

Engenharia de Software

2023

Engenharia de Software

Importância da correta reutilização de código.

- Atualmente: comum a entrega de com atraso, erros e excedendo orçamentos previstos.
- Requisitos mudam constantemente.
- Reinvenção da roda: kit de algoritmos não compartilhado entre programadores.
- Reutilização binária

Engenharia de Software

Alguns exemplos desastrosos

Engenharia de Software

Stanislav Petrov

O sistema de alerta da União Soviética esteve a ponto em 1983, em plena guerra fria, de causar a III Guerra Mundial quando um erro no software indicou que cinco mísseis balísticos tinham sido lançados pelos Estados Unidos.

Stanislav Petrov.

Engenharia de Software

- Colapso da At& T

Em 1990, 75 milhões de chamadas de telefone em todos os EUA ficaram sem resposta depois de um único interruptor em um dos 114 centros de comutação da AT & T sofreu um pequeno problema mecânico, que fechou o centro. Quando o centro voltou-se logo em seguida, ele enviou uma mensagem para outros centros, que por sua vez levaram a tropeçar e desligar e reiniciar.

O culpado acabou por ser um erro em uma única linha de código que tinha sido adicionada durante uma atualização de software altamente complexo. American Airlines sozinho estima que este pequeno erro custou 200 mil reservas.

Engenharia de Software

- Explosão Ariane - 5

Em 1996, o foguete europeu mais novo e não-tripulada de lançamento de satélite, o Ariane 5, foi explodido apenas alguns segundos depois de decolar . A Agência Espacial Europeia estima que o desenvolvimento total custou mais de US \$ 8 bilhões. A bordo foram US \$ 500 milhões de 4 satélites. De acordo com The New York Times Magazine, a auto-destruição foi provocada por software tentando encher "um número de 64 bits em um espaço de 16 bits".

"Essa paralisação ocorreu 36,7 segundo após o lançamento, quando computador do próprio sistema de orientação tentou converter um pedaço de dados - a velocidade para o lado do foguete -. Partir de um formato de 64 bits para um formato de 16 bits. O foguete desviou da rota e explodiu.

Engenharia de Software

- Mars polar lander

Um erro na navegação da nave espacial Mars Polar Lander fez com que ela voasse muito baixo e se espatifasse contra o solo. A falha foi devido a uma empresa terceirizada que confundiu o sistema de medidas americano e europeu.

Engenharia de Software

- Mercy Medical Center of St. Mary

Em fevereiro de 2003, um erro de software de computador no Mercy Medical Center de St. Mary, em Grand Rapids, Michigan, que custou a vida de 8.500 pacientes. Na realidade, cada um dos pacientes, que tiveram procedimentos feitos entre 25 de Outubro a 11 de Dezembro do ano anterior, estavam vivos. No entanto, a falha, atribuída ao sistema do hospital paciente gestão, notificado a Segurança Social, as companhias de seguros de pacientes e os próprios pacientes, dos óbitos "infelizes".

Engenharia de Software

•Therac - 25

Therac-25 era uma máquina de radioterapia controlada por computador, muito moderna para sua época, por permitir a utilização do mesmo equipamento para a aplicação de diversas doses de radiação nos pacientes. Houve uma série de pelo menos 6 acidentes entre 1985 e 1987, nos quais os pacientes receberam overdose de radiação. Pelo menos cinco mortes aconteceram devido aos acidentes, causados por erros no software que controlava a máquina. Este acidente mostrou o perigo que reside em softwares que controlam operações de segurança.

Crise de Software

- Termo utilizado nos anos 70 (aumento de demanda)

Problemas enfrentados

- Identificar adequadamente os requisitos do Sistema, ou seja, saber o que o software deve fazer;
- Que ferramentas, linguagem, sistema operacional usar;
- Como diminuir os tempos e custos de desenvolvimento;
- Prever falhas antes da entrega final;
- Como fazer manutenção e controlar versões;
- Dificuldades de prever o progresso durante o desenvolvimento;
- Inexistência de histórico, ou documentação, no desenvolvimento de Sistemas de Comunicação com os usuários precária;
- Conceitos quantitativos inexistentes tais como confiabilidade, qualidade e reusabilidade;
- Manutenção, no software existente, com difícil execução.

Engenharia de Software

Como garantir a exatidão do programa?

Testando cada recurso. Dependendo do projeto, teste de cada função. Pessoal treinado exclusivamente.

Como era garantida a gerência de projetos?

