



Engenharia de Software II

Aula 02 – Modelos de Processos de Software



Modelos de Processo

- ▶ Pelo fato de software, como todo capital, ser conhecido incorporado, e pelo fato de esse conhecimento ser, inicialmente, disperso, latente e em considerável medida, incompleto, o desenvolvimento de software é um processo de aprendizado social.
- ▶ Mas o que é exatamente um processo de software de ponto de vista técnico? Processo de software é definido como uma metodologia para as atividades, ações e tarefas necessárias para desenvolver um software de alta qualidade.
- ▶ Mais importante, a engenharia de software é realizada por pessoas criativas e com amplos conhecimentos e que devem adaptar um processo de software maduro, de forma que fique apropriado aos produtos desenvolvidos e às demandas do seu mercado.

Panorama

PANORAMA

O que é? Quando se trabalha na elaboração de um produto ou sistema, é importante seguir uma série de passos previsíveis — um roteiro que ajude a criar um resultado de alta qualidade e dentro do prazo estabelecido. O roteiro é denominado “processo de software”.

Quem realiza? Os engenheiros de software e seus gerentes adaptam o processo às suas necessidades e então o seguem. Os solicitantes do software têm um papel a desempenhar no processo de definição, construção e teste do software.

Por que ele é importante? Porque propicia estabilidade, controle e organização para uma atividade que pode, sem controle, tornar-se bastante caótica. Entretanto, uma abordagem de engenharia de software moderna deve ser “ágil”. Deve demandar apenas atividades, controles e produtos de trabalho que sejam apropriados para a equipe do projeto e para o produto a ser produzido.

Quais são as etapas envolvidas? O processo adotado depende do software a ser desenvolvido. Um determinado processo pode ser apropriado para um software do sistema “aviônico” de uma aeronave, enquanto um processo totalmente diferente pode ser indicado para a criação de um site.

Qual é o artefato? Do ponto de vista de um engenheiro de software, os produtos de trabalho são os programas, os documentos e os dados produzidos em consequência das atividades e tarefas definidas pelo processo.

Como garantir que o trabalho foi feito corretamente? Há muitos mecanismos de avaliação dos processos de software que possibilitam às organizações determinarem o nível de “maturidade” de seu processo de software. Entretanto, a qualidade, o cumprimento de prazos e a viabilidade a longo prazo do produto que se desenvolve são os melhores indicadores da eficácia do processo utilizado.



Modelo de Processo Genérico

- Processo foi definido como um conjunto de atividades de trabalho, ações e tarefas realizadas quando algum artefato de software deve ser criado.
- Cada ação é definida por um conjunto de tarefas, o qual identifica as tarefas de trabalho a se completadas, os artefatos de software que serão produzidos, os fatores de garantia da qualidade que serão exigidos e os marcos utilizados para indicar o progresso.
- Uma metodologia de processo genérica para engenharia de software estabelece cinco atividades metodológicas: **comunicação, planejamento, modelagem, construção e entrega**. Além disso, um conjunto de atividades de apoio são aplicadas ao longo do processo, como o acompanhamento e controle do projeto, a administração de riscos, a garantia de qualidade, o gerenciamento das configurações, as revisões técnicas e outras.

Processo de Software

Processo de software

Metodologia do processo

Atividades de apoio

atividade metodológica nº 1

ação de engenharia de software nº 1.1

Conjuntos
de tarefas

⋮

ação de engenharia de software nº 1.k

Conjuntos
de tarefas

tarefas de trabalho
artefatos de software
fatores de garantia da qualidade
pontos de controle do projeto

tarefas de trabalho
artefatos de software
fatores de garantia da qualidade
pontos de controle do projeto

⋮

atividade metodológica nº n

ação de engenharia de software nº n.1

Conjuntos
de tarefas

⋮

ação de engenharia de software nº n.m

Conjuntos
de tarefas

tarefas de trabalho
artefatos de software
fatores de garantia da qualidade
pontos de controle do projeto

tarefas de trabalho
artefatos de software
fatores de garantia da qualidade
pontos de controle do projeto



Fluxo de Processo

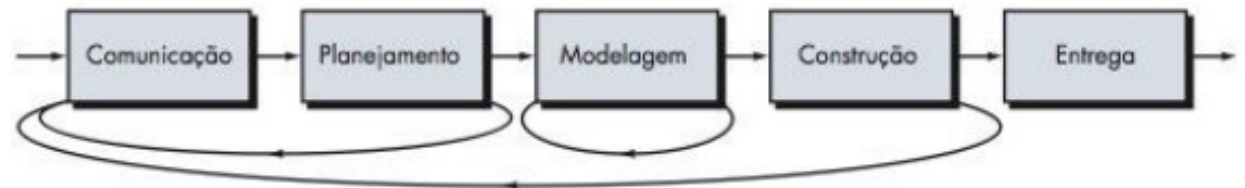


- Deve-se notar que um importante aspecto do processo de software ainda não foi discutido. O fluxo de processo descreve como são organizadas as atividades metodológicas, bem como as ações e tarefas que ocorrem dentro de cada atividade em relação à sequência e ao tempo.
- Um fluxo de **processo linear** executa cada uma das cinco atividades metodológicas em sequência. Um fluxo de **processo iterativo** repete uma ou mais das atividades antes de prosseguir para a seguinte.
- Um fluxo de **processo evolucionário** executa as atividades de uma forma “circular”. Um fluxo de **processo paralelo** executa uma ou mais atividades em paralelo com outras atividades.

