



**UNIVERSIDADE
FEDERAL RURAL
DE PERNAMBUCO**

**UNIDADE
ACADÊMICA DE
SERRA TALHADA**

Processo, Modelos e Modelo Cascata

Ygor Amaral <ygor.amaral@ufrpe.br>

Disciplina: Processo de Desenvolvimento de Software

Curso: Sistemas de Informação (2016.1)

Modelos de processo



- Antes de falar do modelo cascata
 - Vamos falar um pouco sobre engenharia de software, processo e modelo de processo....



**UNIVERSIDADE
FEDERAL RURAL
DE PERNAMBUCO**

**UNIDADE
ACADÊMICA DE
SERRA TALHADA**

Engenharia de Software

Engenharia de Software



- O que é engenharia de software?

Engenharia de Software



- O que é engenharia de software?
- “É o estabelecimento e o emprego de sólidos **princípios** da engenharia de modo a obter software de **maneira econômica**, que seja **confiável** e funcione de forma **eficiente** em máquinas reais.” (Pressman, 2011)

As camadas...

Engenharia de Software



Camadas da engenharia de software. Pressman, 2011

Foco na qualidade

- Qualquer abordagem de engenharia deve estar fundamentada em um **comprometimento com a qualidade**
- Promover a cultura de aperfeiçoamento contínuo de processos
 - **Leva ao desenvolvimento de abordagens cada vez mais efetivas na engenharia de software**
- A pedra fundamental que sustenta a engenharia de software é o **foco na qualidade**

As camadas...

Engenharia de Software



Camadas da engenharia de software. Pressman, 2011

Processo

- O foco da nossa disciplina!
- É **a base** para a engenharia de software!
- É **a liga** que mantém as camadas de tecnologia coesas e possibilita o desenvolvimento de software de **forma racional** e dentro do **prazo**
 - O elo de ligação entre as ferramentas e os métodos

Processo

- Define uma metodologia que deve ser estabelecida para a entrega efetiva
- Constitui a base para o controle do gerenciamento de projetos de software
- Aplicada métodos para produzir:
 - Modelos
 - Documentos
 - Dados
 - Relatórios
 - ...

Processo

- Estabelece **marcos**
- A **qualidade** é garantida
- **Mudanças** são geridas de forma apropriada

Processo

- Um processo define:
 - A sequência em que os métodos serão aplicados
 - Quais os responsáveis por cada tarefa
 - Quando e como o software será entregue
 - Possibilitam aos gerentes de software avaliar o progresso do desenvolvimento

As camadas...

Engenharia de Software



Camadas da engenharia de software. Pressman, 2011

Métodos

- Fornecem as **informações técnicas** para desenvolver software
- Envolvem uma ampla gama de tarefas, que incluem:
 - **Comunicação**
 - **Análise de requisitos**
 - **Modelagem de projeto**
 - **Construção de programa**
 - **Testes**
 - **Suporte**

Métodos

- Baseiam-se em um conjunto de princípios básicos que governam cada área da tecnologia
 - Inclui atividades de modelagem e outras técnicas descritivas

Métodos

- Em resumo, métodos são:
 - Os detalhes de “como fazer” para construir o software
 - Através de métodos para as diferentes tarefas

As camadas...

Engenharia de Software



Camadas da engenharia de software. Pressman, 2011

Ferramentas

- Fornecem **suporte** automatizado ou semiautomatizado para o **processo** e para os **métodos**
- Quando as ferramentas são **integradas**, de modo que as informações criadas por uma ferramenta possam ser usadas por outra, é estabelecido um sistema para o suporte ao desenvolvimento de software
 - Denominado Engenharia de Software com auxílio do computador, ou, Ferramentas CASE (*Computer-Aided Software Engineering*)
- Cada atividade, pode ter uma ou mais ferramentas auxiliando



**UNIVERSIDADE
FEDERAL RURAL
DE PERNAMBUCO**

**UNIDADE
ACADÊMICA DE
SERRA TALHADA**

O Processo de Software

O processo de software



- Processo:

- É um conjunto de atividades, ações, tarefas realizadas na criação de algum produto de trabalho

- Atividade:

- Esforço para atingir um objetivo amplo
- É utilizada independentemente do campo de aplicação, do tamanho do projeto, da complexidade de esforços
 - Por exemplo: comunicar-se com os interessados

