Modelagem de Sistemas de Informação

Ciclo de Vida do SW

Luiz Sérgio Almeida da Silva

luiz.sasilva@sp.senac.br

Curso.: BSI

2025

Senac
CENTRO UNIVERSITÁRIO SENAC

Escolha do Paradigma

- Cada modelo representa um processo a partir de uma perspectiva particular:
  - Natureza do Projeto e Aplicação
  - Escolha de Métodos e Ferramentas
  - Controles e produtos a serem entregues

Modelos de Ciclos de Vida

- Formalmente, modelos de processos

- Fases de evolução de um produto
  - Requerimentos
  - Especificação
  - Projeto
  - Implementação
  - Integração
  - Manutenção
  - Aposentadoria

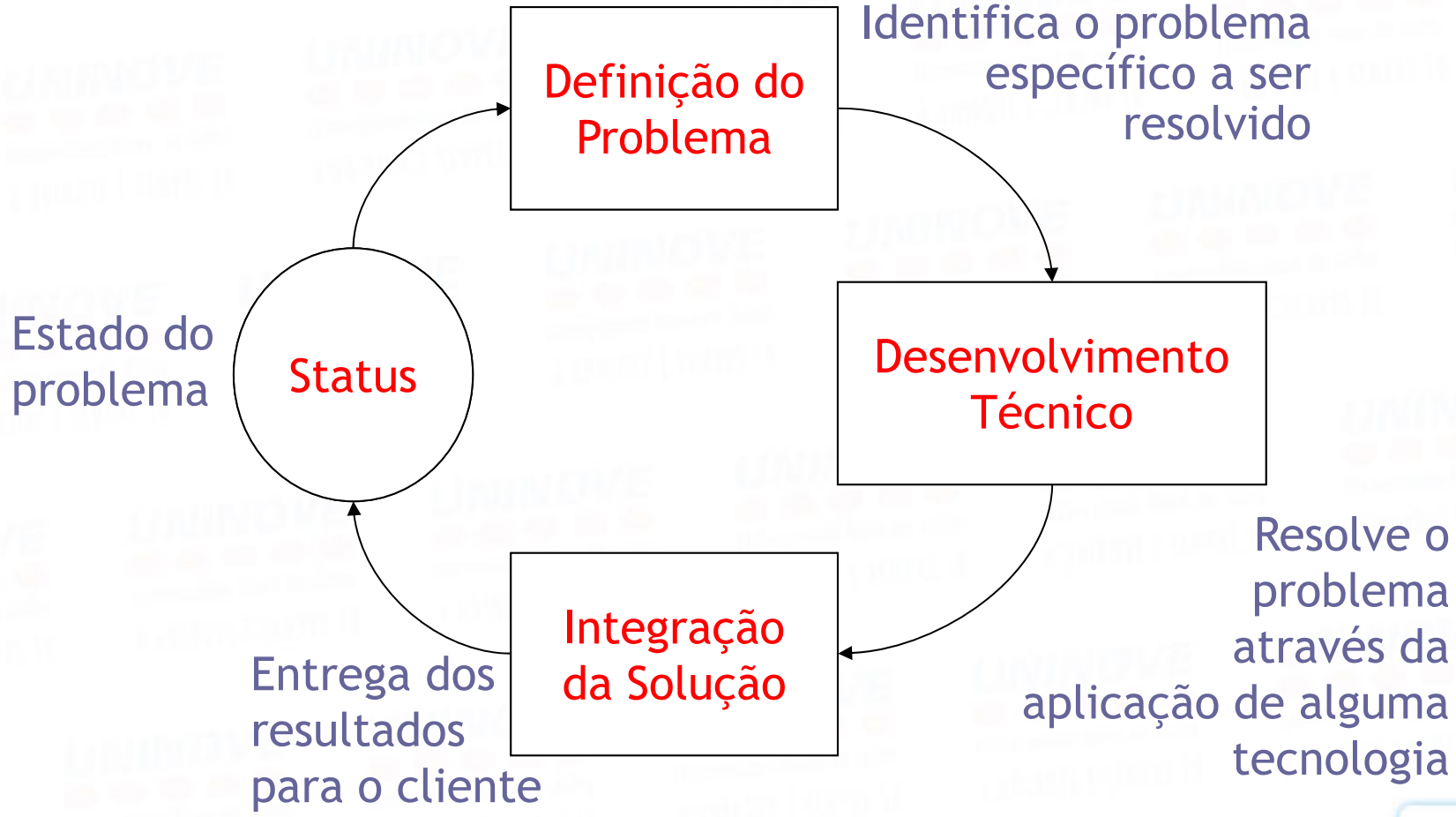
☑ *Definição*

☑ *Desenvolvimento*

☑ *Manutenção*

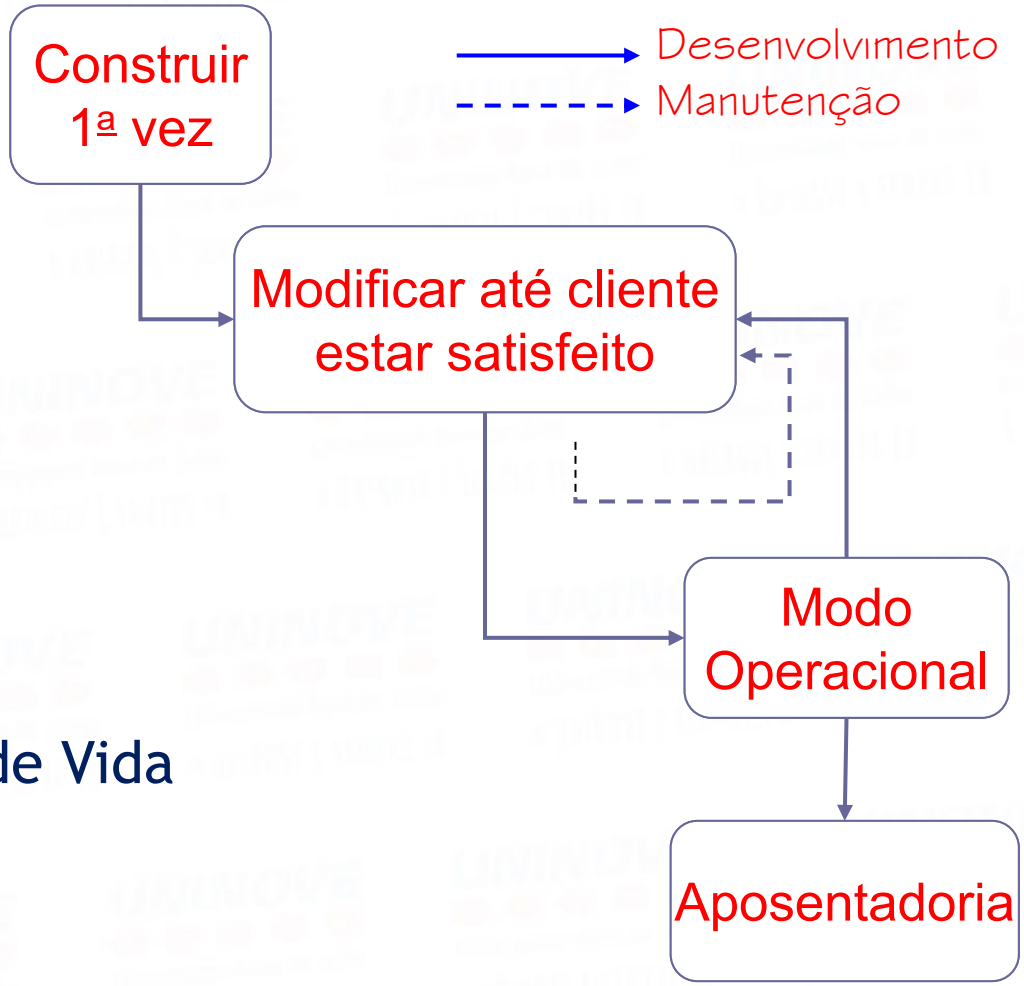
Escolha do Paradigma

- Fases de um *loop* de resolução de um problema



Construir e Consertar

- Pequenas Aplicações
- Problemas
  - Sem especificação
  - Sem Projeto
- Insatisfatório
- Alto Custo de Manutenção
- Falta um Modelo de Ciclo de Vida
  - Planejamento
  - Fases
  - Fundamentos