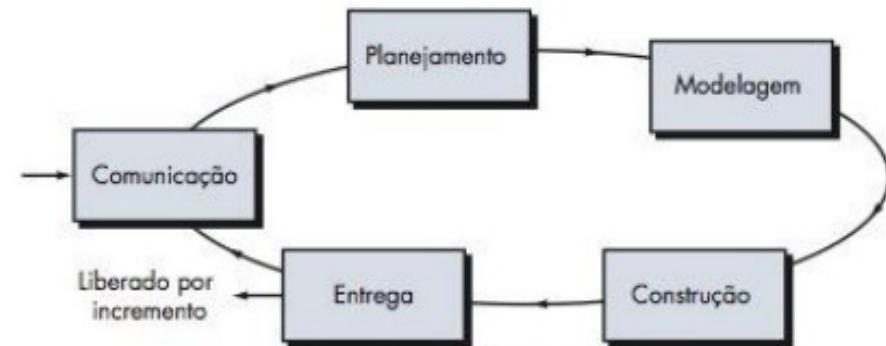
Fluxo de Processo



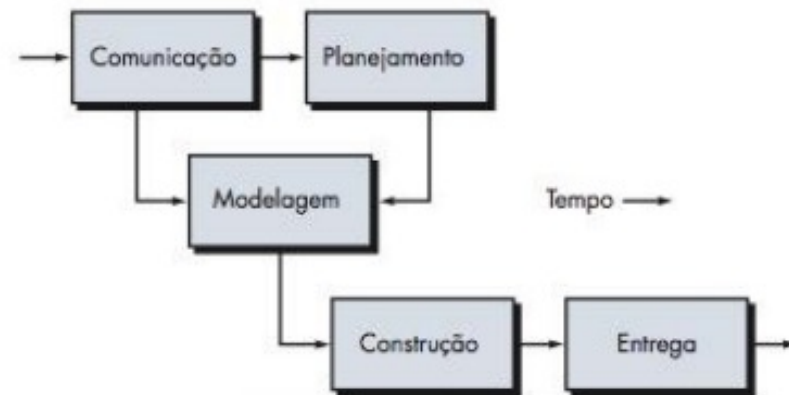
(a) Fluxo de processo linear



(b) Fluxo de processo iterativo



(c) Fluxo de processo evolucionário



(d) Fluxo de processo paralelo

Conjunto de Tarefas

INFORMAÇÕES



Conjunto de tarefas

Um conjunto de tarefas define o verdadeiro trabalho a ser feito para se atingir os objetivos de uma ação de engenharia de software. Por exemplo, elucidação (mais comumente denominada "levantamento de requisitos") é uma importante ação de engenharia de software que ocorre durante a atividade de comunicação. A meta do levantamento de requisitos é compreender o que os vários interessados esperam do software a ser desenvolvido.

Para um projeto pequeno, relativamente simples, o conjunto de tarefas para levantamento das necessidades seria semelhante ao seguinte:

1. Fazer uma lista dos envolvidos no projeto.
2. Fazer uma reunião informal com todos os interessados.
3. Solicitar para cada interessado uma lista com as características e funções necessárias.
4. Discutir sobre os requisitos e construir uma lista final.
5. Organizar os requisitos por grau de prioridade.
6. Destacar pontos de incertezas.

Para um projeto de software maior e mais complexo, necessita-se de um conjunto diferente de tarefas. Tal conjunto pode incluir as seguintes tarefas de trabalho:

1. Fazer uma lista dos envolvidos no projeto.
2. Entrevistar separadamente cada um dos envolvidos para levantamento geral de suas expectativas e necessidades.

3. Fazer uma lista preliminar das funções e características, com base nas informações fornecidas pelos interessados.
4. Agendar uma série de reuniões facilitadoras para especificação de aplicações.
5. Promover reuniões.
6. Incluir cenários informais de usuários como parte de cada reunião.
7. Aprimorar os cenários de usuários, com base no *feedback* dos interessados.
8. Fazer uma lista revisada das necessidades dos interessados.
9. Empregar técnicas de aplicação de funções de qualidade para estabelecer graus de prioridade dos requisitos.
10. Agrupar os requisitos de modo que eles possam ser entregues incrementalmente.
11. Fazer um levantamento das limitações e restrições que serão aplicadas ao sistema.
12. Discutir sobre os métodos para validação do sistema.

Esses dois conjuntos de tarefas atingem o objetivo de "levantamento de necessidades", porém, são bem diferentes em relação aos graus de profundidade e formalidade. A equipe de software deve escolher o conjunto de tarefas que lhe possibilitará atingir o objetivo de cada ação, mantendo, inclusive, a qualidade e a agilidade.

Padrão de Processo

INFORMAÇÕES



Um exemplo de padrão de processo

O padrão de processo sintetizado a seguir descreve uma abordagem que pode ser aplicada quando os interessados têm uma ideia geral do que precisa ser feito, mas estão incertos quanto aos requisitos específicos do software.

Nome do padrão. Requisitos Imprecisos

Intento. Este padrão descreve uma abordagem voltada para a construção de um modelo (um protótipo) passível de ser avaliado iterativamente pelos interessados, num esforço para identificar ou solidificar os requisitos de software.

Tipo. Padrão de fase.

Contexto inicial. As condições seguintes devem ser atendidas antes de iniciar esse padrão: (1) interessados identificados; (2) forma de comunicação entre interessados e equipe de software já determinada; (3) principal problema de software a ser resolvido já identificado pelos interessados; (4) compreensão inicial do escopo do projeto, dos requisitos de negócio básicos e das restrições do projeto já atingida.

Problema. Os requisitos são vagos ou inexistentes, ainda assim há um reconhecimento claro de que existe um problema a

ser solucionado e este deve ser identificado utilizando-se uma solução de software. Os interessados não sabem o que querem, ou seja, eles não conseguem descrever os requisitos de software em detalhe.

Solução. Uma descrição do processo de prototipação poderia ser apresentada nesta etapa, mas é descrita posteriormente na Seção 2.3.3.

Contexto resultante. Um protótipo de software que identifique os requisitos básicos (por exemplo, modos de interação, características computacionais, funções de processamento) é aprovado pelos interessados. Em seguida, (1) o protótipo pode evoluir por uma série de incrementos para se tornar o software de produção ou (2) o protótipo pode ser descartado e o software de produção ser construído usando-se algum outro padrão de processos.

Padrões associados. Os seguintes padrões estão relacionados a esse padrão: **Comunicação Com O Cliente**, **Projeto Iterativo**, **Desenvolvimento Iterativo**, **Avaliação Do Cliente**, **Extração De Requisitos**.

Usos conhecidos e um exemplo. A prototipação é recomendada quando as necessidades são incertas.