O processo de software



- Ação:
 - Envolve um conjunto de tarefas que resultam num artefato de software fundamental
 - Exemplo de ação: projeto de arquitetura
 - Algumas metodologias atuais, consideram atividade e ação como uma só definição
- Uma tarefa se concentra em um objetivo pequeno, porém bem definido
 - Por exemplo: realizar um teste de unidades
 - Produz um resultado

O processo de software

- Um processo não é uma prescrição rígida de como desenvolver um software
- É uma abordagem adaptável que possibilita às pessoas realizar o trabalho de selecionar e escolher o conjunto apropriado de ações e tarefas

O processo de software

- Uma metodologia (*framework*) de processo **estabelece o alicerce** para um processo de engenharia de software
 - Faz isso através de atividades estruturais aplicáveis a todos os projetos de software
- Uma metodologia de processo genérica para engenharia de software compreende cinco atividades:
 - Comunicação, planejamento, modelagem, construção, emprego

As cinco atividades genéricas



- Comunicação
 - É de vital importância comunicar-se e colaborar com o cliente e todos os interessados
 - A intenção é compreender os objetivos das partes interessadas para com o projeto
 - Para poder fazer o levantamento das necessidades que ajudarão a definir as funções e características do software

As cinco atividades genéricas



- Planejamento
 - Qualquer jornada complicada pode ser simplificada caso exista um mapa
 - Um projeto de software é uma jornada complicada!
 - Realizar o planejamento cria um "mapa" que ajuda a guiar a equipe na sua jornada
 - Esse mapa é conhecido como o plano de projeto de software, ele define todo o trabalho, descrevendo as tarefas técnicas a ser conduzidas, os riscos prováveis, os recursos que serão necessários, os produtos resultantes a ser produzidos e um cronograma de trabalho
 - Aprenderão a fazer isso em detalhes na próxima disciplina!

As cinco atividades genéricas



- Modelagem
 - Independentemente de ser um paisagista, um construtor de pontes um engenheiro aeronáutico ou um arquiteto, trabalha-se com modelos todos os dias!
 - Cria-se um "esboço", de modo que se possa ter uma ideia do todo
 - Para compreender melhor o problema e como resolvê-lo
 - Melhor entender as necessidades do software

As cinco atividades genéricas



- Emprego
 - O software é entregue ao cliente, que avalia e fornece um feedback
 - Seja a versão do software completo ou apenas um incremento parcial

Atividades de apoio

- Essas atividades metodológicas do processo são complementadas por uma série de atividades de apoio:
 - Controle e acompanhamento do projeto
 - Administração de riscos
 - Garantia da qualidade de software
 - Revisões técnicas
 - Medição
 - Gerenciamento da configuração de software
 - Gerenciamento da reusabilidade

Atividades de apoio



- Controle e acompanhamento do projeto
 - Possibilita que a equipe avalie o progresso em relação ao plano do projeto
 - Tome as medidas necessárias para cumprir o cronograma

Atividades de apoio

- Administração de riscos
 - Avalia riscos que possam afetar o resultado
 - Riscos também podem afetar a qualidade do produto resultante
 - Realiza o acompanhamento para prevenir que o risco se concretize em um problema
 - Caso se concretize, será necessário realizar uma ação corretiva

Atividades de apoio



- Garantia da qualidade de software
 - Define e conduz as atividades que garantem a qualidade do software
 - Normalmente essas atividades englobam auditorias periódicas de monitoramento

Atividades de apoio



- Revisões técnicas
 - Avaliam artefatos da engenharia de software, tentando identificar e eliminar erros antes que se propaguem para a atividade seguinte
 - Evitar aquele “*bug*” que quebra o sistema!

Atividades de apoio

- Medição
 - Define e coleta medidas
 - Do processo, do projeto e do produto
 - Auxilia na entrega do software de acordo com os requisitos
 - Pode ser usada com as demais atividades
 - Tenta responder vários questionamentos comuns:
 - Quanto cada programador produz?
 - Quanto a equipe produz?
 - Quanto tempo consigo entregar o software?
 - Qual o custo?

