

Análise e Desenvolvimento de Sistemas Engenharia de Software I

Modelos de Processo de Engenharia de Software

Agenda

- Desenvolvimento de software
- Modelos de processo de software
- Modelos prescritivos de processo

Agenda

- Desenvolvimento de software
- Modelos de processo de software
- Modelos prescritivos de processo

Desenvolvimento de software

■ Um problema ...

- A Direção do **Colégio ABC** solicitou o **desenvolvimento** de um **software** para **possibilitar** o **controle** de **uso** e **acesso** ao **laboratório** de **informática** por **alunos** e **professores**. O **diretor acadêmico** espera **obter**, por meio deste software, as seguintes **informações**:
- **Quantidade** de **aulas práticas** ministradas em um **intervalo** de **datas** ou em uma **data específica**, com os **respectivos subtotais** por **curso, turma e Professor**.
- **Além disso**, o **diretor** pediu que o **sistema** possibilite **assegurar** a **emissão** de uma **relação** dos **alunos** que **participaram** de uma **aula**, o **agendamento** de uso de um **laboratório** e que possibilite **verificar** quem **usou** um **equipamento** em **determinado tempo**.
- **Pergunta-se:** O que deve ser feito para construir este software?

Desenvolvimento de software

□ **Como resolver um problema?**

-Passos

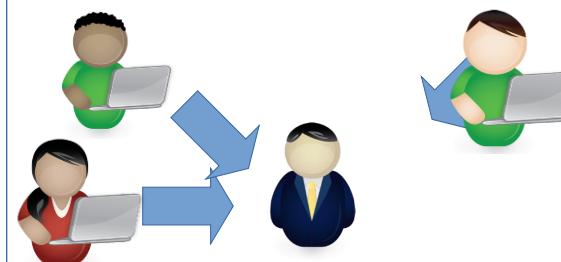
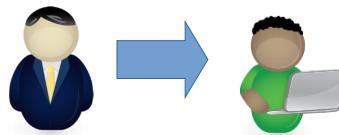
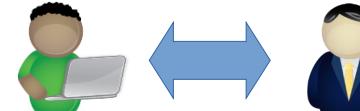
- 1) Entender** qual é o **problema** (comunicação e análise);
- 2) Planejar** a **solução** (modelagem e projeto do software);
- 3) Executar** o **plano** (criação do código);
- 4) Examinar** a exatidão dos resultados (testes e controle de qualidade).

-Princípio W5HH – auxilia na organização do plano

- **Why** – **por que** o sistema está sendo desenvolvido?
- **What** – **o que** deverá ser feito?
- **When** – **quando**?
- **Who** – **quem** é responsável por cada função?
- **Where** – **onde** estão organizacionalmente localizadas?

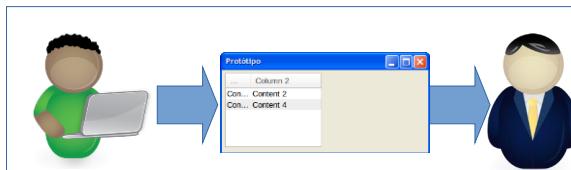
Desenvolvimento de software

Projeto de software ideal

 <p>O cliente possui um problema e acredita que um software possa resolvê-lo</p>	 <p>Ele pede propostas para você e outros desenvolvedores</p>	 <p>Cada desenvolvedor cria uma proposta prometendo software de qualidade e com preço competitivo</p>
 <p>O cliente seleciona um desenvolvedor para realizar o trabalho. Você ganha a o projeto.</p>	 <p>O projeto tem início e uma equipe é organizada para desenvolvê-lo.</p>	 <p>Define-se claramente as necessidades do cliente → especificação do problema → base para detalhar o plano do projeto que guiará as atividades subsequentes.</p>

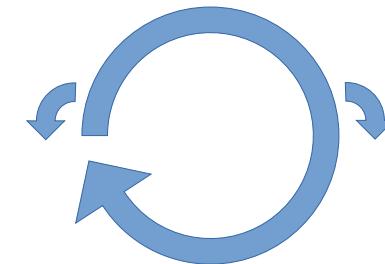
Desenvolvimento de software

Projeto de software ideal

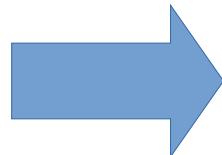


Um protótipo é construído para demonstrar que o problema foi entendido e ajudar o cliente a visualizar o que ele terá no término do desenvolvimento

Tem-se então o projeto (*design*) do software e em paralelo decisões sobre outros recursos e ajuste do plano do projeto

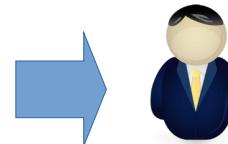


Conforme partes do projeto são validadas, elas são programadas e documentadas



As partes programadas são testadas e integradas. Alterações são esperadas e controladas por um sistema de controle de mudanças

O software está pronto para o cliente, testes do sistema são executados e erros são insignificantes e prontamente consertados



O sistema, documentação e manual do usuário estão prontos. O software é apresentado ao cliente e ele faz seu aceite formal.

