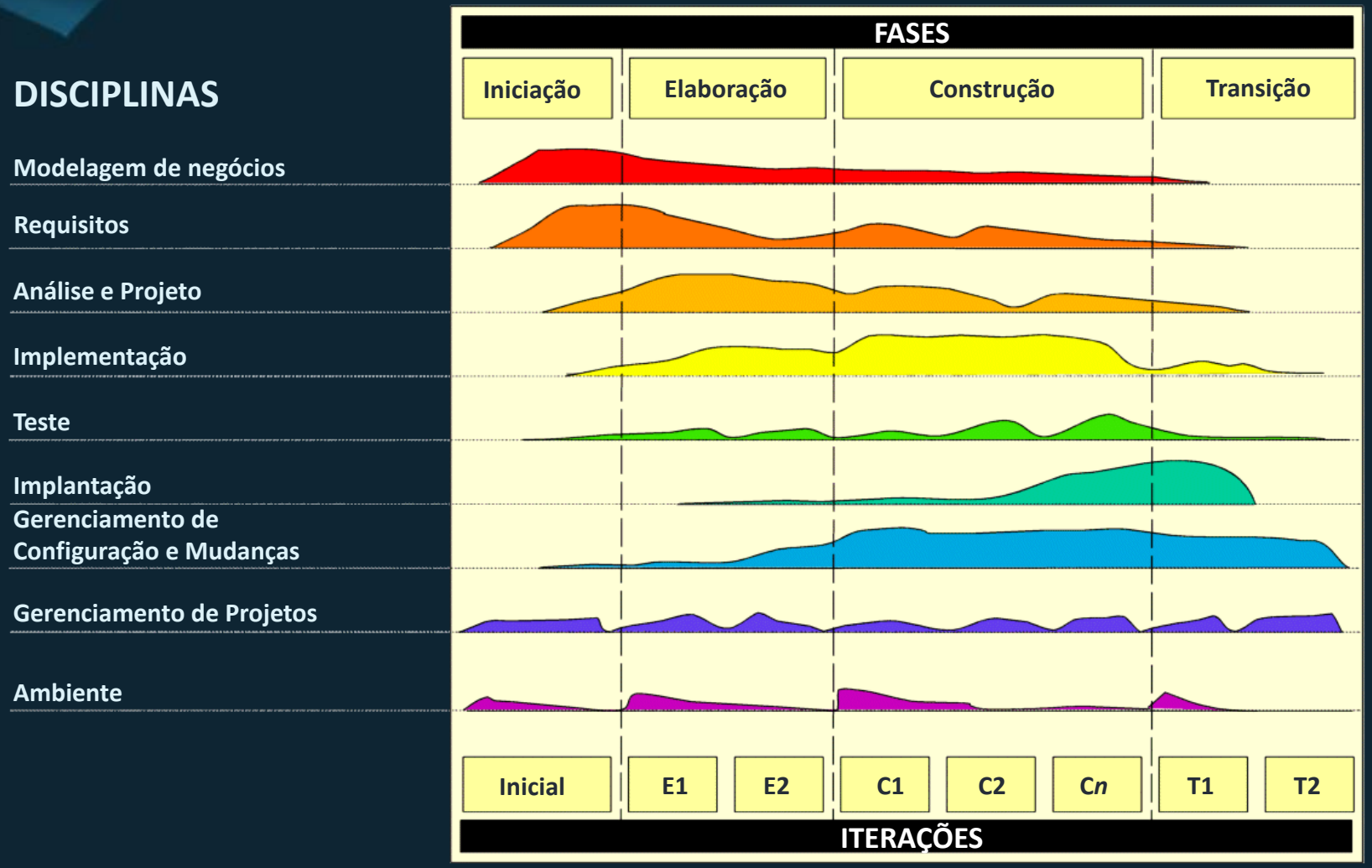
PROCESSO UNIFICADO

- 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)

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 princípios que definem critérios para os processos de desenvolvimento ágil de software: Modelagem Ágil.

VALORES DO MANIFESTO ÁGIL

- Indivíduos e interações valem mais que processos e ferramentas;
- Um software funcionando vale mais que documentação extensa;
- A colaboração do cliente vale mais que a negociação de contrato;
- Responder a mudanças vale mais que seguir um plano.

12 PRINCÍPIOS DO MANIFESTO ÁGIL

1. Satisfazer ao cliente mediante entregas de software de valor em tempo hábil e continuamente.
2. Receber bem mudanças de requisitos, mesmo em uma fase mais avançada de desenvolvimento. Os processos ágeis direcionam as mudanças para obter vantagens competitivas para o cliente.