Atividades de apoio

- Gerenciamento da configuração de software
 - Gerencia os efeitos das mudanças ao longo do processo
 - Por exemplo:
 - Controle de versão
 - Git, svn, mercurial, entre outros...
 - Metodologia/técnicas para atualizar o sistema em “produção”

Atividades de apoio

- Gerenciamento da reusabilidade
 - Define critérios para o reúso de artefatos e estabelece mecanismos para obtenção de componentes reusáveis
 - Por exemplo: desenvolver componentes de software que serão reutilizados em diversos projetos
 - Utilizar *frameworks*, bibliotecas *open source*



**UNIVERSIDADE
FEDERAL RURAL
DE PERNAMBUCO**

**UNIDADE
ACADÊMICA DE
SERRA TALHADA**

Modelos de processo

Modelos de processo



- Um processo é representado por um MODELO prescritivo
- Um modelo prescritivo, nos dá uma referência
 - Entretanto, sempre existe a necessidade de adaptá-lo a cada projeto
 - Todo projeto de software é único (não se esqueça disso)
 - Muda:
 - As pessoas que irão executar o trabalho, o ambiente, e ou o trabalho que será conduzido

Modelos de processo

- Existem diversos modelos de processo como:
 - Modelo cascata (*waterfall*)
 - Modelo evolutivo (prototipação)
 - Desenvolvimento incremental
 - Desenvolvimento iterativo e incremental
 - Modelo espiral
 - Desenvolvimento orientado a reuso
 - RUP (Rational Unified Process)
 - CMMI
 - Metodologias ágeis
 - Scrum
 - Kanban
 - XP



**UNIVERSIDADE
FEDERAL RURAL
DE PERNAMBUCO**

**UNIDADE
ACADÊMICA DE
SERRA TALHADA**

Modelo cascata

Modelo cascata

- Foi o primeiro modelo de processo de desenvolvimento de software a ser publicado
 - Foi derivado de métodos mais gerais da engenharia

Modelo cascata

- Há casos em que os requisitos de um problema são **bem compreendidos**
- Quando o trabalho **flui da comunicação ao emprego** de forma relativamente linear
 - Essa situação ocorre algumas vezes quando adaptações ou aperfeiçoamentos bem definidos precisam ser feitos em um sistema existente

Modelo cascata

- “Adaptações ou aperfeiçoamentos”
 - Uma adaptação em software contábil exigida devido a mudanças das normas governamentais
- Pode ocorrer também em um número limitado de novos esforços de desenvolvimento
 - Só funciona bem quando os requisitos estão bem definidos e são razoavelmente estáveis
 - Ou seja, não lida bem com mudanças de plano e requisitos

Modelo cascata

- É conhecido como 'modelo cascata' devido ao encadeamento entre uma fase e outra
 - É um modelo de processo dirigido a planos
 - Você “deve” planejar e programar todas as atividades do processo antes de começar a trabalhar nelas

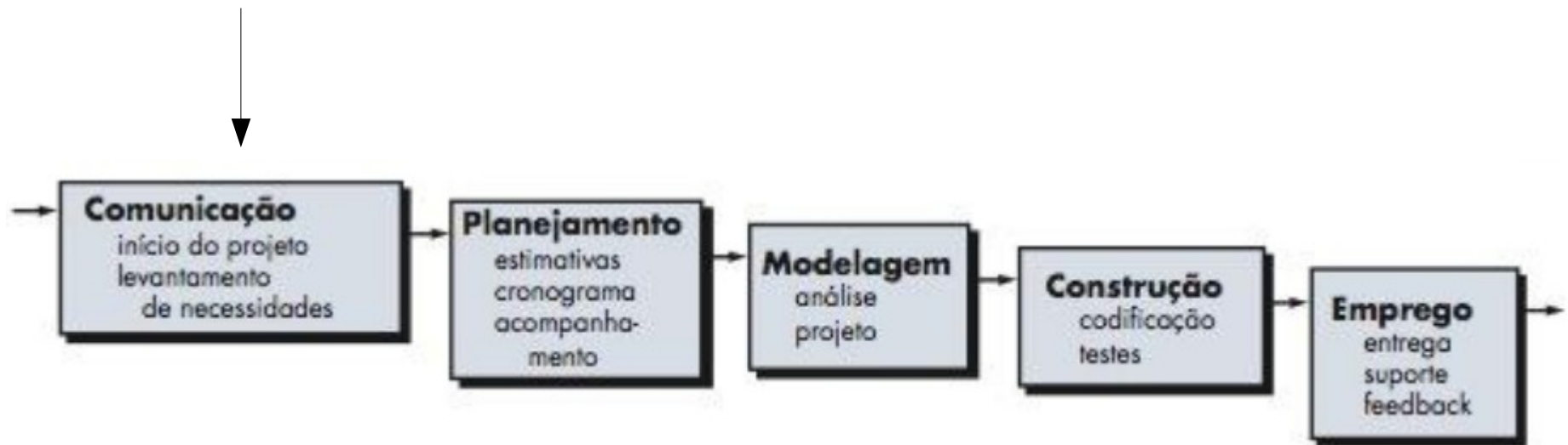
Modelo cascata

- Também é conhecido como ciclo de vida **clássico**
- Sugere uma abordagem **sequencial e sistemática** para o desenvolvimento de software