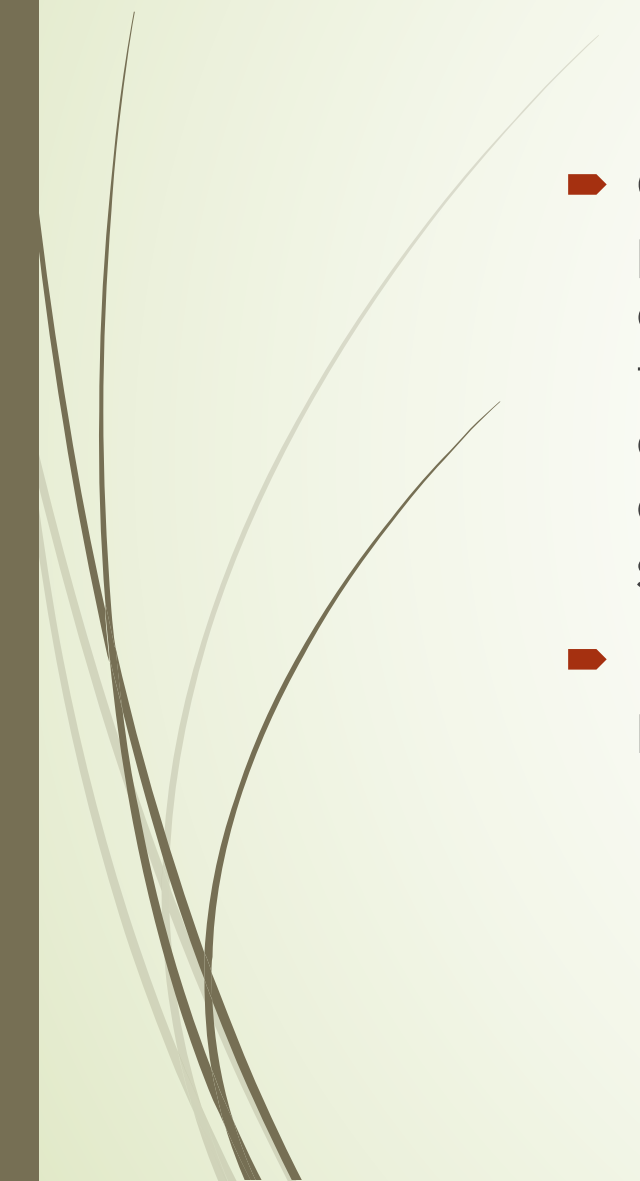


Avaliação e Aperfeiçoamento de Processos

- A existência de um processo de software não garante que o software será entregue dentro do prazo, que estará de acordo com as necessidades do cliente ou que apresentará características técnicas que conduzirão a qualidade de longo prazo.
- Os padrões de processo devem ser combinados com uma prática de engenharia de software consistente. Além disso, o próprio processo pode ser avaliado para assegurar que está de acordo com um conjunto de critérios de processos básicos comprovados como essenciais para uma engenharia de software de sucesso.




Modelos de Processo Prescritivo

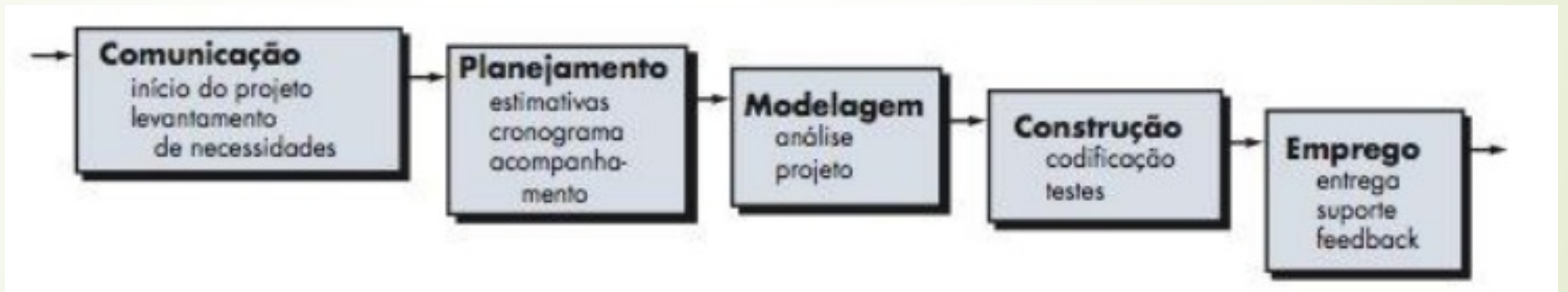
- Originalmente, modelos de processo prescritivo foram propostos para trazer ordem ao caos existente na área de desenvolvimento de software. A história tem demonstrado que esses modelos tradicionais proporcionaram uma considerável contribuição quanto à estrutura utilizável no trabalho de engenharia de software e forneceram um roteiro razoavelmente eficaz para as equipes de software.
 - Entretanto, o trabalho de engenharia de software e o seu produto permanecem “à beira do caos”.
- 