Desenvolvimento de software

- **Projetos reais**
 - **Mudanças são corriqueiras ...**
- **Requisitos mudam;**
- **Projeto (design) muda;**
- **Tecnologia muda;**
- **Sociedade muda;**
- **Pessoas mudam;**
- **Correções devem ser realizadas.**
- **Mudanças devem ser gerenciadas e não eliminadas!**

Agenda

- Desenvolvimento de software
- Modelos de processo de software
- Modelos prescritivos de processo

Modelos de processo de software

■ Lembrando ...

- **Engenharia de Software** tem como **objetivo** a **produção economicamente viável** de **software** que seja **confiável** e que **execute eficientemente** em máquinas reais;
- **Engenharia de Software** é uma **tecnologia em camadas**:



Modelos de processo de software

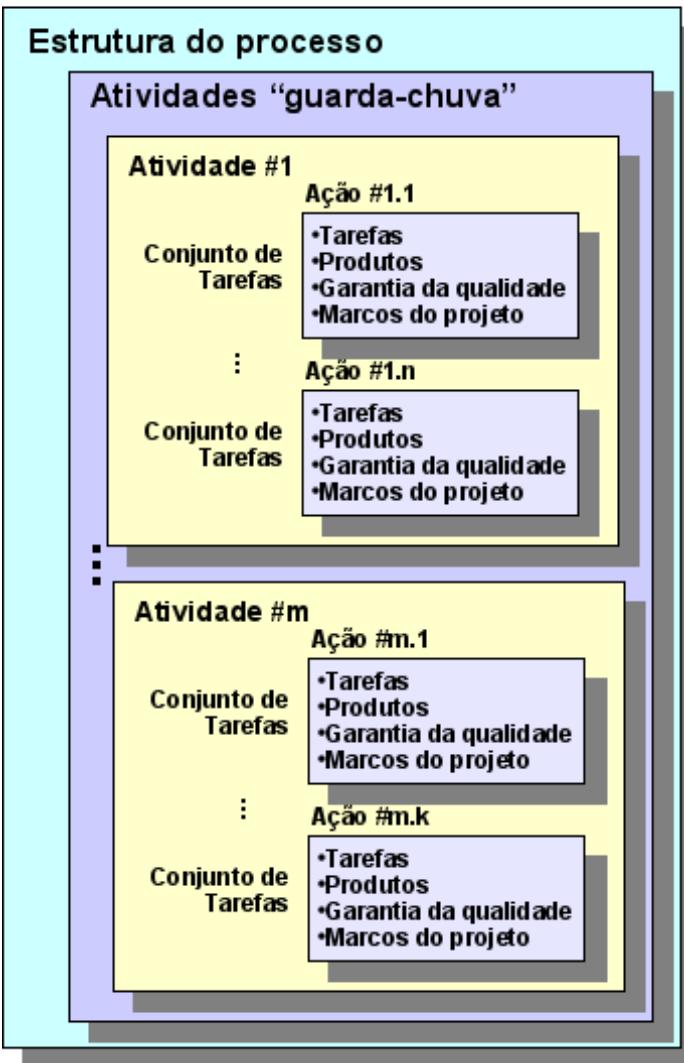
■ Lembrando ...

- **Modelo de processo** define em **etapas** as **atividades** que, após sua **execução**, encaminharão o **desenvolvimento** do **software** de sua **concepção inicial** até sua **implantação, uso e retirada**;
- **Modelo de processo** é também conhecido como **modelo de ciclo de vida** ou ainda **paradigma de desenvolvimento de software**;
- **Atividades genéricas** de todo processo:
 - *Comunicação, Planejamento, Modelagem, Construção e Implantação.*
- **Atividades de suporte** (“guarda-chuva”) de todo processo:
 - *Gestão do projeto, Gestão de riscos, Revisões técnicas, Garantia da qualidade do software, Gestão da configuração do software, Preparação e produção dos produtos do trabalho, Gestão da reutilização, Medições.*