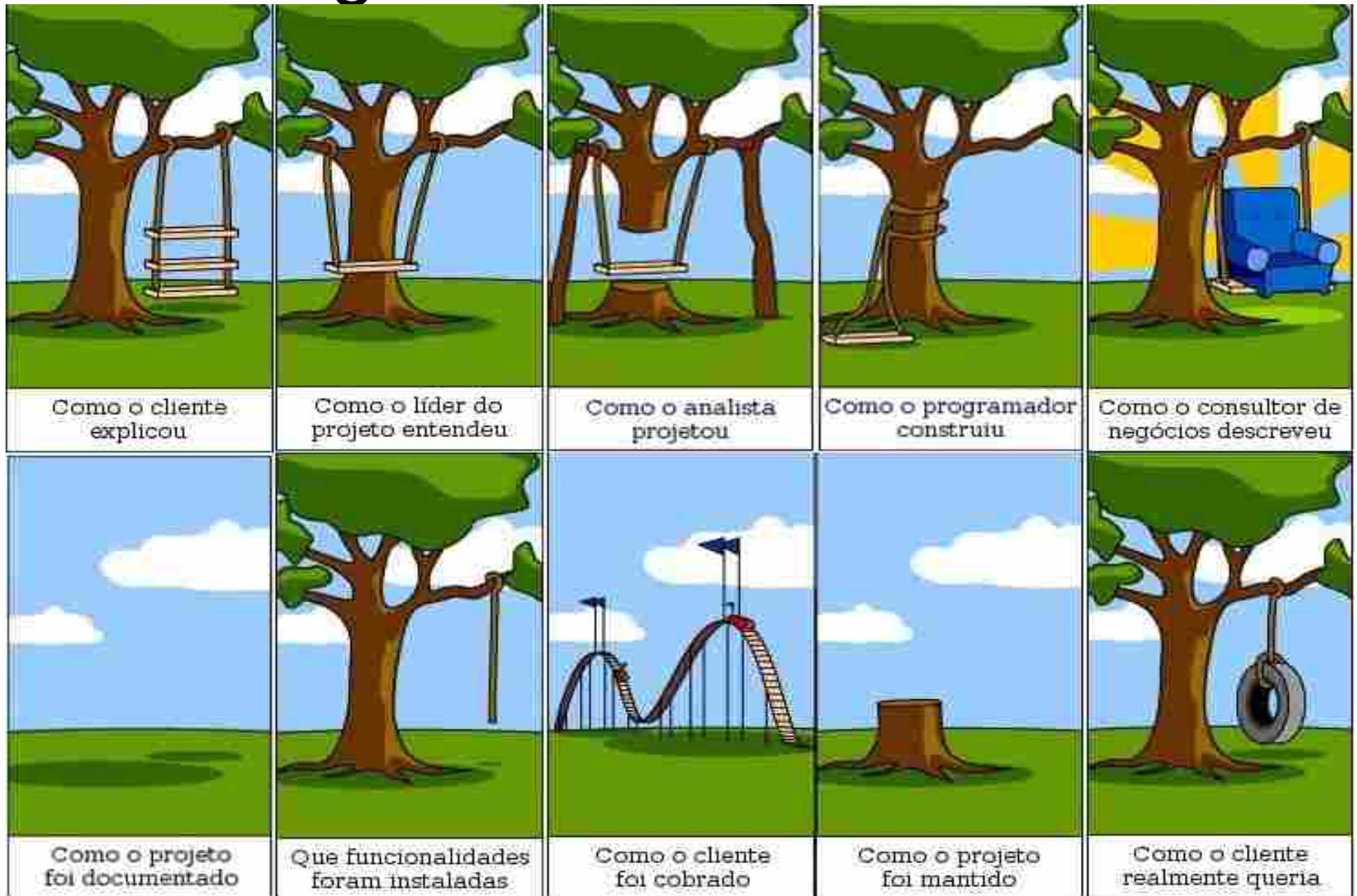
Não permitir entrada de novos desenvolvedores.

O gerente monitorava o progresso. Dividia o projeto em duas fases: entrega um subconjunto de funcionalidades na data, e na segunda fase o restante.

Engenharia de Software

Como desenvolver um software?
Fases do processo de desenvolvimento
Definição
Desenvolvimento
Manutenção

Engenharia de Software



Modelos de Processo

Utilizados para descrever as principais etapas de desenvolvimento. Há diversos modelos. Em comum, a maioria possuem como princípios cinco macro-etapas:

- **Requisitos**
- **Projeto**
- **Desenvolvimento**
- **Implantação**
- **Manutenção**

Nomenclatura utilizada por autores: Ciclo de Vida de Software; Prescritivos de Processo; Paradigmas de Ciclo de Vida; Modelagem do Ciclo de Vida.