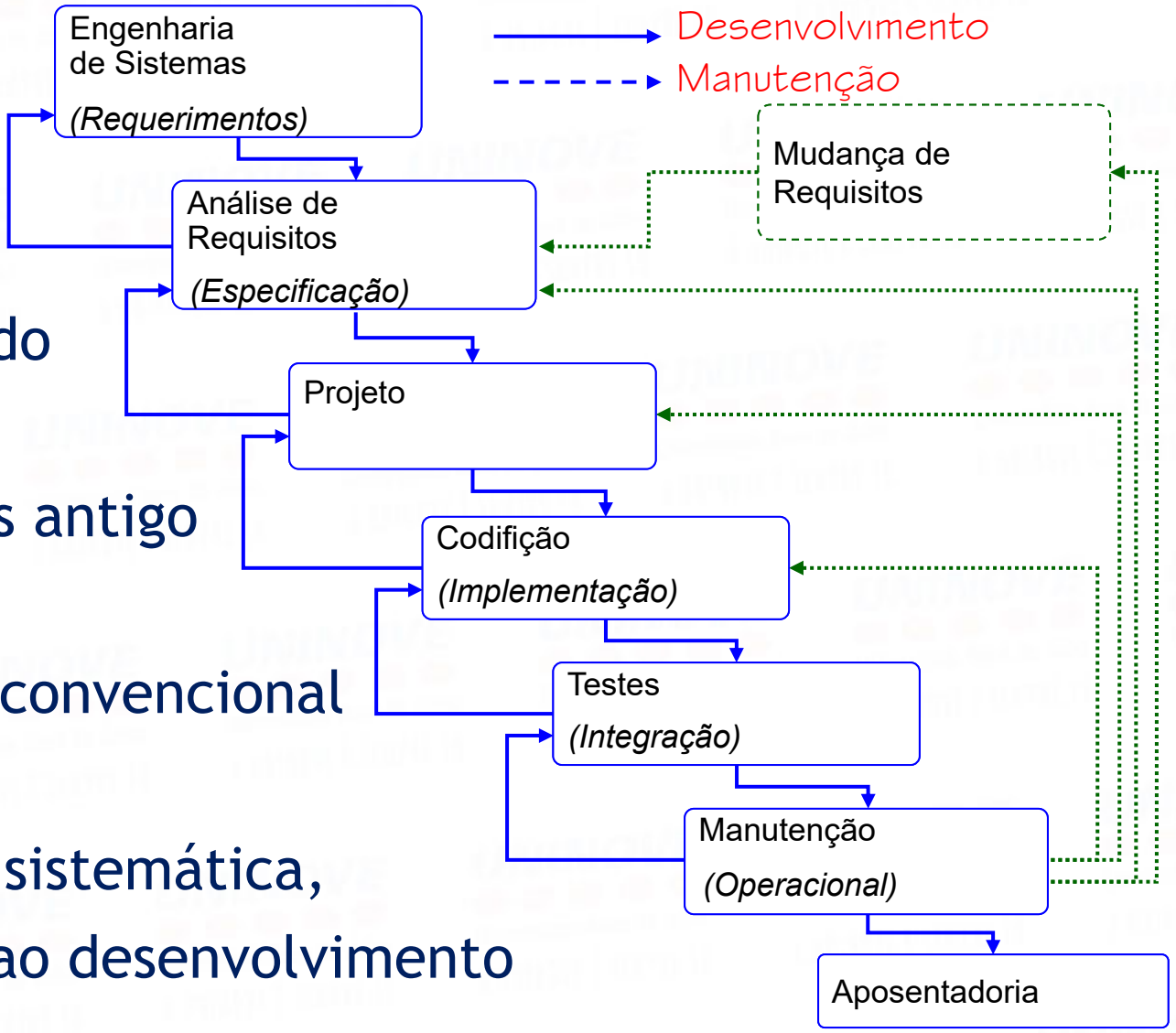


TIPOS DE CICLO DE VIDA DE SW

Modelo em Cascata

Ciclo de Vida do Software

- Mais utilizado
- Modelo mais antigo
- Engenharia convencional
- Abordagem sistemática, seqüencial ao desenvolvimento do software



Modelo em Cascata
Atividades

- Análise de Engenharia de Sistemas
  (Requerimentos)
  - Coleta de requisitos do sistema
  - Essencial para interface com demais elementos *(hardware, pessoas, banco de dados)*

- *Análise de Requisitos de Software*
  (Especificação)
  - Coleta de requisitos intensificada e concentrada no software
  - Compreender o domínio da informação, função, desempenho e interfaces
  - Requisitos documentados e revistos (com cliente)

Estratégia
Entradas
Saídas
Estimativas
Restrições