Modelos de processo de software

■ Visão geral da estrutura de um processo de software

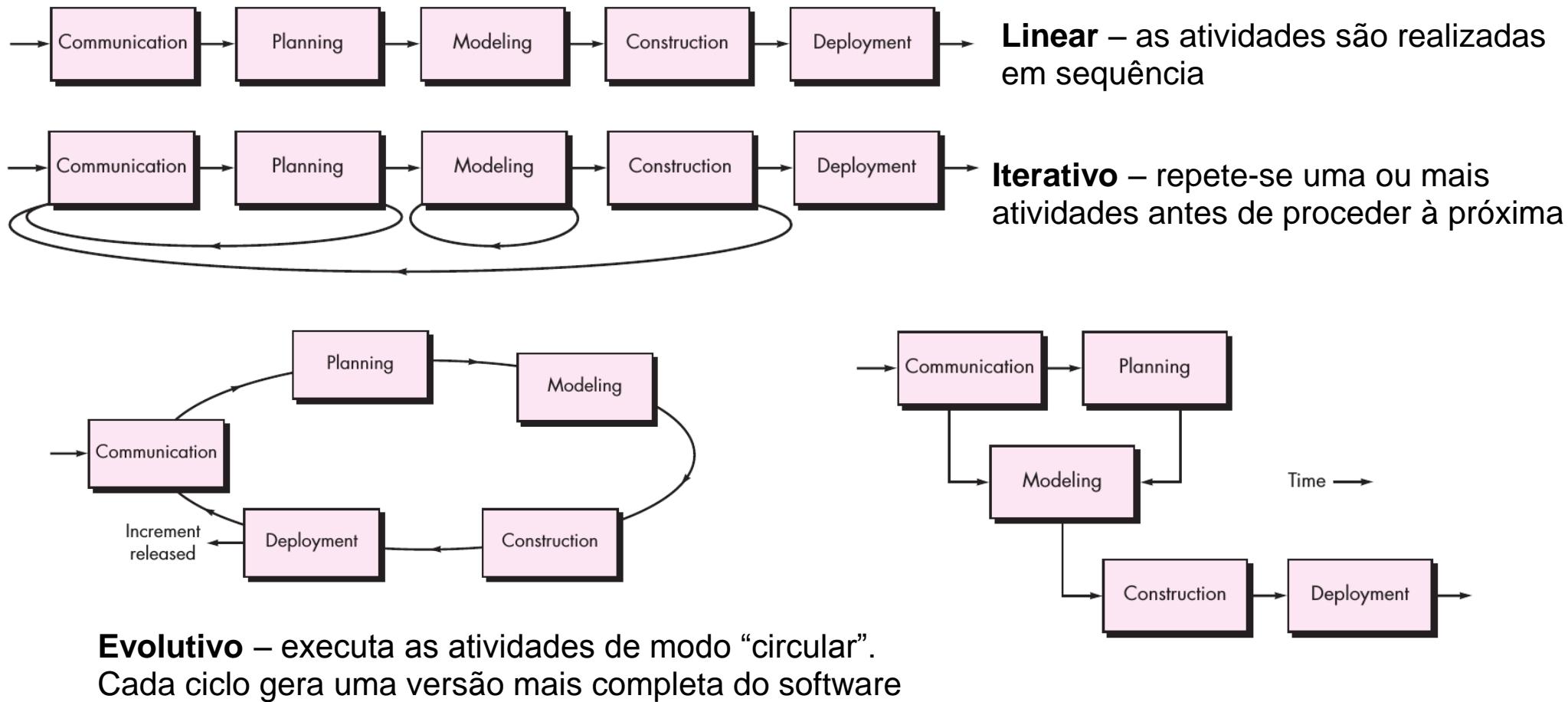
Processo de software



- Um **processo** é uma **coleção** de **atividades** de trabalho, **ações** e **tarefas** que são **executadas** quando algum **produto** precisa ser **criado**;
- Cada **atividade** é **populada** por um **conjunto de tarefas** que identifica **como** as **tarefas** serão **completadas**, os **produtos** de **trabalho** a serem produzidos, **pontos** de **qualidade** de software que serão requeridos e **marcos** de **projeto** para **indicar** o **progresso**.