Modelos de Processo

Utilizados para descrever as principais etapas de desenvolvimento. Há diversos modelos. Em comum, a maioria possuem como princípios três macro-etapas:

- **Requisitos:** Momento em que o analista deve entrevistar o usuário. As respostas às perguntas devem contemplar o que é esperado do software, sua abrangência, seu escopo, quais formas possíveis de resolver o problema, restrições, tipos de dados que serão utilizados, etc.
- **Projeto**
- **Desenvolvimento**
- **Implantação**
- **Manutenção**

Modelos de Processo

Utilizados para descrever as principais etapas de desenvolvimento. Há diversos modelos. Em comum, a maioria possuem como princípios três macro-etapas:

- **Requisitos:**
- **Projeto:** Com o(s) documento(s) de requisitos em mãos, o analista elabora o detalhamento da solução a ser desenvolvida através de especificações técnicas.
- **Desenvolvimento**
- **Implantação**
- **Manutenção**

Modelos de Processo

Utilizados para descrever as principais etapas de desenvolvimento. Há diversos modelos. Em comum, a maioria possuem como princípios três macro-etapas:

- **Requisitos:**
- **Projeto:**
- **Desenvolvimento:** Momento em que os desenvolvedores codificam o projeto em alguma linguagem de programação.
- **Implantação**
- **Manutenção**

Modelos de Processo

Utilizados para descrever as principais etapas de desenvolvimento. Há diversos modelos. Em comum, a maioria possuem como princípios três macro-etapas:

- **Requisitos:**
- **Projeto:**
- **Desenvolvimento:**
- **Implantação:** Geralmente nesta etapa pode ocorrer diversos problemas imprevistos nas etapas anteriores.
- **Manutenção**

Modelos de Processo

Utilizados para descrever as principais etapas de desenvolvimento. Há diversos modelos. Em comum, a maioria possuem como princípios três macro-etapas:

- **Requisitos:**
- **Projeto:**
- **Desenvolvimento:**
- **Implantação:**
- **Manutenção:** Etapa que dura toda a vida útil do software, atuando nas correções de problemas existentes, adaptações necessárias devido a novas funcionalidades, sejam elas, internas ou externas e melhoramentos futuros.

Modelos de Processo

Modelo da Balbúrdia:

Geralmente consiste em um ciclo de duas fases:

- * Implementação
- * Implantação

Novos requisitos são quase sempre em com stress e urgência.

Nesta época começaram os estudos para criar um ciclo inteligente.

Fases de Desenvolvimento

Modelo Cascata

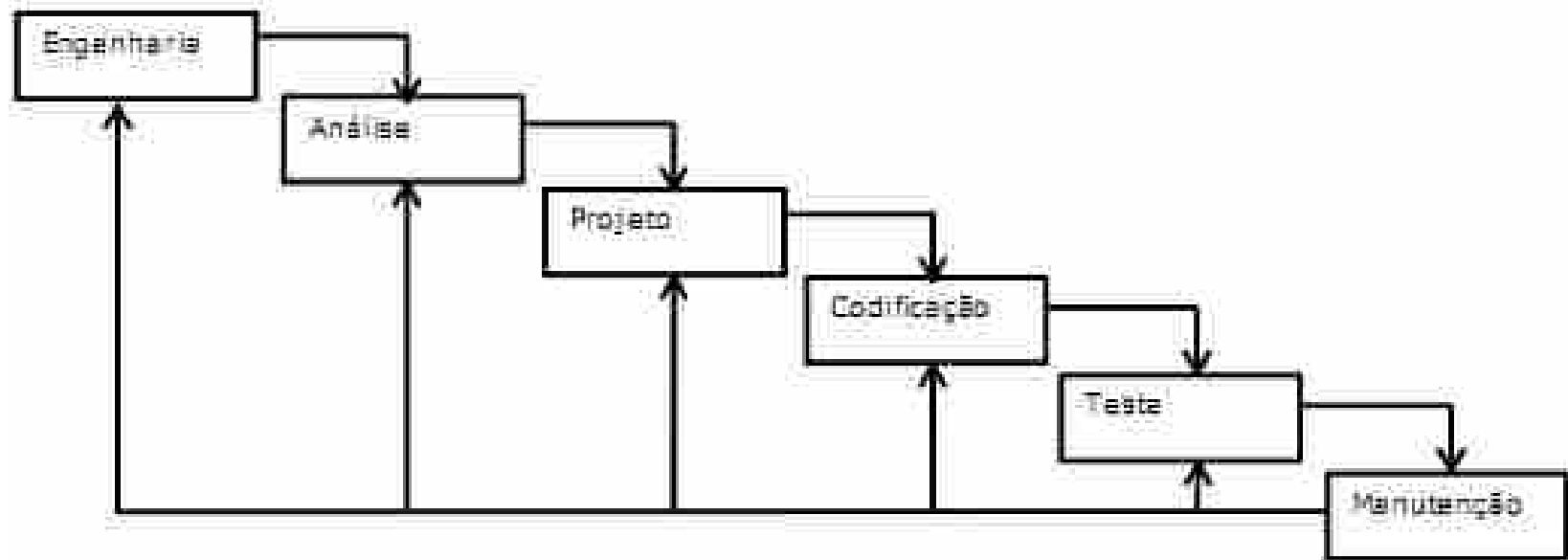
ou Ciclo de Vida Clássico
Características:

- Mais antigo
- Ordenação linear para realização das etapas
- Resultado de uma fase constitui a entrada da próxima

Fases de Desenvolvimento Modelo Cascata

ou Ciclo de Vida Clássico
Características:

- Ordenação linear para realização das etapas
- Resultado de uma fase constitui a entrada da próxima



Fases de Desenvolvimento

Modelo Cascata

- 1) Engenharia de Sistemas: estabelecem-se os requisitos gerais de todos os elementos do sistema computacional (*software*, *hardware*, pessoal, banco de dados, documentações e procedimentos).
- 2) Análise de requisitos: são levantados os requisitos específicos do *software*, de forma mais detalhada.
- 3) Projeto: são projetados a estrutura dos dados, a arquitetura do *software*, os detalhes procedimentais e de interface.
- 4) Codificação: concretiza-se o projeto, sendo criado o *software* utilizando uma linguagem de programação.
- 5) Testes: uma vez gerado o código, este deve ser testado no que diz respeito a aspectos internos e externos.
- 6) Manutenção: refere-se às mudanças (correções, adaptações e expansões) que o *software* sofrerá depois de ser entregue ao cliente.

Fases de Desenvolvimento

Modelo Cascata

De acordo com Pressman (2006), duas diretivas que norteiam o desenvolvimento, segundo o modelo **Cascata**, são:

- 1) Todas as etapas definidas no modelo devem ser realizadas, isso porque, em projetos de grande complexidade, a realização formal das etapas determinará o sucesso ou não do desenvolvimento.
- 2) Respeitar rigorosamente a ordenação das etapas na forma como foi apresentada.

Fases de Desenvolvimento

Modelo Cascata

Apesar de ser mais popular, o modelo possui algumas limitações:

- projetos reais raramente seguem o fluxo sequencial que o modelo propõe;
- há dificuldade em estabelecer explicitamente todos os requisitos logo no início, pois, no começo dos projetos, sempre há uma incerteza natural;
- paciência do cliente, porque uma versão executável do *software* só será disponível em uma etapa avançada do desenvolvimento.

Fases de Desenvolvimento

Modelo de Prototipação

Características

- Mesmo objetivo de uma maquete.
 - Visa melhorar o entendimento entre analista e cliente, reduzindo incertezas.