Modelo Cascata

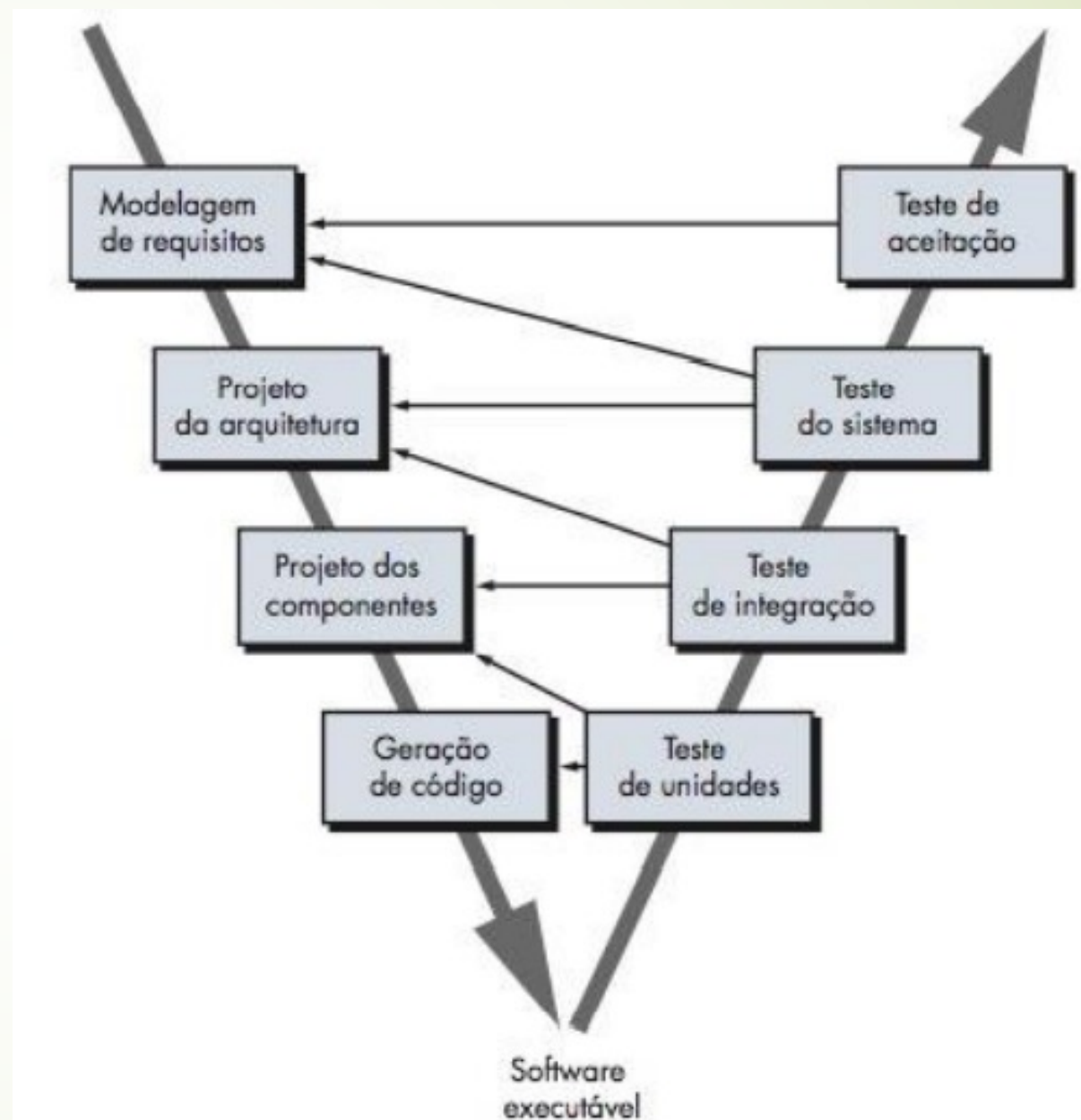
- Há casos em que os requisitos de um problema são bem compreendidos – quando o tratamento flui da **comunicação** ao **emprego** de forma relativamente linear.
 - O modelo cascata, algumas vezes chamado de ciclo de vida clássico, sugere uma abordagem sequencial e sistemática para o desenvolvimento de software, começando com o levantamento de necessidades por parte do cliente, avançando pelas fases de planejamento, modelagem, construção, emprego e culminando no suporte contínuo de software concluído.
- 

Representação Modelo Cascata




Modelo V

- O modelo casca é o paradigma mais antigo da engenharia de software. Entretanto, ao longo das últimas três décadas, as críticas a este modelo de processo fez com que até mesmo seus mais ardentes defensores questionassem sua eficácia.

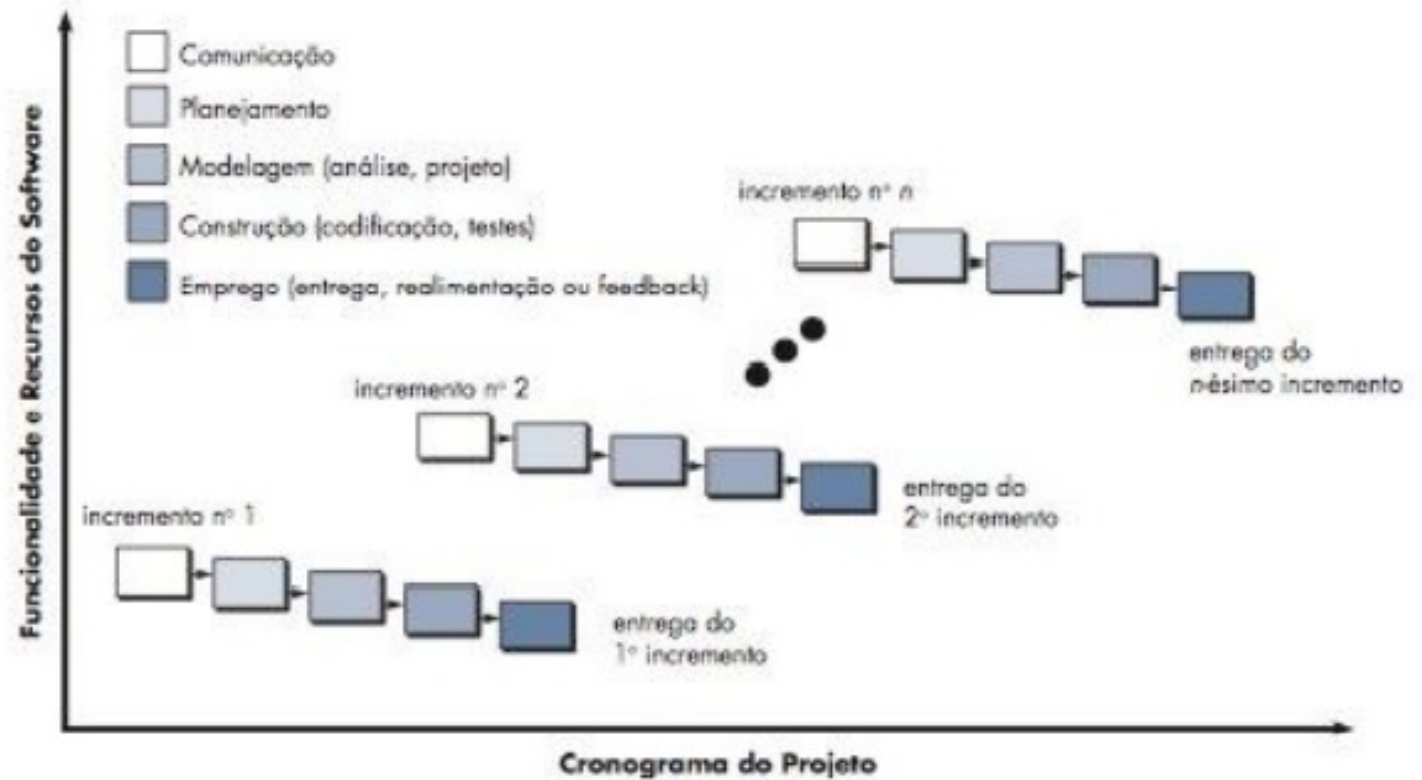




Modelos de Processo Incremental

- Em várias situações, os requisitos iniciais do software são razoavelmente bem definidos, entretanto, devido ao escopo geral do trabalho de desenvolvimento, o uso de um processo puramente linear não é utilizado.
 - O modelo incremental combina elementos dos fluxos de processo lineares e paralelos. O modelo incremental aplica sequências lineares, de forma escalonada, à medida que o tempo vai avançando. Cada sequência linear gera “incrementais” (entregáveis/aprovados/liberados do software).
- 