Documentação

Modelo em Cascata
Atividades

- *Projeto*
  - Tradução dos requisitos para representações que podem ser avaliadas

Atributos do programa
Estrutura de Dados
Arquitetura de Software
Detalhes Procedimentais
Caracterização de Interfaces

- Codificação (Implementação)
  - Tradução para linguagem *artificial*
- *Testes (Integração)*
  - Unidades integradas e testadas como um sistema completo
  - Garantir que os requisitos foram atendidos
- *Manutenção*
  - Corrigir erros não encontrados anteriormente
  - Melhorar implementação
  - Aumentar função (novos requisitos)

Modelo em Cascata

Vantagens e Problemas

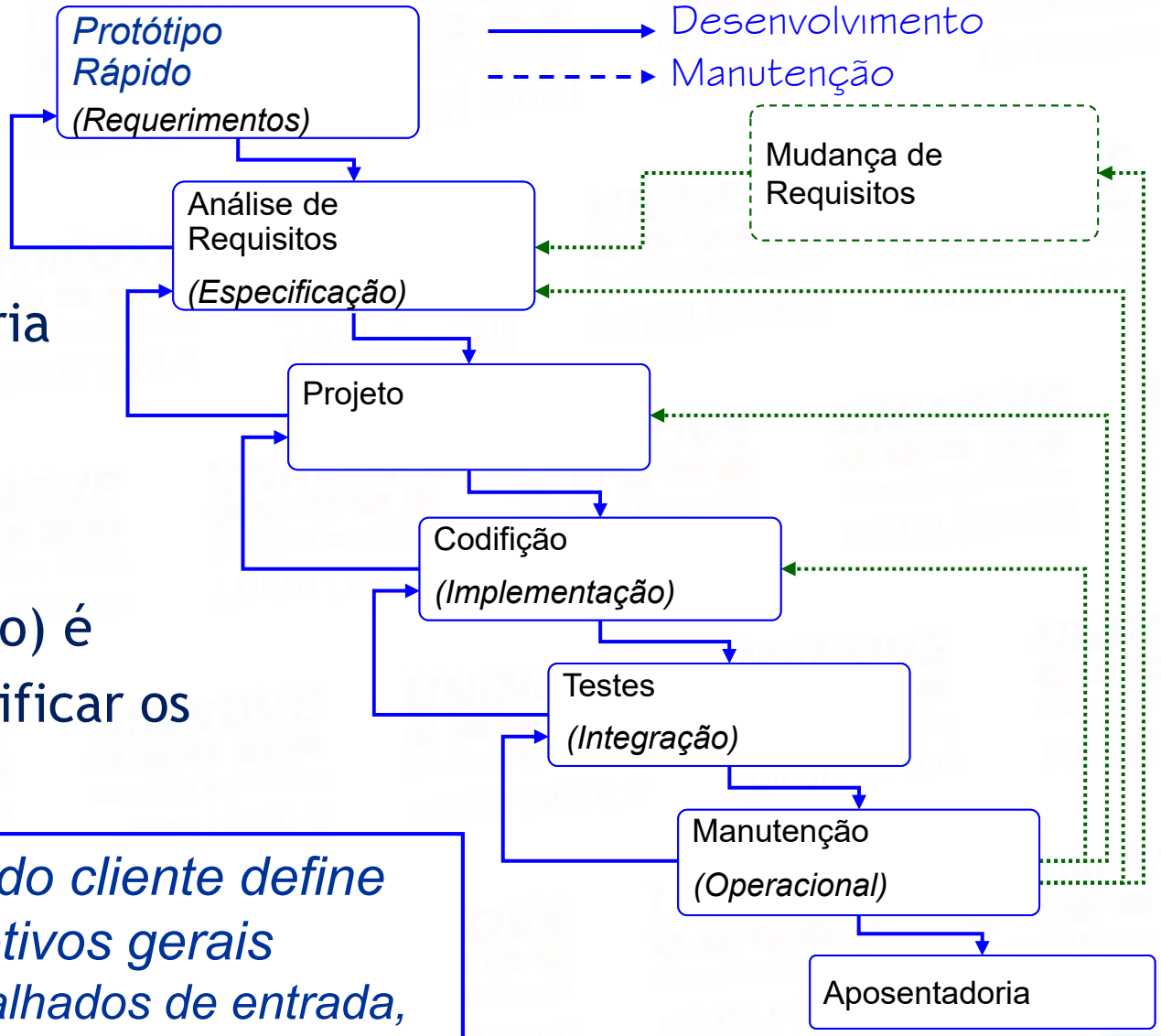
- Vantagens
  - Dirigido à documentação
  - Abordagem disciplinada
  - Manutenção relativamente fácil