Modelos de processo de software

■ Possíveis fluxos das atividades de um processo genérico



Modelos de processo de software

- Visão geral da estrutura de um processo de software
 - Definição das atividades
- Depende
 - Da **natureza** do problema a ser resolvido;
 - Das **características** das pessoas que realizarão o trabalho;
 - Das **partes interessadas** no projeto.
- Exemplo de atividade: **Comunicação**
 - **Pequeno projeto:** pode ser resolvida apenas com **ações simples** como ligações telefônicas entre o cliente e o desenvolvedor, com troca de e-mails para aprovação final.
 - **Projeto complexo:** a mesma atividade pode requerer **várias ações** relacionadas a requisitos do software tais como *concepção, elicitação, elaboração, negociação, especificação e validação*.¹⁴

Modelos de processo de software

- Visão geral da estrutura de um processo de software
 - Definição dos conjuntos de tarefas
 - São **definidas** de acordo com as **ações necessárias**;
 - Por exemplo, a **ação Elicitação de Requisitos (atividade de Comunicação)** de um projeto pequeno, o **conjunto de tarefas** poderia ser assim definido:
 - 1) Elaborar uma **lista** dos **interessados** no projeto;
 - 2) **Convidar** os **interessados** para uma reunião informal;
 - 3) **Pedir** a cada interessado uma **lista de características e funções** que deve ser contemplado pelo software;
 - 4) **Discutir** os requisitos e criar uma **lista final**;
 - 5) **Priorizar** os requisitos;
 - 6) **Marcar possíveis** áreas de **incertezas** nos requisitos.
- Em **projeto mais complexo**, este conjunto pode ser maior ...

Modelos de processo de software

■ Características de um modelo de processo

- **Entendimento:** representa a facilidade de compreender o modelo e sua aplicabilidade;
- **Visibilidade:** representa a capacidade de acompanhar externamente o progresso do projeto;
- **Suportabilidade:** representa a facilidade de se utilizar ferramentas CASE para auxiliar o processo;
- **Aceitabilidade:** representa a facilidade do modelo ser aceito por seus praticantes;
- **Confiabilidade:** representa a facilidade de descobrir erros antes que eles sejam descobertos no produto final, no cliente;
- **Robustez:** representa a capacidade do modelo em se manter mesmo na presença de problemas inesperados;

Modelos de processo de software

- **Características de um modelo de ciclo de vida (cont.)**
 - **Sustentabilidade:** a capacidade do modelo evoluir para abrigar mudanças que são reflexos de necessidades organizacionais;
 - **Rapidez:** representa quanto rápido um produto pode ser obtido utilizando o modelo.

Modelos de processo de software

■ Avaliação e melhoria de processos de software

- Apenas a adoção de um modelo de processo não garante o sucesso da Engenharia de Software;
- O próprio processo precisa ser avaliado se cumpre com os critérios básicos de processo para poder garantir o sucesso da Engenharia de Software;

- Abordagens de avaliação de processos de software:

- *Standard CMMI Asessment Method for Process Improvement (SCAMPI);*
- *CMM-Based Appraisal for Internal Process Improvement (CBA IPI);*
- **SPICE (ISO/IEC 15504);**
- **ISO 9001:2000 para software.**

Agenda

- Desenvolvimento de software
- Modelos de processo de software
- Modelos prescritivos de processo

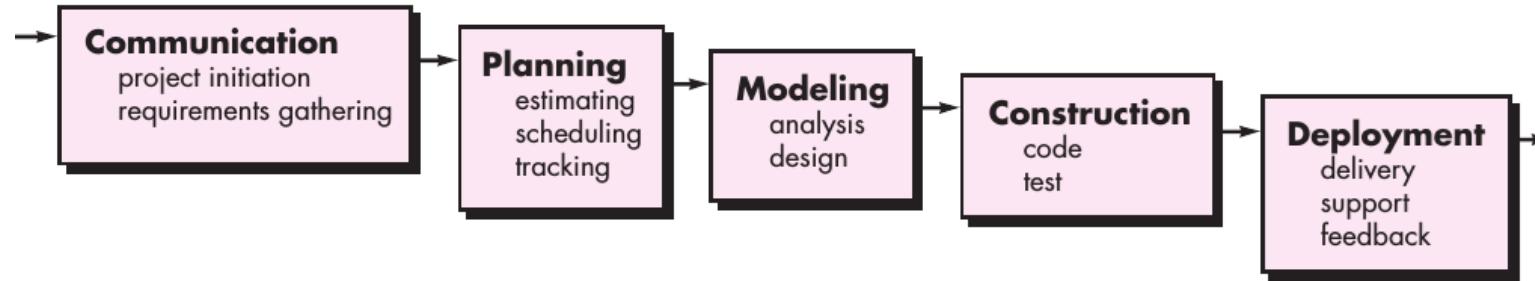
Modelos prescritivos de processo

- **Vantagens de se estudar modelos de processos**
 - Proporcionam um **entendimento comum** dos seus **elementos**;
 - **Servem como uma base** na qual o **processo real** a ser **aplicado** é **inspirado e ajustado** de acordo com as **necessidades** do **projeto**;
 - **Modelos prescritivos: determinam a ordem das atividades** do **modelo**. Serão estudados:
- Modelo “em **Cascata**” ou Clássico;
- Modelos **Incrementais**;
- Modelos **Evolucionários**
- Modelo de **Prototipação**;
- Modelo **Espiral**.