Fases de Desenvolvimento

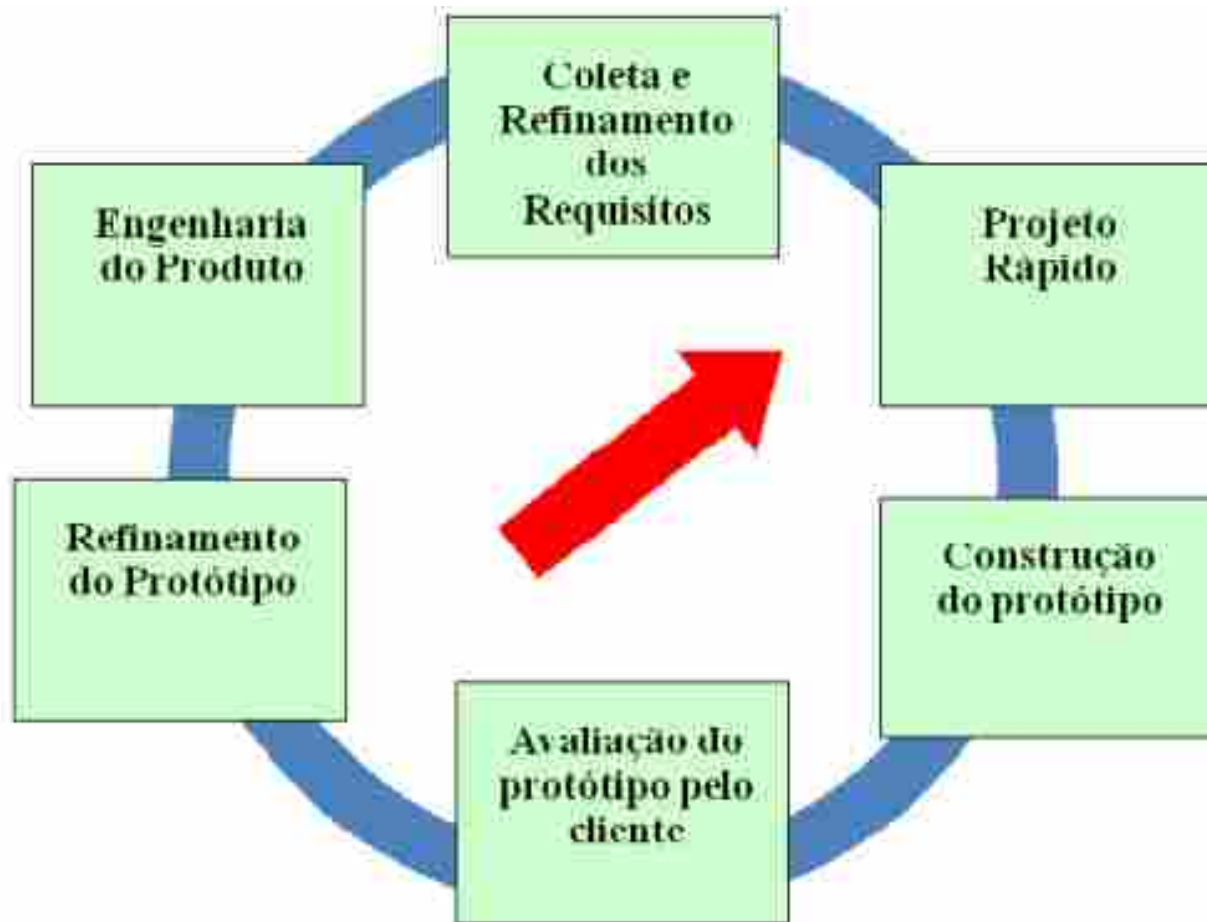
Modelo de Prototipação

Etapas

- Desenvolvimento de um protótipo com base no conhecimento dos requisitos iniciais para o sistema.
- Testes e experimentos devem ocorrer com usuários. Nesta etapa requisitos são revisados, com alterações detalhadas e documentadas.
- Alterações são apresentadas e a etapa anterior é repetida até a entrega do sistema.

Fases de Desenvolvimento

Modelo de Prototipação



Fases de Desenvolvimento

Modelo de Prototipação

Alguns problemas

- 1) O cliente não sabe que o *software* que ele vê não considerou, durante o desenvolvimento, a qualidade global e a manutenibilidade a longo prazo.
- 2) O desenvolvedor, frequentemente, faz uma implementação comprometida (utilizando o que está disponível) com o objetivo de produzir rapidamente um protótipo.
- 3) Os protótipos não são sistemas completos e deixam a desejar em alguns aspectos, como, por exemplo, na interface com o usuário.
- 4) Os esforços de desenvolvimento são concentrados, especialmente, nos algoritmos que implementam as principais funções associadas aos requisitos apresentados, a interface pertence a esse nível, o qual é considerado parte supérflua do desenvolvimento, o que permite caracterizar essa etapa por um custo relativamente baixo.

Fases de Desenvolvimento

Modelo Incremental ou Interativo

Características

- um sistema deve ser desenvolvido de forma incremental;
- cada incremento adiciona ao sistema novas funcionalidades, até a obtenção do sistema final.