Modelo Incremental

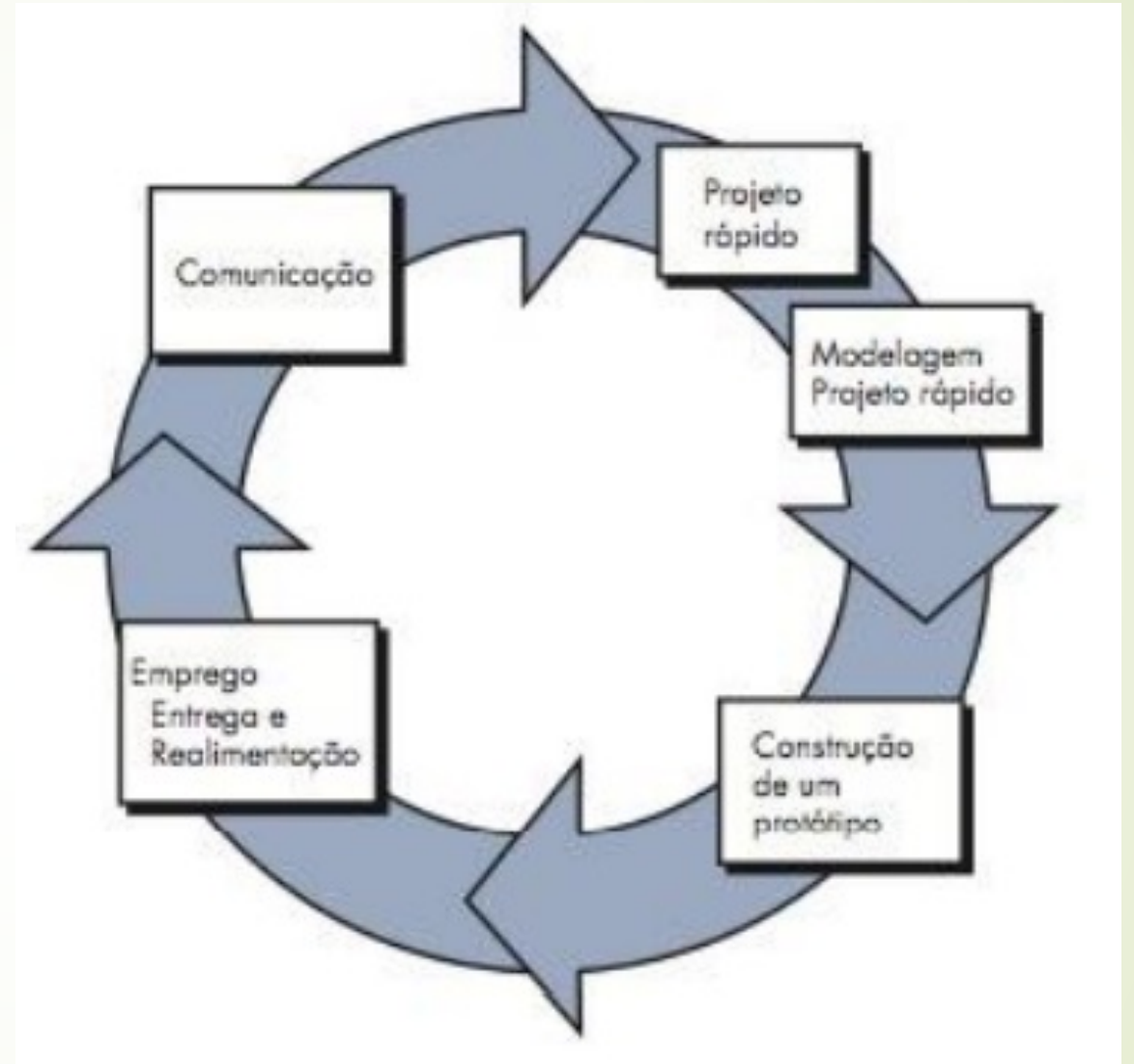




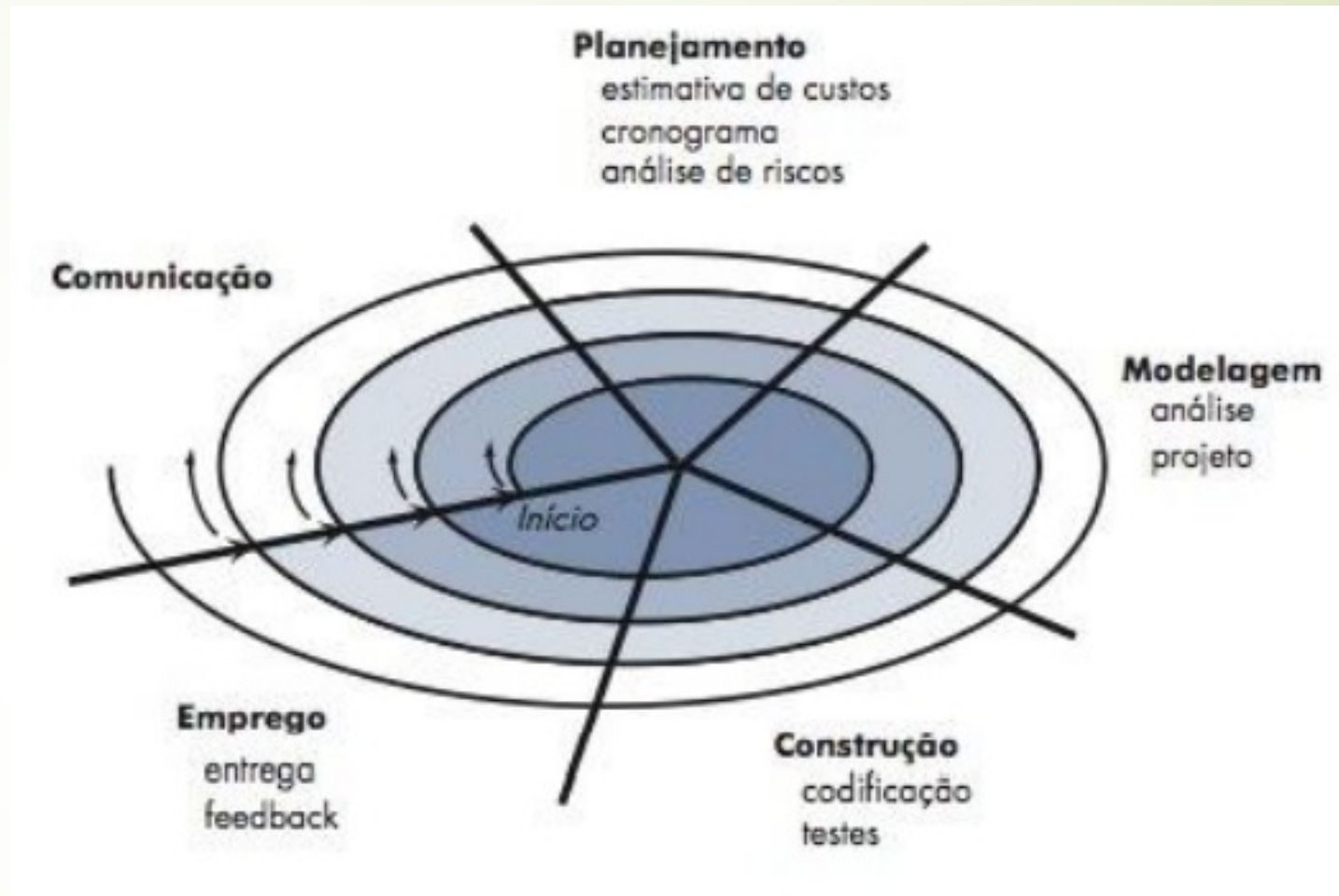
Modelos de Processo Evolucionário

- Software, assim como todos sistemas complexos, evolui ao longo do tempo. Conforme o desenvolvimento do projeto avança, as necessidades de negócio e de produto mudam frequentemente, tornando inadequado seguir um planejamento em linha reta de um produto final.
- Modelos evolucionários são iterativos. Apresentam características que possibilitam desenvolver versões cada vez mais completas do software. Nos parágrafos seguintes, são apresentados dois modelos comuns em processo evolucionários.
- Prototipação e Modelo Espiral.