Lembrando sempre que:
□ **Mudanças** são **corriqueiras**;
□ Logo a estrutura **não** deve ser **muito rígida** (senão não se admite mudanças);
□ **Nem muito solta** (senão não há coordenação).

Modelos prescritivos de processo

Modelo “em Cascata” ou Clássico



Útil quando se domina bem o problema a ser resolvido

-Pontos positivos

- É de fácil compreensão para os **desenvolvedores**;
- É de fácil compreensão para os **clientes**;
- Torna explícito produtos intermediários que serão necessários para a próxima etapa;
- É utilizado como **base** para outros processos similares.

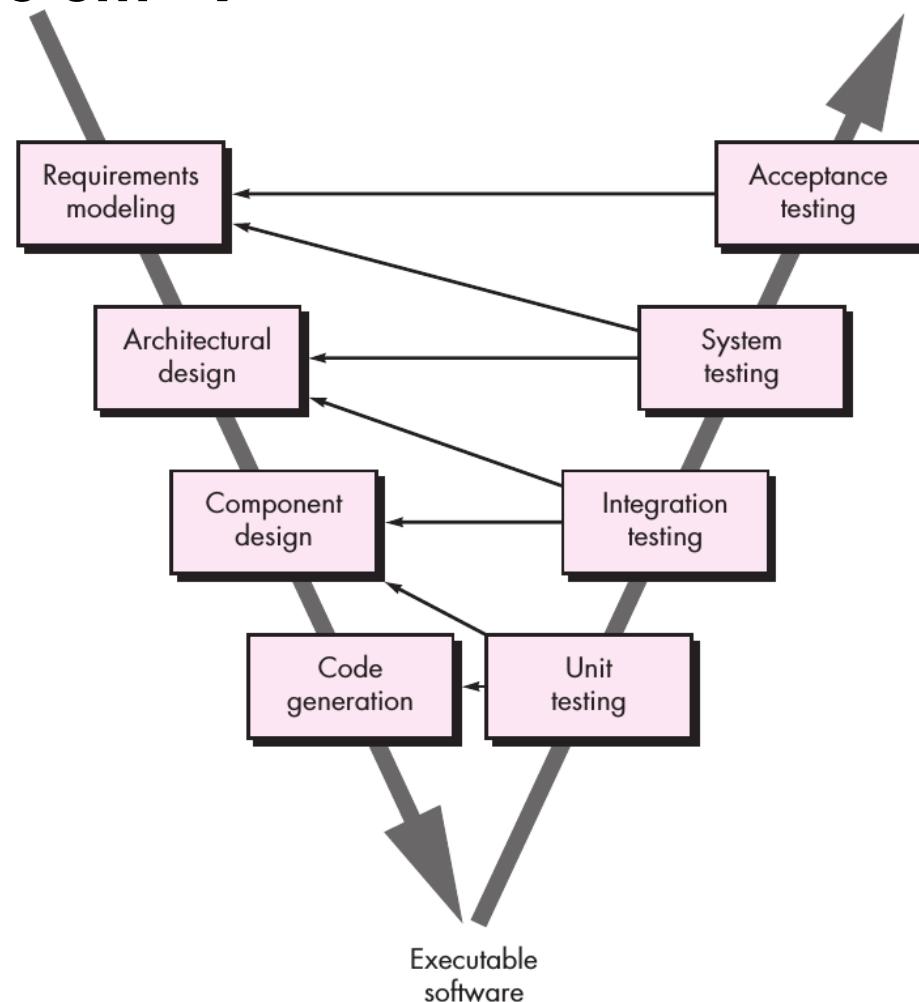
-Pontos negativos

- Não suporta **iterações**;
- Próximo da construção de **hardwares** (visão de manufatura);
- É difícil acomodar mudanças;
- Focaliza documentos e artefatos;
- Demora para que o cliente experimente o produto!
- Projetos reais raramente são puramente sequenciais!