- *Problemas*
  - Dificuldade em seguir o fluxo
  - Determinação dos Requisitos Iniciais (Incerteza natural)
  - Resultado Efetivo apenas no final
  - Inflexibilidade torna difícil responder as mudanças

Usado somente quando os requisitos forem bem compreendidos

Modelo de Prototipação

- Ciclo de Vida Clássico Modificado
- Desenvolvedor cria um modelo de software
- Modelo (protótipo) é usado para identificar os requisitos



Apropriado quando cliente define apenas objetivos gerais
(sem requisitos detalhados de entrada, processamento e saída)

Modelo de Protipação

Ciclo de Vida

Modelo de Protipação
Atividades

- *Obtenção dos Requisitos*
  - Desenvolvedor e cliente
    - Definição de objetivos gerais
    - Identificação de requisitos conhecidos e áreas que necessitam definições adicionais
- *Projeto Rápido*
  - Representações do que é visível para o usuário (interface de entrada e saída)
- *Construção do protótipo*
  - Implementação do projeto rápido

Modelo de Protipação
Atividades

- *Avaliação pelo Cliente*
  - Cliente e Desenvolvedor avaliam o protótipo

- *Refinamento dos Requisitos*
  - Processo de interação que pode conduzir ao retorno do projeto rápido
  - Repete-se até cliente estar satisfeito e desenvolvedor compreenda o que deve ser feito

- *Construção do Produto*
  - Com requisitos identificados, protótipo é descartado e Produto construído

Modelo de Protipação

Vantagens e Problemas

- *Vantagens*
  - Facilidade para determinar Requisitos Iniciais
  - Garantia de atingir as necessidades do Cliente

- *Problemas*
  - Implementação do protótipo rápido comprometido
  - Determinar um fim para o desenvolvimento
  - Tendência a utilizar o protótipo como produto final

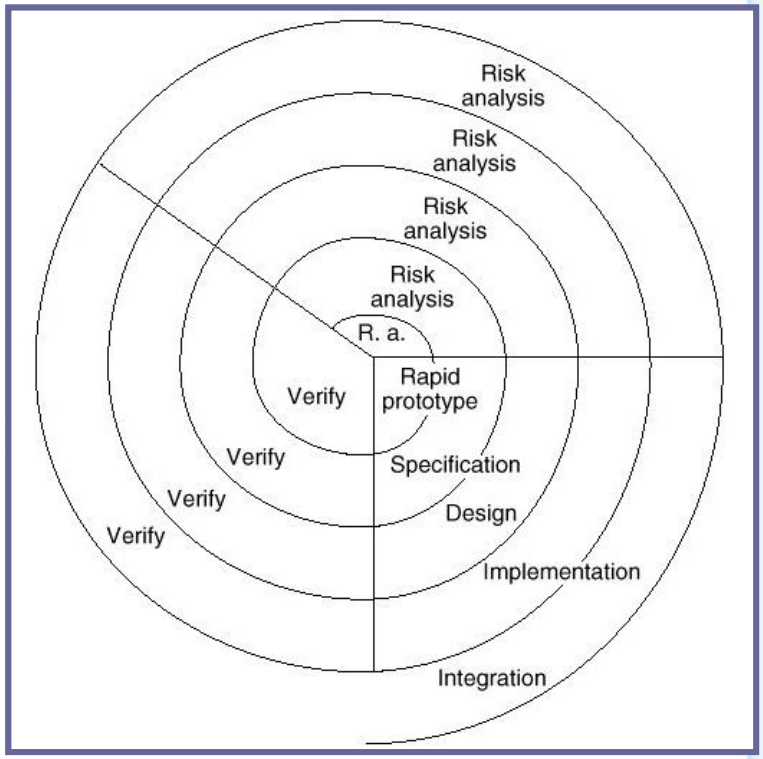
Ciclo de vida eficiente quando as regras são definidas no início e há concordância que o protótipo é um protótipo

Ciclo de Vida em Espiral

- *Evolução dos Ciclos de Vida Clássico* e Prototipação com acréscimo da *Análise de Risco*

- *Processo é representado por uma espiral de atividades*
  - Cada contorno da espiral representa uma fase do processo