12 PRINCÍPIOS DO MANIFESTO ÁGIL

3. Entregar software em funcionamento com frequência de algumas semanas a alguns meses, de preferencia na menor escala de tempo.
4. As equipes de negócios e de desenvolvimento devem trabalhar juntas diariamente e durante todo o projeto.
5. Construa projetos ao redor de indivíduos motivados. Dê-lhes o ambiente e o apoio de que eles precisam e confie neles para realizar o trabalho.

12 PRINCÍPIOS DO MANIFESTO ÁGIL

6. O método mais eficiente para levar informações para a equipe de desenvolvimento e fazê-las circular é a conversa cara a cara.
7. Ter o software funcionando é a principal medida de progresso.
8. Processos ágeis promovem o desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários deveriam ser capazes de manter um ritmo constante indefinidamente.

12 PRINCÍPIOS DO MANIFESTO ÁGIL

9. Atenção contínua à excelência técnica e a um bom projeto aumentam a agilidade.
10. Simplicidade – a arte de maximizar a quantidade de trabalho não realizado – é essencial.
11. As melhores arquiteturas, requisitos e projetos provêm de equipes organizadas.
12. Em intervalos regulares, a equipe deve refletir sobre como se torna mais eficaz e então se ajustar e adaptar seu comportamento.

MODELAGEM ÁGIL

- A Modelagem Ágil (MA) é uma metodologia baseada na prática para modelagem e documentação eficazes de sistemas baseados em software;
- É baseada em um conjunto de práticas (princípios e valores) para profissionais de software aplicarem em seu dia a dia;
- 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.

OBJETIVOS DA MODELAGEM ÁGIL

- Definir e mostrar como colocar em prática um conjunto e valores, princípios e práticas relativas uma modelagem eficaz e leve;
- Lidar com a questão de como aplicar técnicas de modelagem em projetos de software adotando uma perspectiva ágil;
- Discutir como você pode melhorar seus atividades de modelagem adotando uma perspectiva “quase ágil” para o desenvolvimento de software e equipes de projeto.

PAPÉIS DA MODELAGEM ÁGIL

- Modelador – É qualquer pessoa que siga a metodologia MA, aplicando as práticas desta com seus princípios e valores.
- Desenvolvedor – É quem adota uma perspectiva ágil de desenvolvimento de software.

O QUE É OU NÃO MODELAGEM ÁGIL?

- A MA é uma atitude, não um processo prescritivo;
- A MA é um suplemento dos métodos pré-existentes não uma metodologia completa;
- A MA é complementar aos processos de modelagem;
- A MA é uma maneira de trabalhar em conjunto de modo eficaz e alcançar os objetivos dos clientes do projeto;
- A MA é eficaz e trata de eficácia;
- A MA é algo que funciona na prática, não é uma teoria acadêmica;

O QUE É OU NÃO MODELAGEM ÁGIL?

- A MA não é uma bala de prata;
- A MA foi feita para o desenvolvedor médio, mas não é uma substituição de pessoas competentes;
- A MA não é um ataque à documentação;
- A MA não é um ataque às ferramentas CASE.

VALORES DA MODELAGEM ÁGIL

- São valores da Modelagem Ágil:
  - ✓ Comunicação;
  - ✓ Simplicidade;
  - ✓ Retorno;
  - ✓ Coragem;
  - ✓ Humildade.
- Entenda os valores a seguir:

VALOR: COMUNICAÇÃO

- É uma via de duas mãos, ambas fornecem e obtém informações como resultado;
- Comunicação eficaz entre todos os envolvidos (desenvolvedores e cliente);
- Ilustre sua comunicação, faça-se entender, desenhe se for preciso...

VALOR: SIMPLICIDADE

- Não complique, use a Regra KISS (*Keep It Simple, Stupid*) – Mantenha Isto Simples, Estúpido.
- Não inclua complicações do tipo:
  - ✓ Padrões complexos demais;
  - ✓ Criar arquiteturas em excesso para que o sistema suporte possíveis requisitos futuros;
  - ✓ Desenvolver infraestrutura complexa;
  - ✓ Não crie cenários com base em suposições.

VALOR: RETORNO

- A única forma de verificar se seu modelo está correto é obtendo *feedback*.