Modelos prescritivos de processo

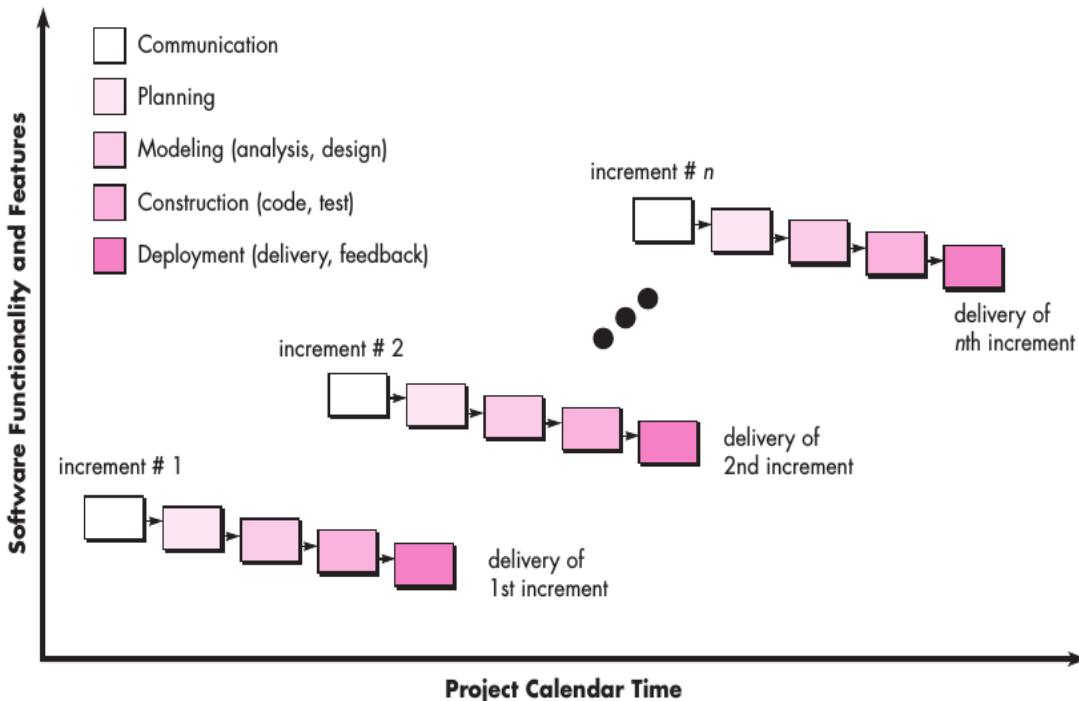
Modelo “em Cascata” ou Clássico

- Modelo em “V”



É uma **variação** do modelo “em cascata” que adiciona ações de **garantia da qualidade** (**verificação e validação**) às **atividades** do processo.

Modelos prescritivos de processo

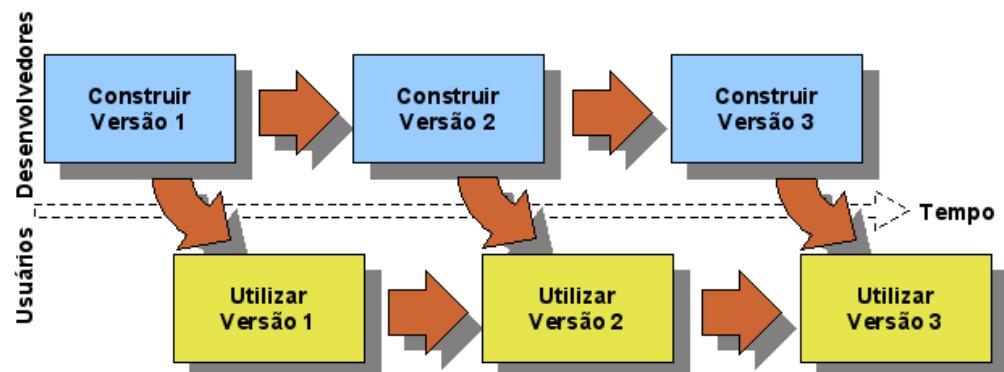


Características

- Combina elementos de fluxo de processo linear e paralelo;
- Desenvolve uma série de versões do software – *incrementos* – que cada vez mais acrescentam funcionalidades para o cliente.

Pontos positivos

- Muito utilizado;
- O treinamento do sistema pode começar rapidamente;
- Pode-se descobrir novos mercados com funcionalidades que ainda não existem;
- Melhoria contínua com “feedback” dos usuários;
- Facilita a divisão do trabalho por equipes especializadas.