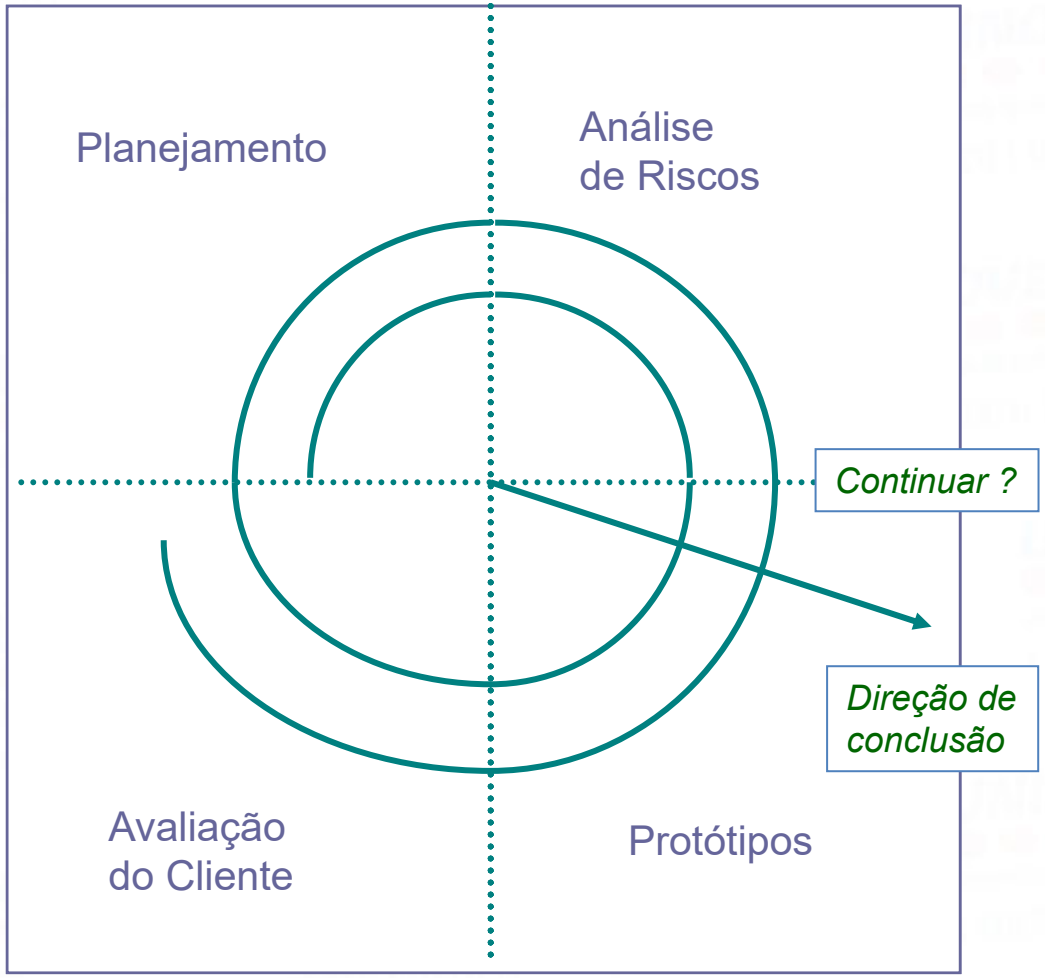
- *Cada fase :*
  - Precedida de *Alternativas* e *Análise de Risco*
  - Seguida de *Avaliação* e *Planejamento* para a próxima fase



Abordagem sistemática do Modelo em Cascata com uma estrutura interativa para refletir o mundo real

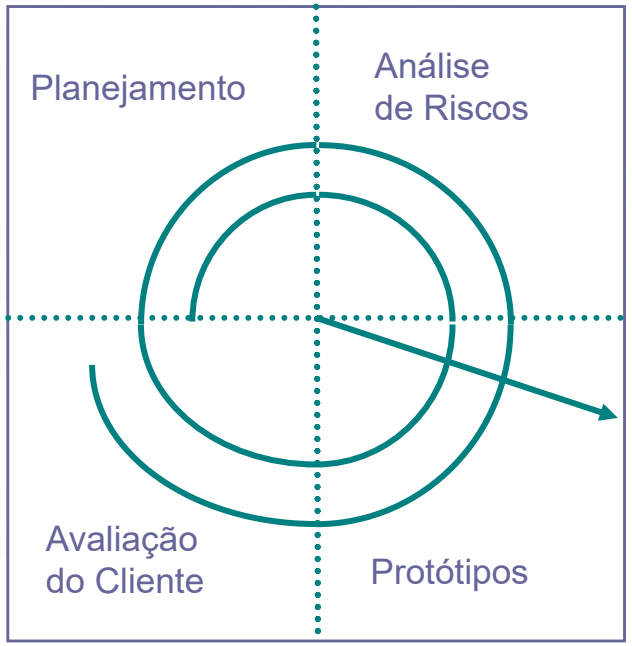
Ciclo de Vida em Espiral
Visão Simplificada

- Direção Radial
  - custo (tempo) acumulativo
  - Se todos os riscos não puderem ser resolvidos, o projeto deve ser finalizado imediatamente
- Direção Angular
  - Progresso através da espiral