- Desenvolva o modelo em equipe;
- Revise o modelo com seu público-alvo;
- Implemente o modelo;
- Teste a aceitação.

- Revisões informais e revisões formais.

VALOR: CORAGEM

- Utilizar MA é um desafio, pois é uma novidade para a maioria das pessoas;
- Acredite em pessoas e em si mesmo;
- Coragem para manter a estratégia nos momentos difíceis;
- Coragem para reconhecer falhas e que comete erros;
- Coragem para confiar que poderá superar os problemas que surgirão no futuro.

VALOR: HUMILDADE

- Reconhecer que não sabe tudo.
- Humildade para respeitar as pessoas que trabalham com você.
- Ter consciência que elas podem ser melhores que você em alguns aspectos (um conhecimento complementa o outro).
- Ter consciência que as pessoas que trabalham com você tem pontos de vistas, conhecimentos e expectativas diferentes de você.

10 PRINCÍPIOS DA MODELAGEM ÁGIL

- Lembre que o software é seu objetivo principal;
- Saiba que possibilitar o próximo trabalho é seu objetivo secundário;
- Diminua a carga de trabalho;
- Adote a simplicidade;
- Encampe a mudança;
- Mude de forma incremental;
- Modele com um propósito;
- Tenha mais de um modelo;
- Incentive o trabalho de qualidade;
- Maximize o retorno que seus cliente obterão.

PRINCÍPIOS SUPLEMENTARES DA MA

- São princípios suplementares da MA:
  - ✓ O conteúdo é mais importante que a forma;
  - ✓ Todos podem aprender com todos;
  - ✓ Conheça seus modelos;
  - ✓ Adaptação local;
  - ✓ Comunicação aberta e honesta;
  - ✓ Trabalhe com o instinto das pessoas.
- Entenda os princípios suplementares:

PRINC.: O CONTEÚDO É MAIS IMPORTANTE QUE A FORMA

- Qualquer modelo pode ser representado de várias maneiras:
  - ✓ Esboço mental, papel, post-it, quadro, utilizando ferramentas, protótipos, linguagem de programação...;
- O meio pode deixar o modelo mais atrativo, porém o conteúdo do mesmo é o mais importante.

PRINC.: TODOS PODEM APRENDER COM TODOS

- Os modeladores ágeis reconhecem que nunca sabem tudo sobre algo;
- Existe sempre uma oportunidade de aprendizado:
  - ✓ Estender o conhecimento;
- As tecnologias mudam rapidamente.

PRINC.: CONHEÇA SEUS MODELOS

- Muitos modelos podem ser empregados;
- Então é preciso conhecer os pontos fortes e os pontos fracos de cada modelo, somente assim pode-se utilizá-los com eficiência e eficácia.

PRINC.: ADAPTAÇÃO LOCAL

- Talvez seja necessário modificar a MA para que a mesma reflita seu ambiente;
- Analise seu ambiente, reflita sobre sua organização, seus colegas, seus cliente e o próprio projeto;
- Adaptar a MA para atender sua necessidade;
- Pessoas diferentes → aplicações diferentes de MA;
- A aplicação varia, mas os valores, princípios e práticas continuam os mesmos;
- MA não funciona em todas as situações.

PRINC.: COMUNICAÇÃO ABERTA E HONESTA

- Liberdade para oferecer sugestões;
- Liberdade para expressar opiniões;
- Humildade para ouvir opiniões, discutir novas possibilidade e adotar novas estratégias.

PRINC.: TRABALHE COM O INSTINTO ...

- Acredite no seu *feeling*. Muitas vezes sabemos que algo não vai funcionar e não conseguimos embasamento para justificar, mas acredite em seu instinto;
- O tempo e a experiência nos levam a ter percepção dos fatos e torna nosso instinto mais aguçado, apurado;
- Não esqueça que um dos valores do MA é a coragem.

PRÁTICAS DA MODELAGEM ÁGIL

- São o coração da MA;
- São elas que devem ser aplicadas nos projetos.

PRÁTICAS BÁSICAS

1. Modelagem iterativa e incremental:
   - ✓ Aplique o(s) artefato(s) correto(s);
   - ✓ Crie diversos modelos em paralelo;
   - ✓ Itere em outro artefato;
   - ✓ Modele incrementalmente.
2. Trabalho em Equipe:
   - ✓ Modele com outras pessoas;
   - ✓ Organize uma participação ativa dos cliente;
   - ✓ Promova a posse coletiva;
   - ✓ Mostre os modelos publicamente.