Modelos prescritivos de processo

■ Modelos Incrementais

- Exemplo

- Um **software** de **edição** de **texto** poderia ser **desenvolvido** assim, de modo **incremental**:
 - **Incremento #1**: operações de arquivo, funções de edição e produção de documentos;
 - **Incremento #2**: capacidades avançadas de edição e produção de documentos;
 - **Incremento #3**: verificador ortográfico e gramatical;
 - **Incremento #4**: capacidades avançadas de layout de página.
- **NOTA**: os **incrementos** podem ser **desenvolvidos** de modo **evolutivo** com **prototipação**.

Modelos prescritivos de processo

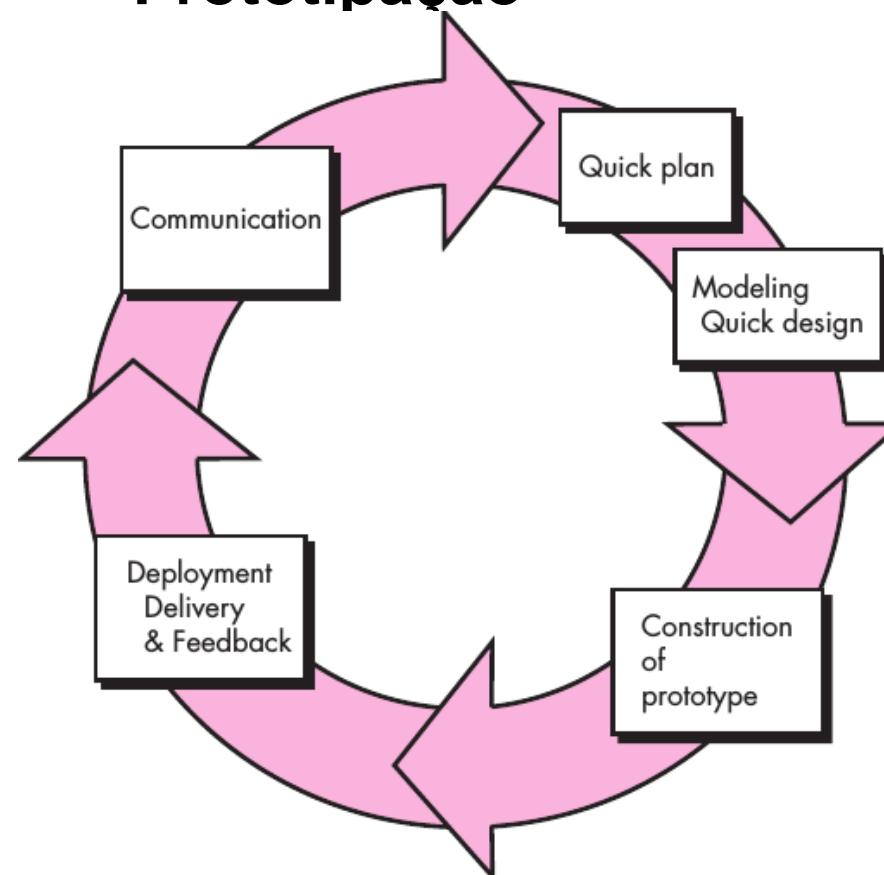
Modelos Evolucionários

- É comum o **software em desenvolvimento** sofrer **pressões** para ser **rapidamente lançado ao mercado**;
- Assim, **parte-se** para a **construção** de um **produto base** (*core product*), de **requisitos bem definidos**, mas cuja **extensão** ainda **não se conhece** completamente;
- Assim, é necessário um **modelo de processo** para **construir** um **produto** que irá **evoluir** com o **tempo**;
- **Modelos evolutivos** empregam **modelos de fluxo iterativo** que produzem **progressivamente** uma **versão mais completa** do **software** com cada **iteração** – permitem acomodar a evolução do software;
- Dois **modelos evolucionários** muito **utilizados**: Prototipação e Modelo Espiral.

Modelos prescritivos de processo

■ Modelos Evolucionários

- Prototipação



- Pontos positivos

- Permite que **todo ou parte** do sistema seja **construído rapidamente** (similar a protótipos de Engenharia);
- Permite uma **maior comunicação** entre desenvolvedores, usuários e cliente;
- **Reduz riscos e incertezas** do projeto.