Modelo cascata

- Começa com o levantamento de necessidades por parte do cliente
- Avançando pelas fases de:
 - Planejamento
 - Modelagem
 - Construção
 - Emprego
- Culminando no suporte contínuo do software concluído

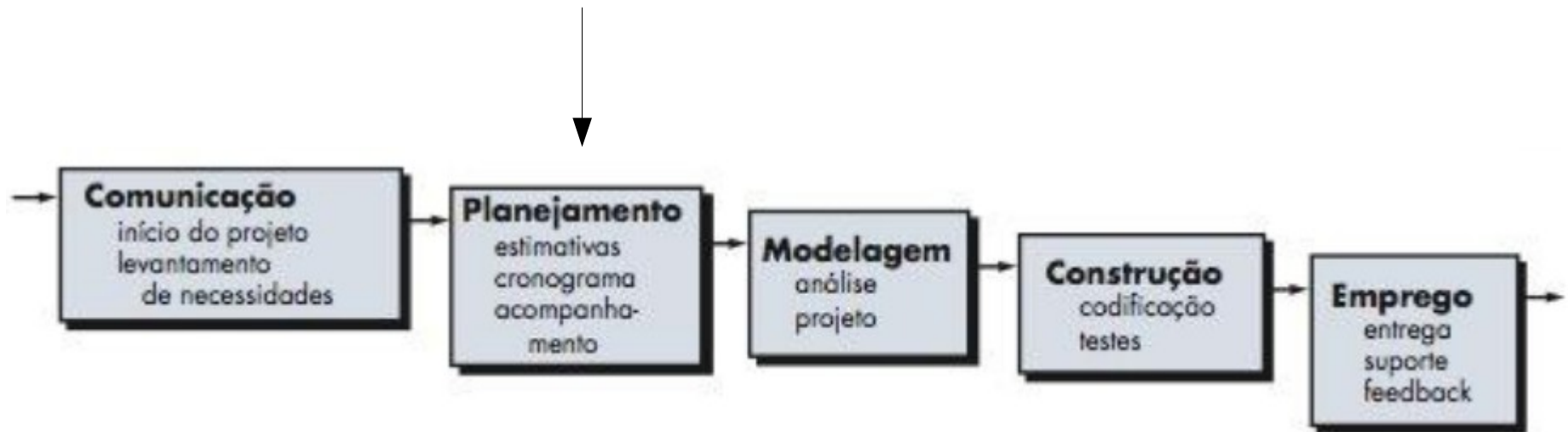
Modelo cascata



Modelo cascata

- Fase “comunicação”
 - Os serviços, restrições e metas do sistema são estabelecidos por meio de consulta aos usuários e demais partes interessadas
 - Em seguida, são “definidos em detalhes” e funcionam como uma especificação do sistema
 - Elicitação de requisitos