Vantagem

- o sistema é testado durante cada fase de desenvolvimento

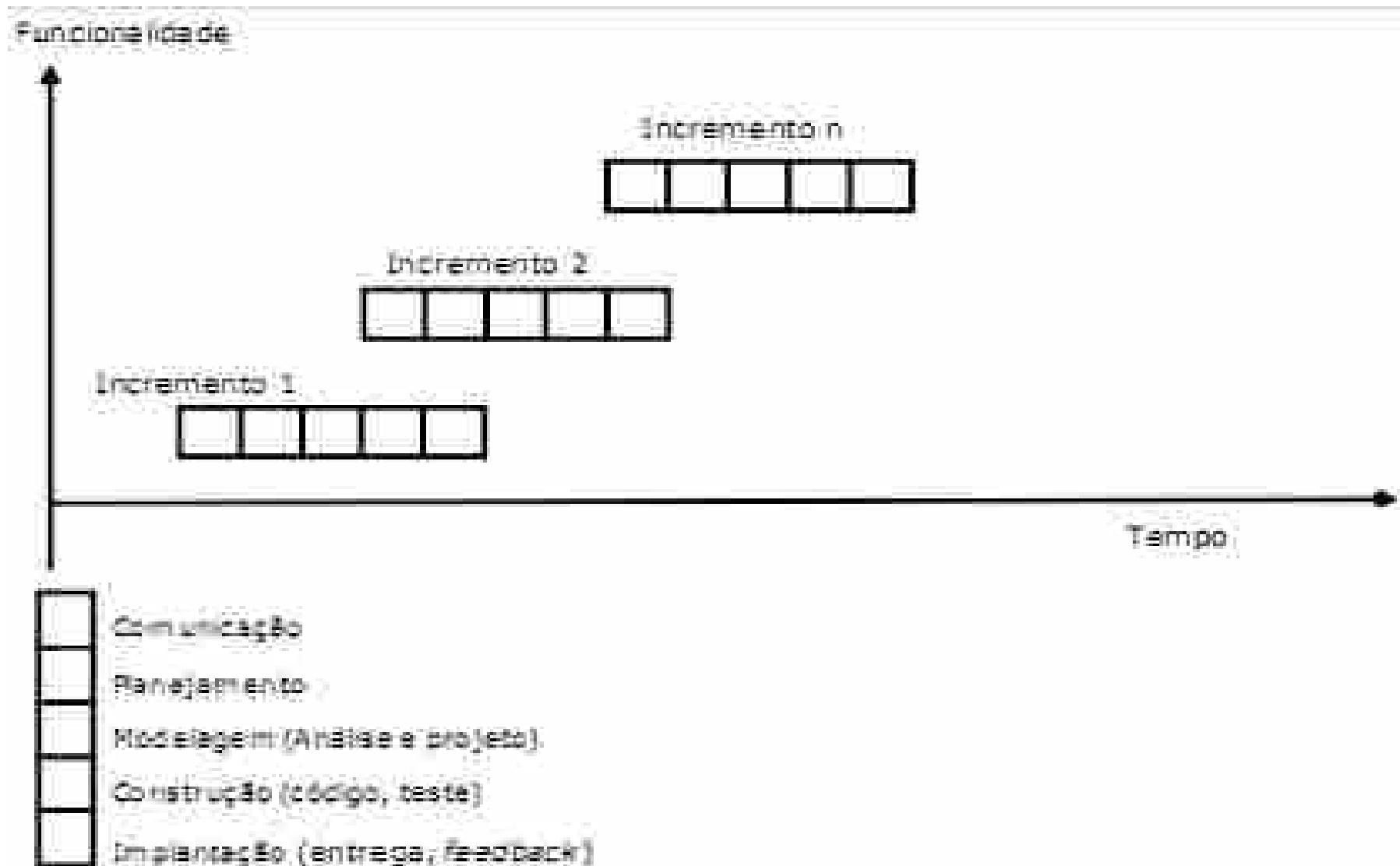
Fases de Desenvolvimento

Modelo Incremental ou Iterativo

- Utilizado o conceito de versões.
- Após a análise do documento de requisitos o sistema é segmentado por subsistemas e funcionalidades.
- A partir de um pequeno subsistema funcional, a cada versão, são adicionadas novas sub-funcionalidades.

Fases de Desenvolvimento

Modelo Incremental



Fases de Desenvolvimento

Modelo Espiral

Características

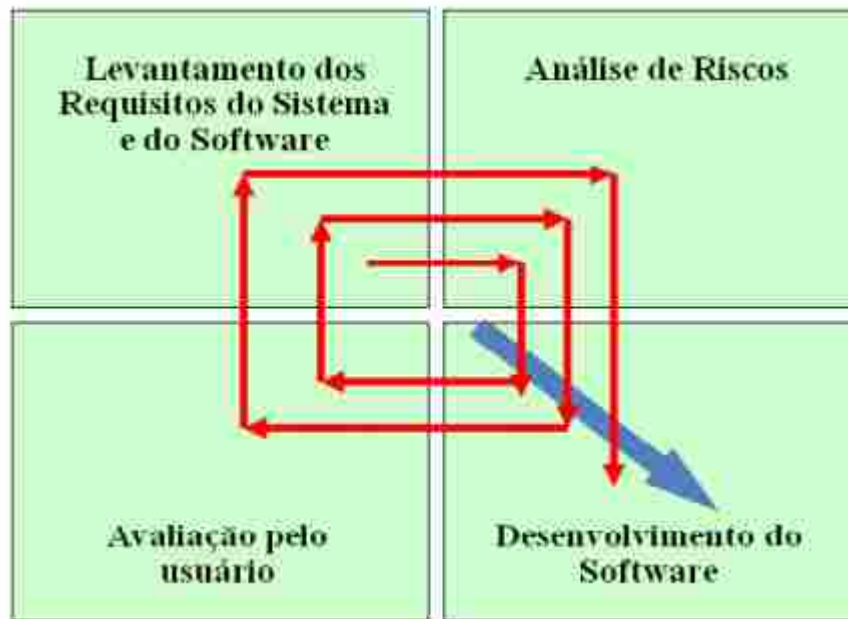
- Se confunde com prototipação.
- Mais adequado para sistemas complexos.