PRÁTICAS BÁSICAS

3. Simplicidade:
   - ✓ Crie conteúdo simples;
   - ✓ Mostre os modelos de modo simples;
   - ✓ Use as ferramentas de modo simples.

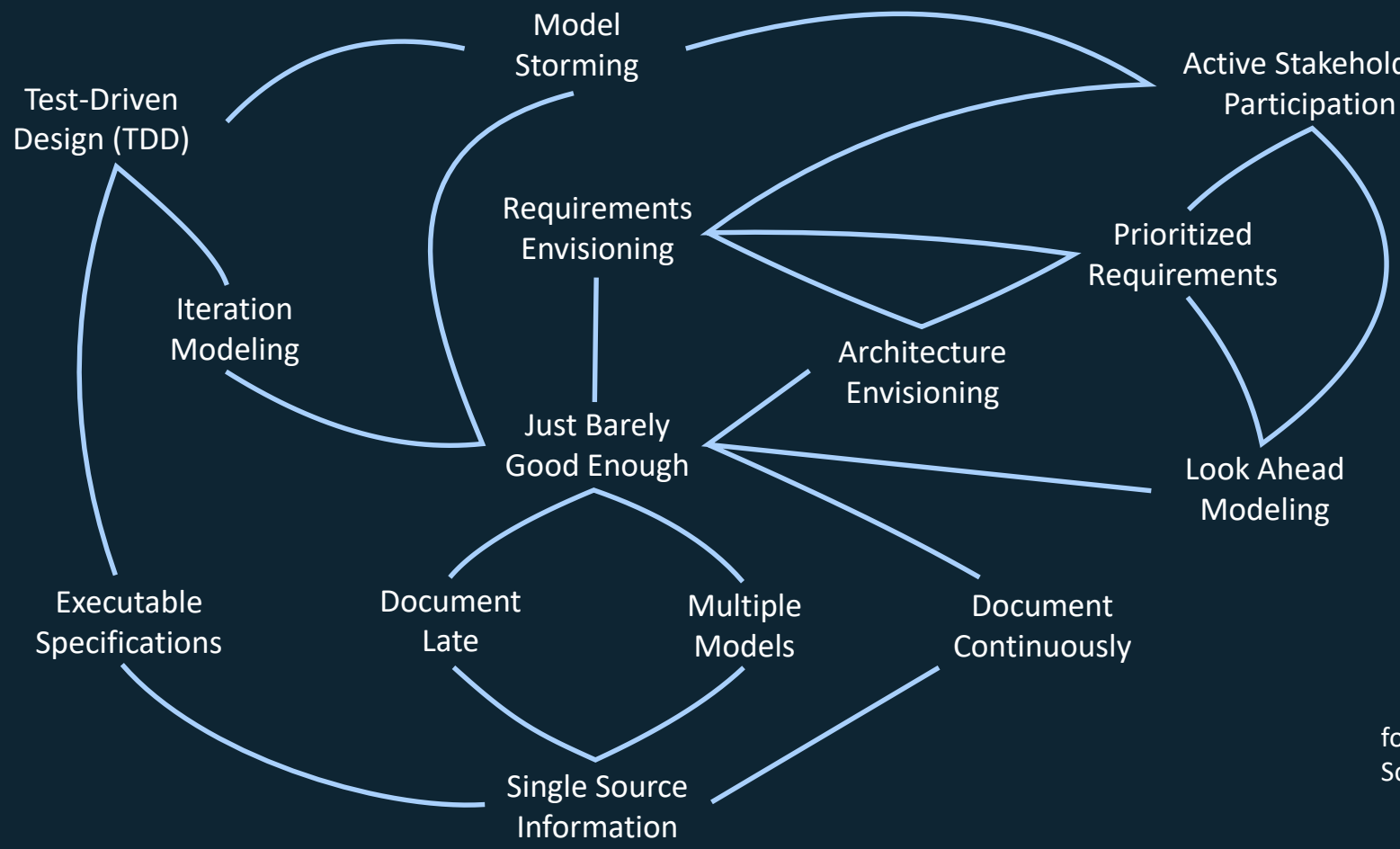
4. Validação:
   - ✓ Considere a testabilidade;
   - ✓ Comprove com código.