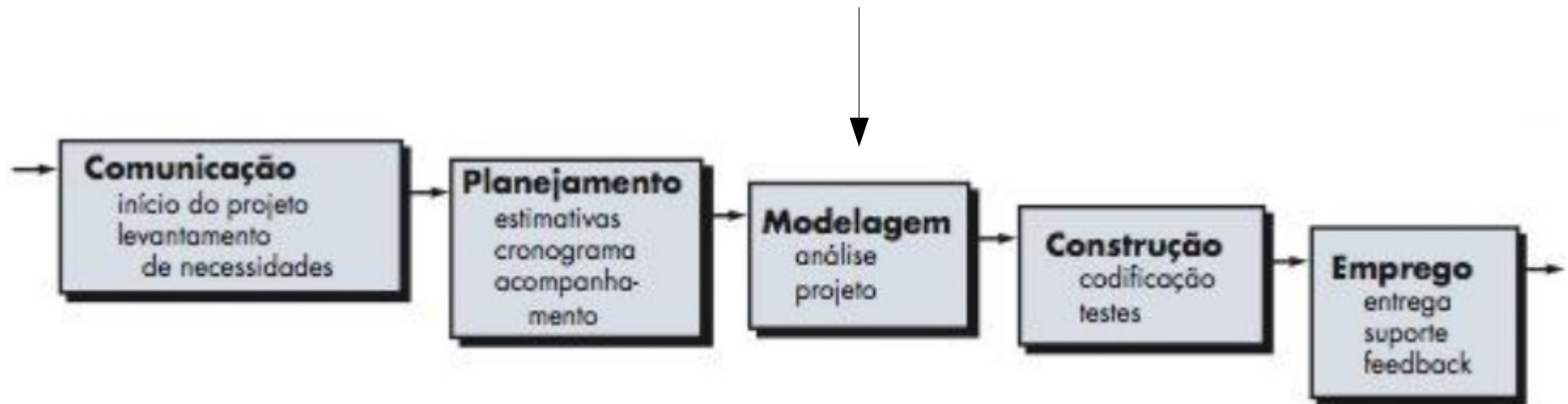
Modelo cascata



Modelo cascata

- Fase “planejamento”
 - Estima a complexidade de todas as atividades
 - Qual o tempo/custo para desenvolver/atender a todos os requisitos levantados?
 - Elabora o cronograma completo da execução das atividades
 - Planeja como será realizado o acompanhamento do projeto
 - Definindo o que seria ideal, para nas fases subsequentes comparar com o projeto em andamento
 - Saber se está dentro do “planejado”

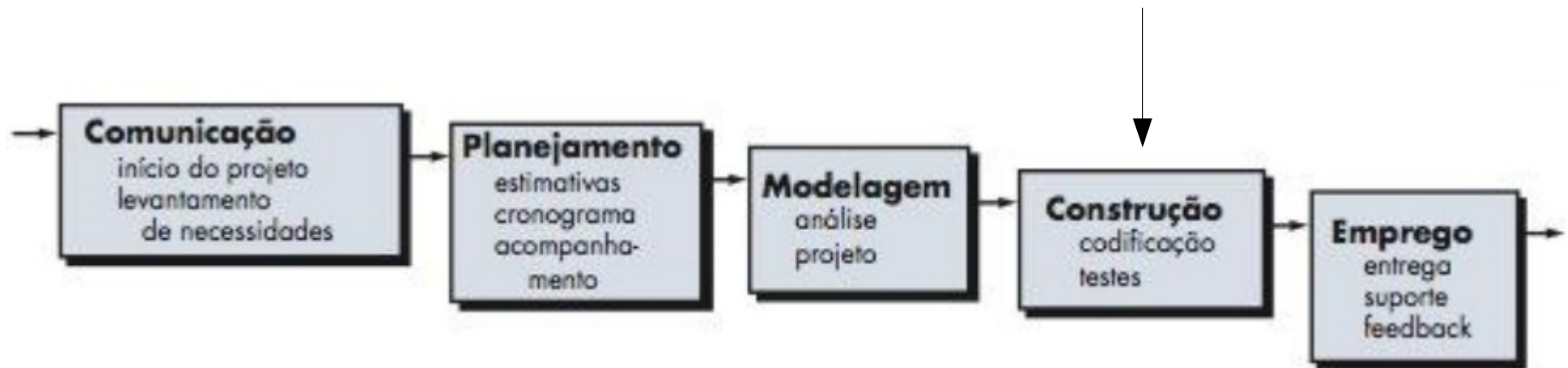
Modelo cascata



Modelo cascata

- Fase “modelagem”
 - Envolve a identificação e descrição das abstrações fundamentais do sistema de software e seus relacionamentos
 - O projeto da arquitetura do sistema é criado
 - Modelos e diagramações são desenvolvidas para auxiliar a fase de construção do projeto