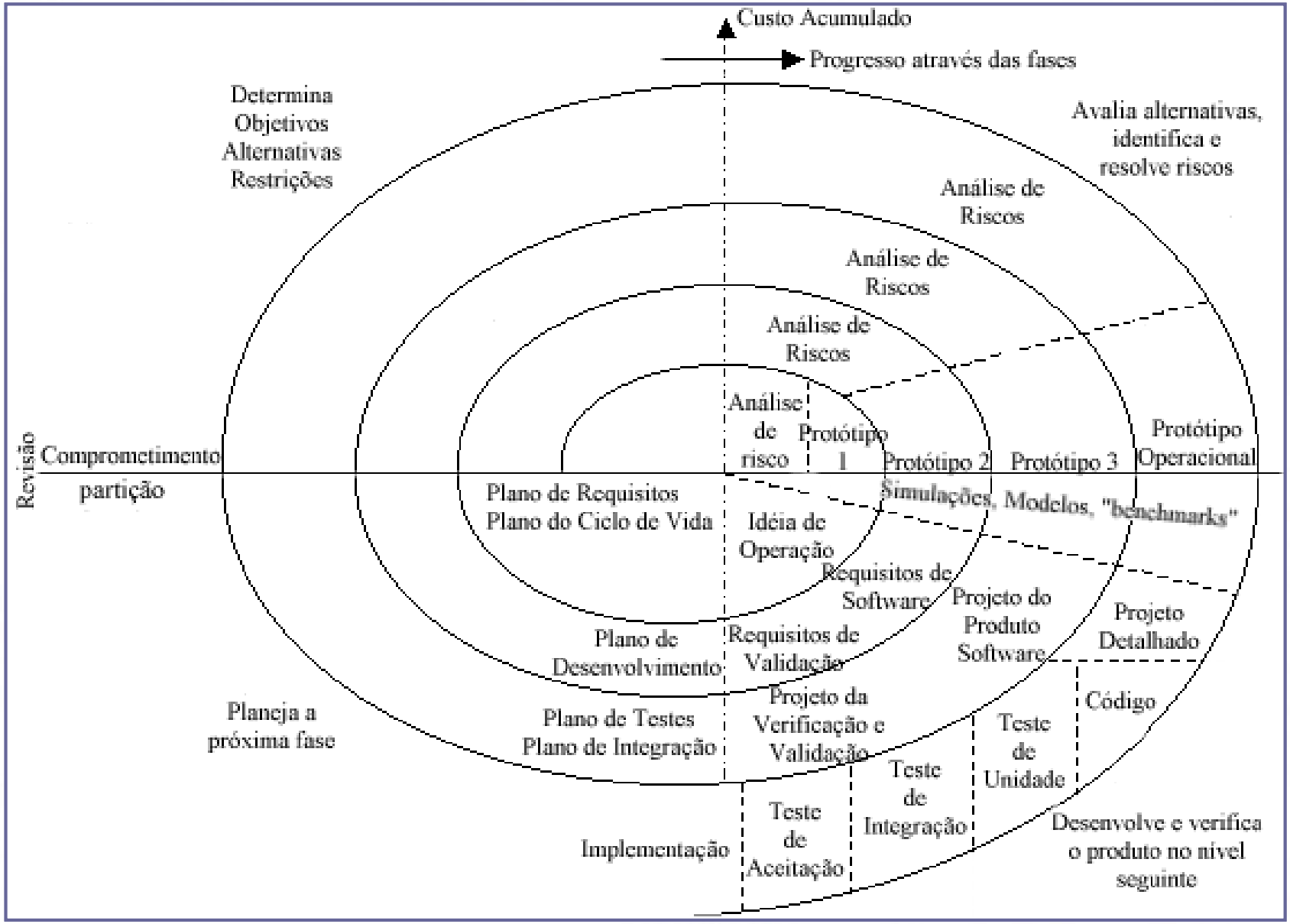
Ciclo de Vida em Espiral
Atividades

- *Planejamento*
  - Determinação dos objetivos, alternativas e restrições
- *Análise de Risco*
  - Análise das alternativas e identificação / resolução dos riscos
- *Construção (Protótipos)*
  - Desenvolvimento do produto no nível seguinte
- *Avaliação do Cliente*
  - Avaliação do produto e planejamento de novas fases



Ciclo de Vida em Espiral

Modelo Completo - Boehm 1988



Ciclo de Vida em Espiral

Vantagens e Problemas

- *Vantagens*
  - Análise de Risco
  - Fácil adaptação à Evolução
  - Sem distinção entre desenvolvimento e manutenção