Paradigma da Prototipação



Modelo Espiral

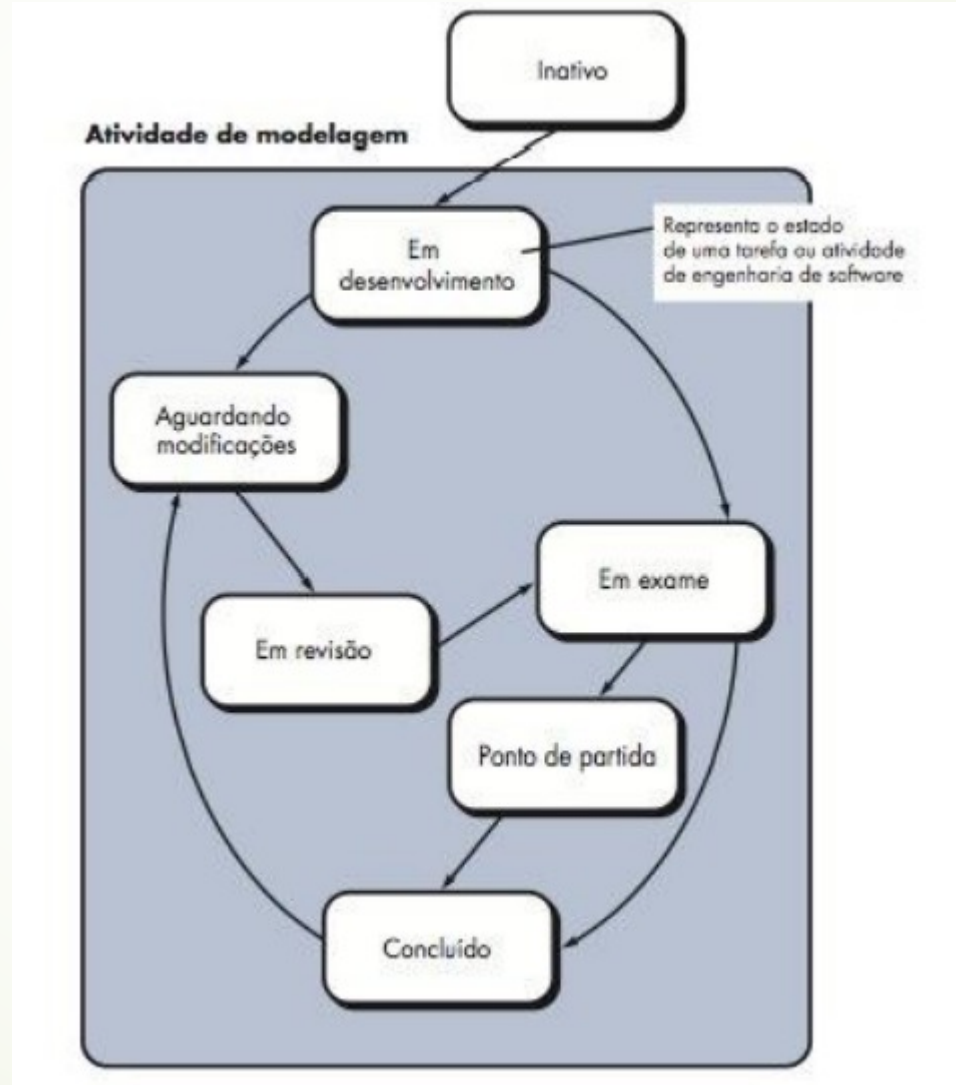




Modelos Concorrentes


- O modelo de desenvolvimento concorrente, algumas vezes denominado engenharia concorrente, possibilita à equipe de software representar elementos concorrentes e iterativos de qualquer um dos modelos de processos descritos.
- No início do projeto, a atividade de comunicação, completou sua primeira iteração e se encontra no estado de **aguardando modificações**.
- A atividade de modelagem que se encontra no estado **inativo**, agora faz uma transição para o estado **em desenvolvimento** para o estado **aguardando modificações**.