Fases de Desenvolvimento

Modelo Espiral

Características

associa as características do modelo de prototipação e do modelo cascata



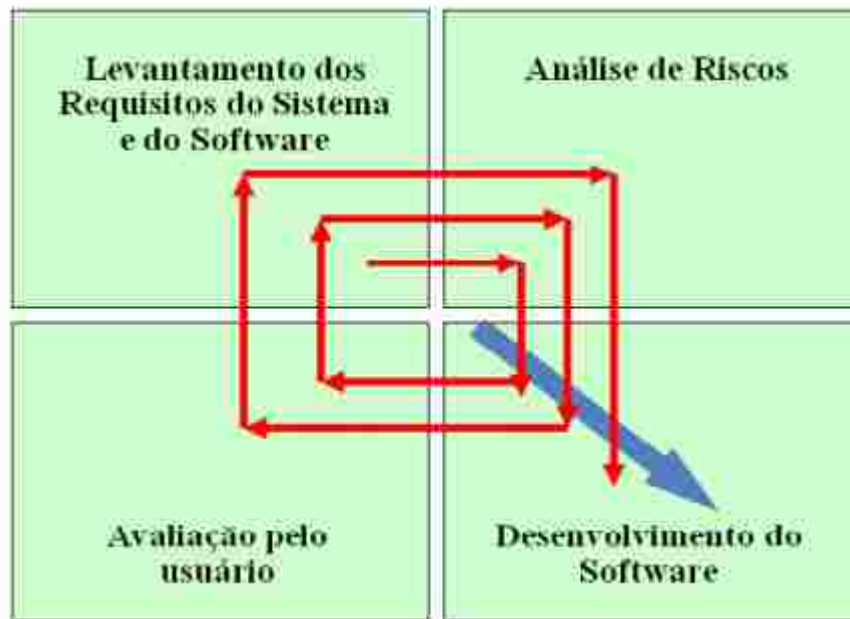
Parte interna se relaciona com a viabilidade do sistema. A análise de riscos é sempre uma atividade constante.

Fases de Desenvolvimento

Modelo Espiral

Setores:

Levantamento dos Requisitos do Sistema e do Software (ou Ativação):



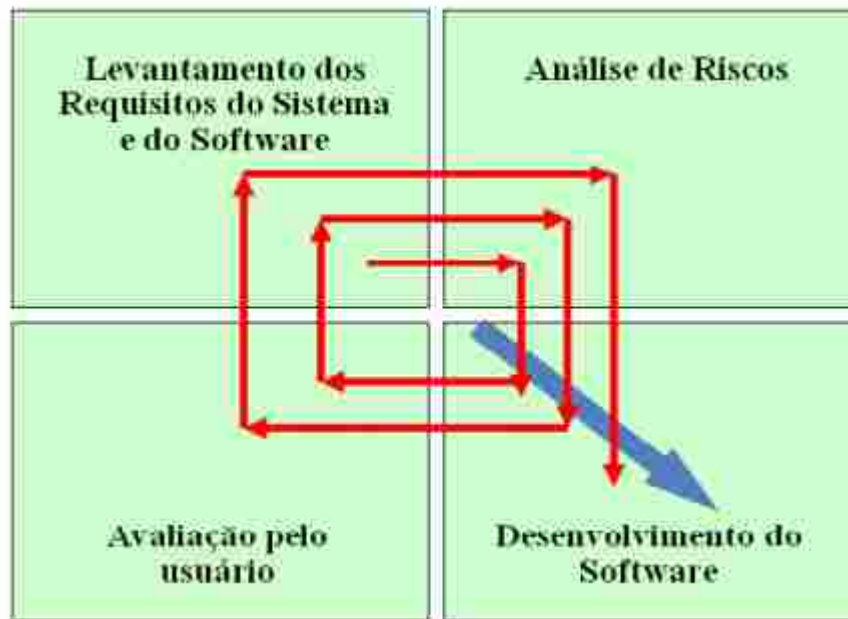
- Quais objetivos específicos?
- Quais as restrições?
- Plano de gerenciamento é construído.
- Riscos começam a ser identificados, porém sem análise dos mesmos.