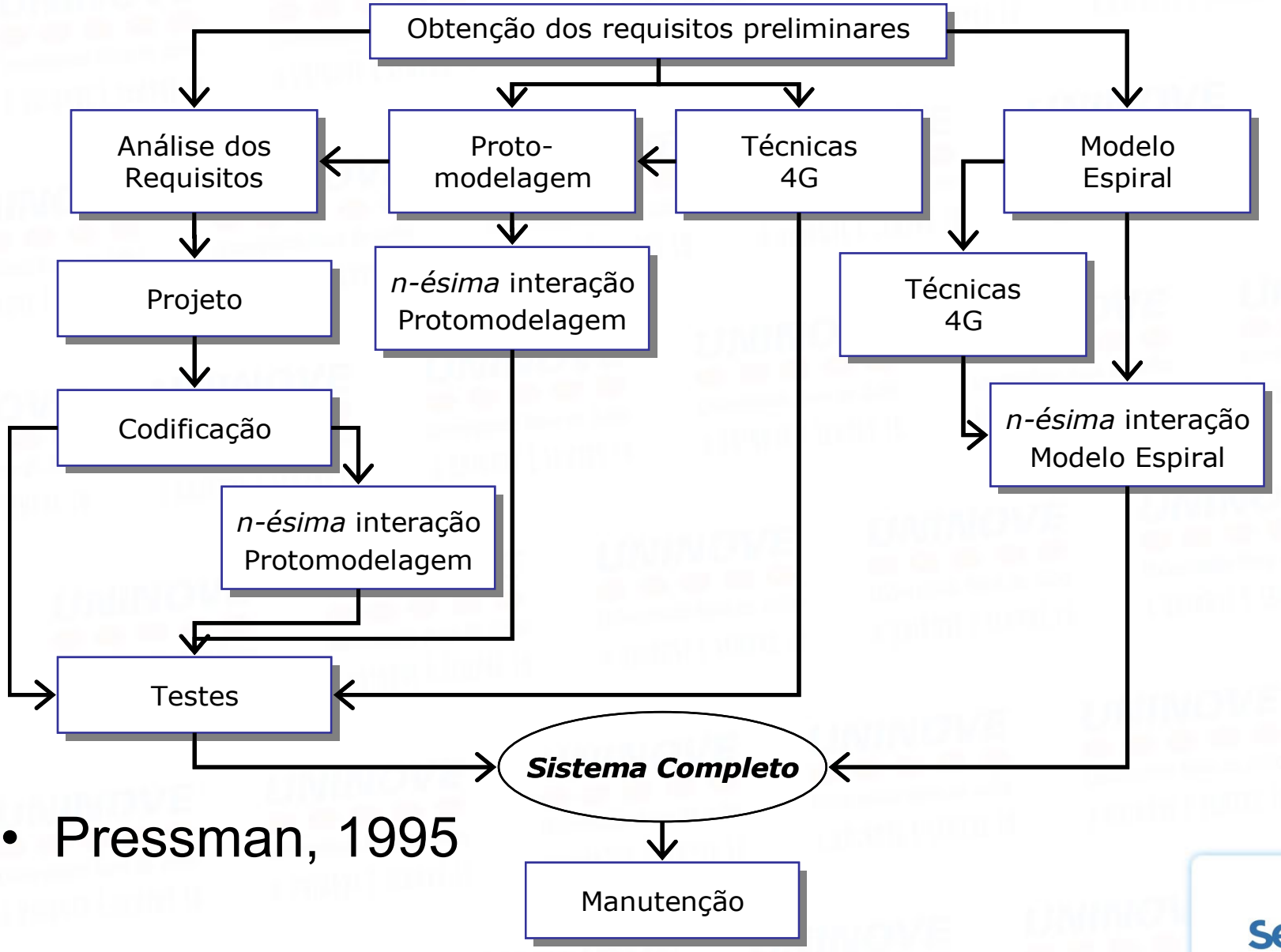
- *Problemas*
  - Produtos “caseiros” em larga escala
  - Necessita Competência em Análise e Resolução de Risco

É possível controlar a Evolução !?

Conclusões

- *Diferentes Modelos de Ciclo de Vida*
  - Cada um tem suas Vantagens
  - Cada um tem suas Desvantagens
- *Critérios para escolher um modelo incluem:*
  - A Organização
  - Seu Gerenciamento
  - Capacitação dos trabalhadores
  - Natureza do Produto
- *Melhor Sugestão*
  - Modelo de Vida “Misture-e-Ajuste”

Conclusões



- Pressman, 1995


Dúvidas?
Não mais..

QUESTÕES

Exercício - Case´s

Case (1):

Uma Pizzaria tem um sistema desenvolvido em VB6 para controlar seus pedidos e suas entregas. A pizzaria esta ampliando suas instalações e uma das mudanças seria re-construir o sistema de controle de pedidos e entregas para uma plataforma moderna, onde de qualquer lugar os cliente poderão ter acesso ao sistema para realizar pedido e consultar o cardápio da pizzaria. Sua fabrica de software foi contratada para desenvolver esse sistema. Então, você como gerente de projeto precisa decidir qual e o melhor ciclo de vida para utilizar no projeto, **justifique a sua resposta e coloque as fases e os motivos para a escolha do ciclo de vida.**

Exercício - Case´s

Case (2):

A sua fabrica de software ganhou uma licitação para desenvolver um sistema para Policia Federal (PF). Esse sistema será utilizado para gerenciar e controlar as operações policiais, onde todas as informações inseridas no sistema devem ser de caráter sigiloso, ou seja, as pessoas que não estiverem envolvidas nessa operação policial não poderão ter acesso aos registros dessa operação. Sua fabrica de software precisará levantar todos os requisitos do sistema, uma vez que, esse processo é realizado hoje de forma manual. Sua missão como gerente de projeto é gerenciar e controlar todo desenvolvimento do software até a sua implantação, sendo que seu maior desafio é entregar o sistema ainda esse ano. **Pergunta, qual ciclo de vida você utilizara para conduzir esse projeto e quais fases serão utilizadas, justifique.**

Exercício - Case´s

Case (3):

Uma banca de revista deseja desenvolver um site para divulgar seus produtos na internet. O dono da banca de revista sabe que seu vizinho que estuda computação tem um grupo de amigos na faculdade que desenvolve pequenos sistemas para levantar uma grana para se divertirem. Ele acredita que os meninos da computação sejam capazes de desenvolverem seu site com um preço acessível e com alta qualidade, já que eles adoram tecnologia. A missão desse grupo de alunos da computação é desenvolver um site para o dono da banca de forma que atenda a sua expectativa, uma vez que, ele poderá divulgar o trabalho desse grupo para outras pessoas. **Pergunta, qual ciclo de vida você utilizaria para desenvolver esse projeto e quais fases serão utilizadas, justifique.**

FÁBRICA DE SOFTWARE

O que é a Fábrica de Software?