Atividade de Modelagem



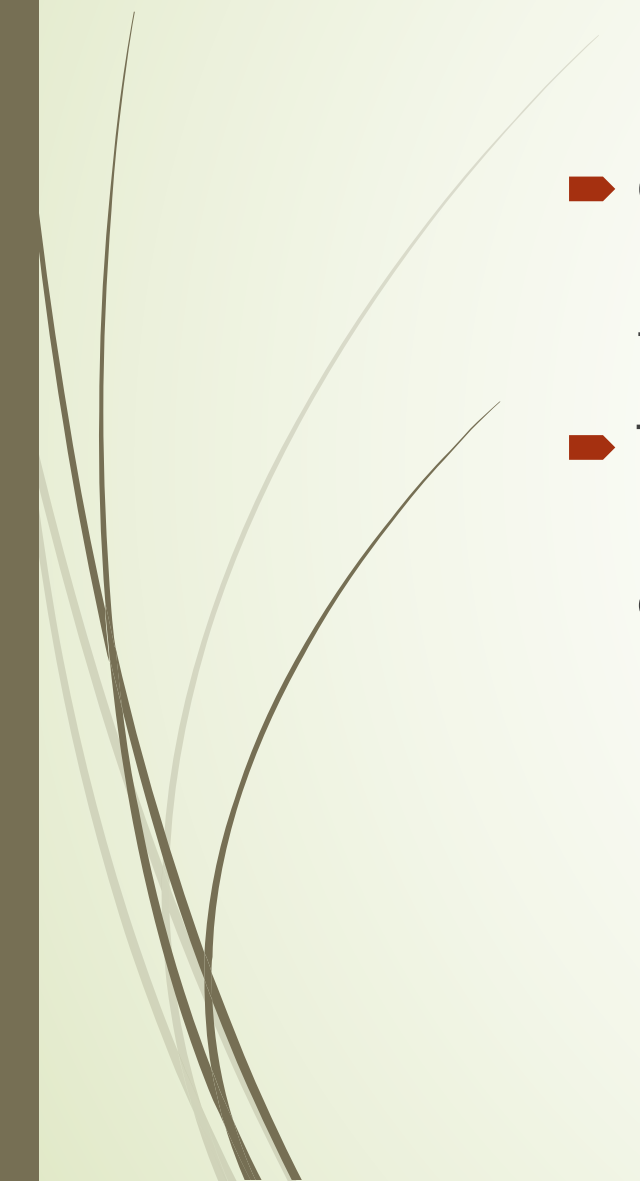


Finalizando Processo Evolucionários

- O objetivo dos modelos evolucionários é desenvolver software de alta qualidade de modo iterativo ou incremental. Entretanto, é possível usar um processo evolucionário para enfatizar a flexibilidade, a extensibilidade e a velocidade do desenvolvimento.
 - O desafio pra as equipes de software e seus gerentes será estabelecer um equilíbrio apropriado entre esses parâmetros críticos de projeto e produto e a satisfação do cliente.
- 



Processo Especializado

- Os modelos de processo especializado levam em conta muita das características de um ou mais dos modelos tradicionais apresentados.
 - Tais modelos tendem a ser aplicados quando se opta por uma abordagem de engenharia de software especializada ou definida de forma restrita.
- 




Desenvolvimento Baseado em Componentes

1. Produtos baseados em componentes disponíveis são pesquisados e avaliados para o campo de aplicação em questão.
2. Itens de integração de componentes são considerados.
3. Uma arquitetura de software é projetada para acomodar os componentes.
4. Os componentes são integrados na arquitetura.
5. Testes completos são realizados para assegurar funcionalidade adequada.



Modelo de Métodos Formais

- O modelo de métodos formais engloba um conjunto de atividades que conduzem à especificação matemática formal do software.
 - Os métodos formais possibilitam especificar, desenvolver e verificar um sistema baseado em computador através da aplicação de uma notação matemática rigorosa.
 - Uma variação dessa abordagem, chamada de engenharia de software “cleanroom”, é aplicada por algumas organizações de desenvolvimento do software.
- 