PRÁTICAS SUPLEMENTARES

1. Produtividade
   - ✓ Aplique as convenções da modelagem;
   - ✓ Utilize os padrões com moderações;
   - ✓ Reuse os recursos já existentes.
2. Documentação
   - ✓ Descarte os modelos temporários;
   - ✓ Formalize os modelos de contrato;
   - ✓ Atualize apenas quando necessário.
3. Motivação
   - ✓ Modele para entender;
   - ✓ Modele para comunicar.

RESUMO

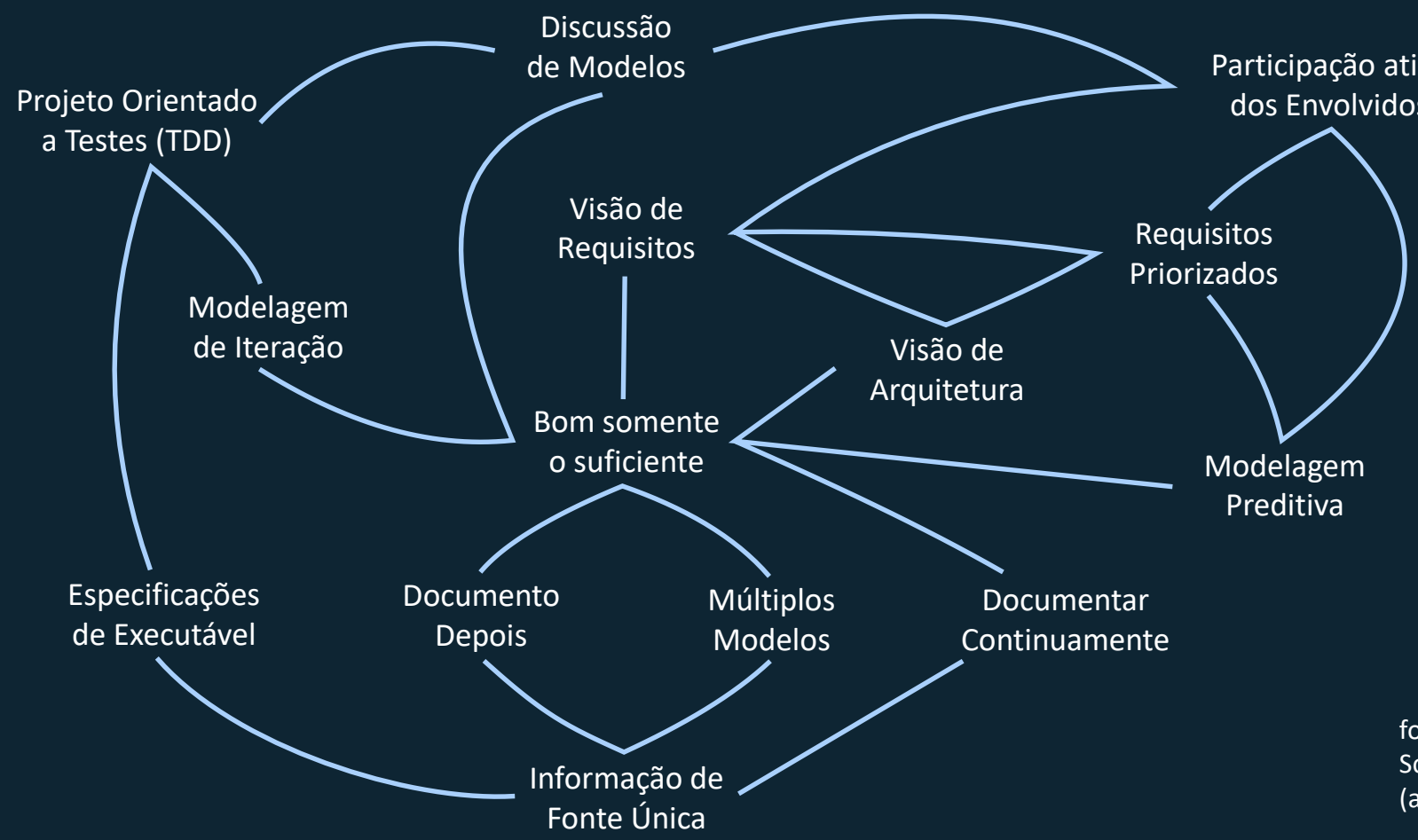
The Best Practices of Agile Modeling



fonte:
Scott W. Ambler (2005-2011)

RESUMO

As Melhores Práticas da Modelagem Ágil



fonte:
Scott W. Ambler (2005-2011)
(adaptado)

ATIVIDADE PRÁTICA

REFERÊNCIAS

- 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.