Fases de Desenvolvimento

Modelo Espiral

Setores:

Análise de Riscos:



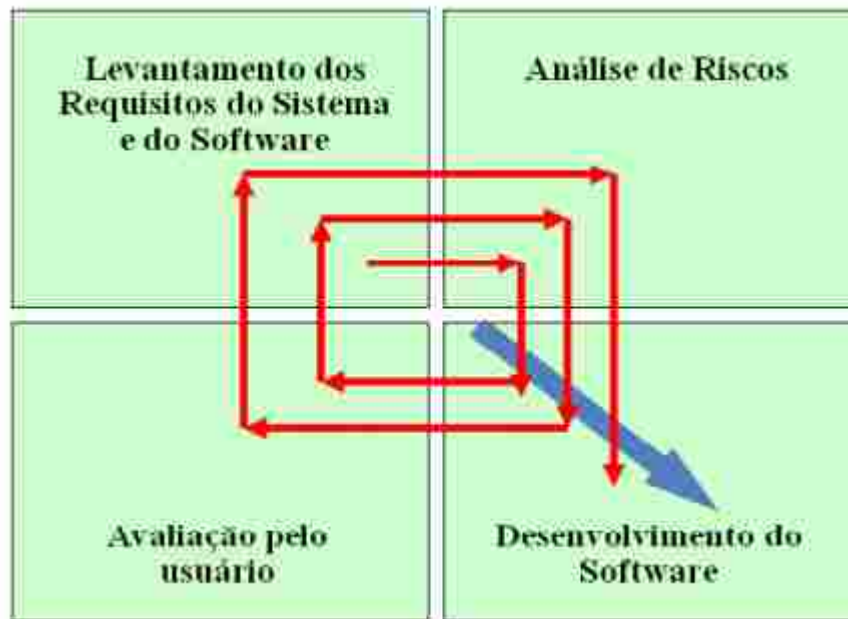
- São feitas análises detalhadas dos riscos identificados.
- Quais providências devem ser tomadas para amenizar os riscos?
- Nesta fase é recomendada a criação de vários protótipos para apoiar as análises.

Fases de Desenvolvimento

Modelo Espiral

Setores:

Desenvolvimento:



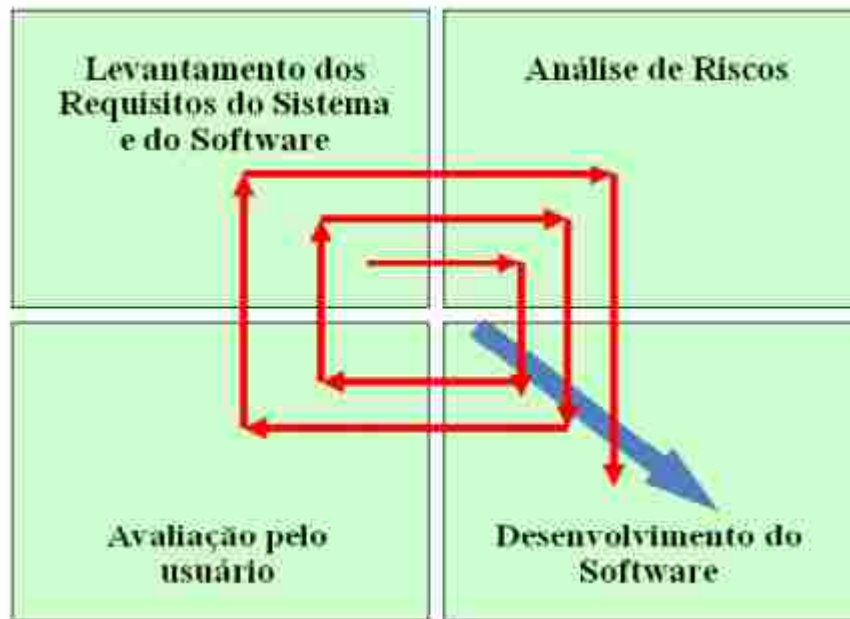
- Sempre fundamentado nas análises anteriores, o modelo mais adequado é selecionado.
- A bagagem dos profissionais envolvidos devem ser consideradas na estratégia.
- Sistemas de alta complexidades quase sempre necessitam da presença de um consultor especialista.

Fases de Desenvolvimento

Modelo Espiral

Setores:

Avaliação pelo Usuário ou Planejamento:



- Fase de revisão de projeto.
- A decisão de um novo espiral é tomada aqui.
- Caso uma nova espiral for realizada, um novo plano é traçado.