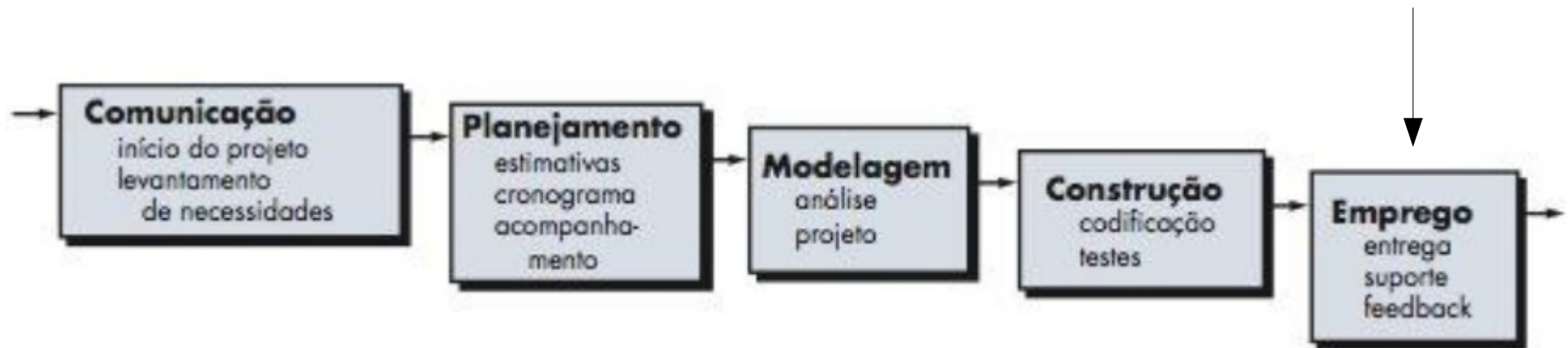
Modelo cascata



Modelo cascata

- Fase “construção”
 - O projeto de software é finalmente desenvolvido
 - O desenvolvimento do software é em conjunto com os testes unitários
 - Para cada nova funcionalidade (requisito), um ou mais testes unitários são desenvolvidos
 - O teste unitário envolve a verificação de que cada unidade atenda a sua especificação

Modelo cascata



Modelo cascata

- Fase “emprego”
 - As unidades individuais do programa ou programas são integradas e testadas como um sistema completo
 - Após feito isso, o sistema é entregue ao cliente
 - O sistema é instalado e colocado em uso (produção)
 - Essa fase também envolve a manutenção com correções de erros que não foram descobertos nas atividades de testes
 - Também pode haver ampliação de serviços com a descoberta de novos requisitos
 - Normalmente necessita de um novo contrato de prestação de serviço

Modelo cascata

- O resultado de cada fase é a aprovação de um ou mais documentos (ou artefatos)
- A fase seguinte não deve ser iniciada até que a fase anterior seja concluída
 - Na prática isso é muito difícil de acontecer
 - Essas fases acabam se sobrepondo e alimentam umas as outras

Modelo cascata

- Durante o projeto, vários problemas com requisitos podem ser identificados
- Durante a codificação, problemas de projeto são encontrados e assim por diante...

Modelo cascata

- O processo de software não é um modelo linear simples
 - Por isso o modelo cascata tradicional tem alguns problemas
 - Na prática, organizações que usam o modelo cascata, implementam *feedbacks* de uma fase para a outra
 - Assim, documentos produzidos em cada fase podem ser modificados para refletirem as alterações feitas em cada um deles

Modelo cascata

- Por causa dos **custos** de produção e aprovação de documentos, as iterações (*feedbacks*) entre as fases podem envolver significativo retrabalho
 - **Cascata não lida bem com mudanças!**

Modelo cascata

- É normal **congelarem partes** do desenvolvimento, como a especificação detalhada, e dar-se continuidade as fases posteriores
 - Por que fazer hoje o que pode ser feito amanhã?
 - Deixa o problema para depois....
- Isso pode significar que o sistema não fará o que o usuário final quer
 - Também pode levar a sistemas mal estruturados

Modelo cascata

- É um modelo consistente com outros modelos de processos
 - Documentação é produzida em cada fase do ciclo de vida do projeto
 - Essa característica torna o projeto “visível”
 - Ajuda os gerentes no monitoramento do progresso de acordo com o planejado

Modelo cascata

- Seu maior problema é a **divisão inflexível** do projeto em **fases distintas**
- Os compromissos devem ser assumidos em um estágio **MUITO** inicial do processo
 - **O que dificulta o processo inevitável de mudança de requisitos dos clientes**

Modelo cascata

- Deve ser usado apenas quando os requisitos são bem compreendidos
- Deve ser usado apenas quando os requisitos pouco provavelmente venham a ser radicalmente alterados

Modelo cascata

- No entanto, reflete o tipo de processo usado em outros projetos de engenharia
 - O que torna mais fácil de entender e gerenciar
- Como é mais fácil usar um modelo comum, processos baseado no modelo cascata ainda são comumente utilizados

Referências

- Pressman, Roger S.; Engenharia de Software: Uma Abordagem Profissional. 7ª edição, McGraw Hill Brasil, 2011.
- Sommerville, Ian.; Engenharia de Software. 9ª edição, Pearson Brasil, 2011.