É uma solução tecnológica, que funciona em linha de montagem, para modelar processos e desenvolver sistemas, composta por metodologia, equipamentos e programas organizados em determinado padrão ambiental.
Nela são executadas todas as fases de desenvolvimento de sistemas, da especificação à implementação final.

Como Funciona a Fabrica de Software

Uma fabrica de software possui is mesmos recursos de uma fábrica modelo. A função da fábrica é maximizar a produção de software. A programação orientada a objetos e por componentes permite criar objetos reutilizáveis que podem ser usados em diferentes programas, reduzindo os custos de programação. A fábrica deve ter uma estrutura com: atendimento à clientes, planejamento e controle de produção, produção, garantia e qualidade.

Estrutura de uma fábrica de software

- Área de atendimento a clientes
- Área de planejamento e controle da produção
- Área de planejamento
- Área de qualidade e garantia
- Área de suporte

O que é um software de qualidade?

- O software que satisfaz os requisitos solicitados pelo usuário. Deve ser fácil de manter, ter boa performance, ser confiável e fácil de usar
- Alguns atributos de qualidade
  - Manutenibilidade
    - O software deve evoluir para atender os requisitos que mudam
  - Eficiência
    - O software não deve desperdiçar os recursos do sistema
  - Usabilidade
    - O software deve ser fácil de usar pelos usuários para os quais ele foi projetado

Qualidade de Software (um exemplo para o Varejo)

- *Correto*
  - A loja não pode deixar de cobrar por produtos comprados pelo consumidor
- *Robusto e altamente disponível*
  - A loja não pode parar de vender
- *Eficiente*
  - O consumidor não pode esperar
  - A empresa quer investir pouco em recursos computacionais (CPU, memória, rede)

Qualidade de Software (um exemplo para o Varejo)

- *Amigável e fácil de usar*
  - A empresa quer investir pouco em treinamento
- *Altamente extensível e adaptável*
  - A empresa tem sempre novos requisitos (para ontem!)
  - A empresa quer o software customizado do seu jeito (interface, teclado, idioma, moeda, etc.)
- *Reusável*
  - Várias empresas precisam usar partes de um mesmo sistema

Qualidade de Software (um exemplo para o Varejo)

- *Aberto, compatível, de fácil integração com outros sistemas*
  - A empresa já tem controle de estoque, fidelização, etc.
- *Portável e independente de plataforma (HW e SW)*
  - A empresa opta por uma determinada plataforma
- *Baixo custo de instalação e atualização*
  - A empresa tem um grande número de PDVs

Modelo de Processo

- É uma representação de um processo, usualmente envolvendo
  - atividades a serem realizadas
  - agentes que realizam as atividades
  - artefatos (produtos) gerados
  - recursos necessários (consumidos)

Modelo de Processo

- Um modelo é usado para entendimento e comunicação do processo, e como base para análise, execução, gerência e melhoria do processo
- Idealmente a descrição deve ser formal e completa para permitir, por exemplo, automação
- A descrição deve ser apresentada em diferentes níveis de abstração

Exemplos de processos

- Processos tradicionais (pesados)
  - RUP,OPEN, Catalysis
- Processos ágeis (leves)
  - XP, Agile modeling, Crystal, pragmatic programming, Internet Speed, Scrum, ...

Exemplos de processos

- Consenso em torno de:
  - Iteratividade
  - Participação de usuários
  - Flexibilidade de configuração para projetos específicos
  - Comunicação entre membros da equipe

Exemplos de processos

- *Divergências em torno de...*
  - Detalhamento de atividades a serem seguidas
  - Critério de conclusão da execução das atividades
    - Arquitetura robusta (RUP)
    - Arquitetura para o contexto da iteração atual (agile modeling)
  - Rigor na atribuição de tarefas a responsáveis
    - workers (RUP)
    - alocação sob demanda e interesse (XP)
  - Artefatos (documentação) gerados
  - Nível de Automação
  - Nível de (im)pessoalidade

O que é RUP (Rational Unified Process) ?

É um processo configurável de Engenharia de Software.

O RUP é um guia para como usar efetivamente a UML.

O objetivo do RUP é assegurar uma produção de alta qualidade de software, que realiza a necessidade do usuário seguindo prazos e o orçamento.

RUP - Utilidades

- O RUP, como processo de desenvolvimento de software, tem 4 regras:
  - servir de guia;
  - especificar quais artefatos devem ser desenvolvidos e quando devem ser desenvolvidos;
  - dirigir as tarefas individuais e do time como um todo;
  - oferecer critérios para monitorar e medir os produtos e atividades do projeto.


Dúvidas?
Não mais..

Questões

1. Quais são as principais características positivas, de acordo com ciclo de vida espiral?
2. Explique brevemente cada uma das fases de um ciclo de vida de prototipação.
3. Quais são as desvantagens do ciclo de vida em espiral?
4. O que é um protótipo? Que cuidados são importantes e próprios desse modelo no que diz respeito a comunicação com os clientes?
5. Qual a importância da engenharia de software e como se justificam os custos a ela associados?
6. O que você entende por processo de desenvolvimento de software e qual a sua importância para a qualidade dos produtos de software? Qual a diferença entre processo e projeto de software?

Obrigado !!!

Senac
CENTRO UNIVERSITÁRIO SENAC