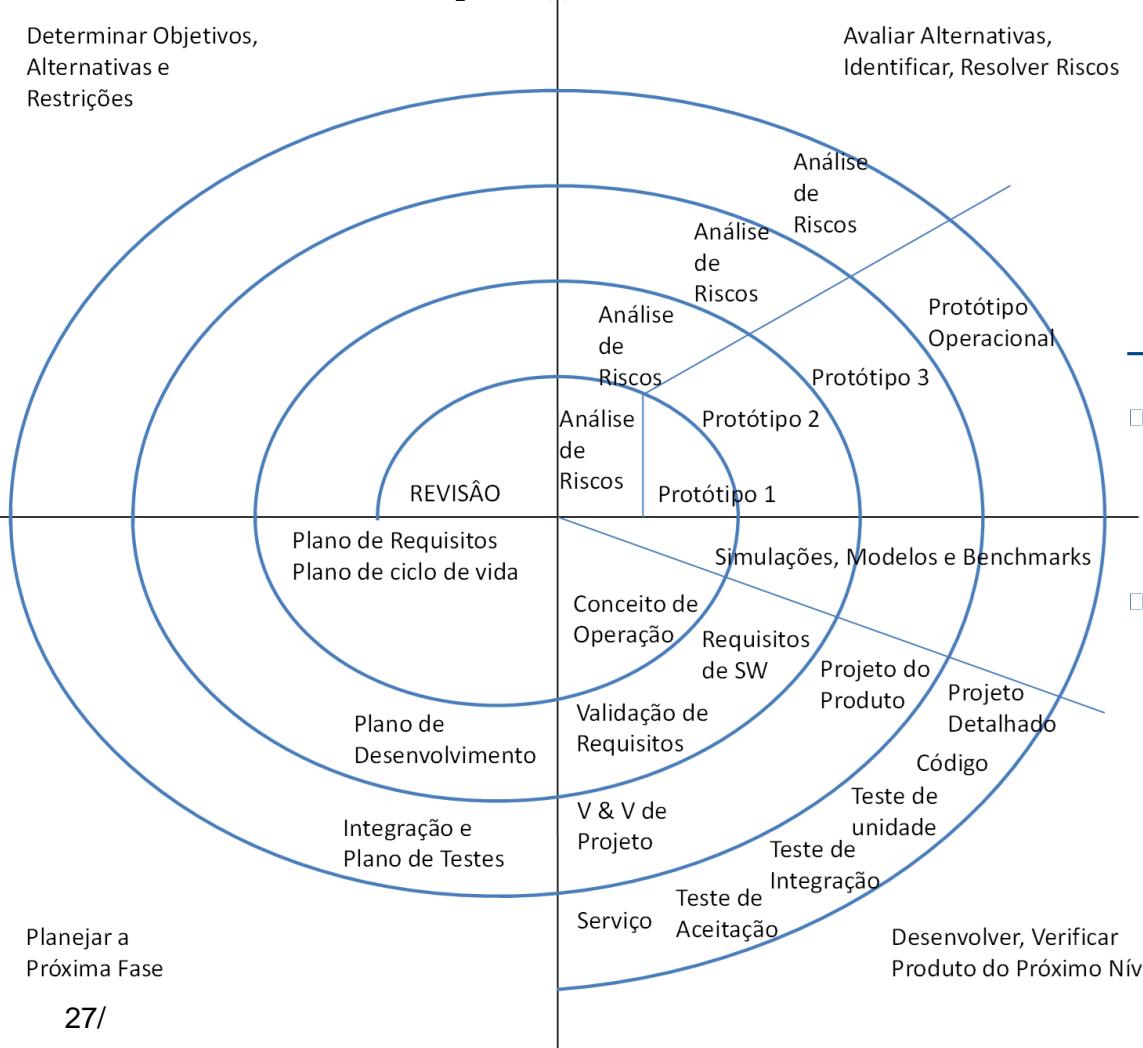
- Pontos negativos

- **Falta de visibilidade** no processo;
- **Sistemas produzidos** podem ser **mal estruturados**;
- É necessário **conhecimento** técnico de prototipagem para se envolver no processo;
- Tende a gerar **documentação de baixa qualidade**.

Modelos prescritivos de processo

Modelos Evolucionários

- Modelo Espiral



- Pontos positivos

- Boa visibilidade do processo;
- Herda os **pontos positivos** dos **modelos de desenvolvimento iterativos** e de **prototipagem**.

- Pontos negativos

- Pode ser **difícil convencer os clientes a utilizar este modelo evolucionário** (questões de controle);
- Exige o conhecimento de como avaliar riscos.**

Referências bibliográficas

- BOEHM, B. **Anchoring the software process.** IEEE Software, v. 13, n. 4, p. 73–82, jul. 1996.
- METZGER, P. W.; BODDIE, J. **Managing a programming project: processes and people.** [s.l.] Prentice Hall PTR, 1996.
- PRESSMAN, R. S. **Software engineering: a practitioner's approach.** 7. ed. New York: McGraw-Hill Higher Education, 2010.
- SOMMERVILLE, I. **Software Engineering.** 9. ed. Boston, MA: Pearson, 2011.