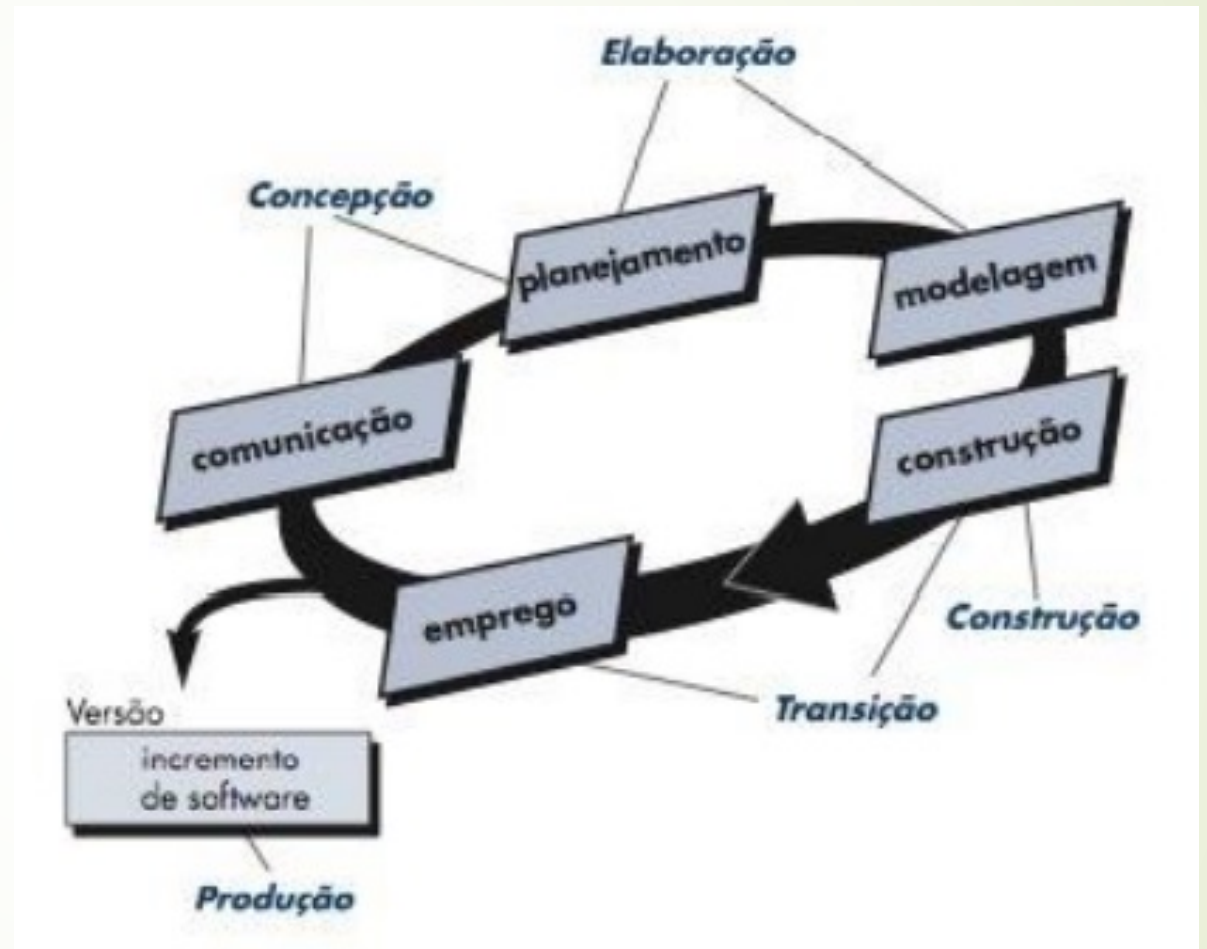


Processo Unificado

- No livro que deu origem ao Processo unificado, Ivan Jacobson, Grady Booch e James Rumbaugh discutem a necessidade de um processo de software “dirigido a casos de uso, centrado na arquitetura, iterativo e incremental”


Hoje em dia, a tendência do software é no sentido de sistemas maiores e mais complexos. Isso se deve, em parte, ao fato de que os computadores tornam-se mais potentes a cada ano, levando os usuários a ter uma expectativa maior em relação a eles. Essa tendência também foi influenciada pelo uso crescente da Internet para troca de todos os tipos de informação... Nosso apetite por software cada vez mais sofisticado aumenta à medida que tomamos conhecimento de uma versão do produto para a seguinte, como o produto pode ser aperfeiçoado. Queremos software que seja mais e mais adaptado a nossas necessidades, mas isso, por sua vez, simplesmente torna o software mais complexo. Em suma, queremos cada vez mais.

Fases Processo Unificado



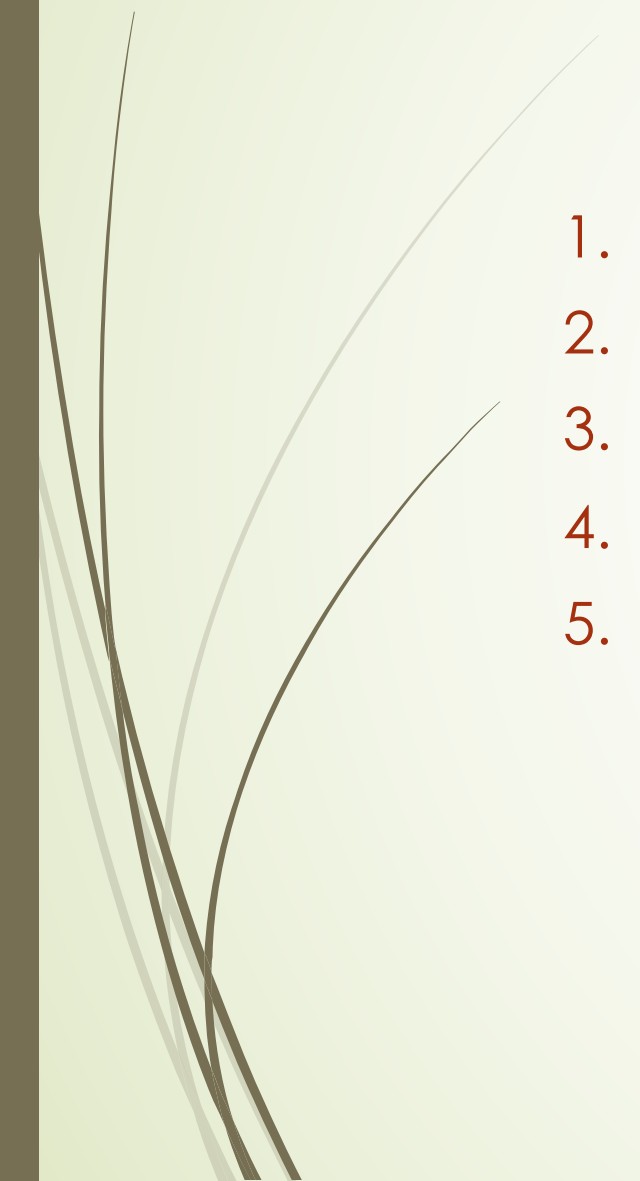


Processo Pessoal e de Equipe

- O melhor processo de software é aquele próximo às pessoas que realizarão o trabalho. Se um modelo de processo de software for desenvolvido a nível corporativo ou organizacional, ele apenas será efetivo se for aberto a significativas adaptações a fim de atender às necessidades da equipe do projeto.
- 



Processo de Software Pessoal (PSP)

- 
1. Planejamento.
 2. Projeto de alto nível.
 3. Revisão de projeto de alto nível.
 4. Desenvolvimento.
 5. Autópsia.



Processo de Software em Equipe (TSP)

- Criar equipes autogeridas que planejem e acompanhem seu próprio trabalho, estabeleçam metas e sejam proprietárias de seus processos e planos.
- Mostrar aos gerentes como treinar e motivar suas equipes e como ajudá-las a manter alto desempenho.
- Acelerar o aperfeiçoamento dos processos de software, tornando o comportamento do Modelo de Maturidade de Capacidade.
- Fornecer orientação para melhorias a organizações com elevado grau de maturidade.
- Facilitar o ensino universitário de habilidade de trabalho em equipe de nível industrial.

Tecnologia de Processos



Ferramentas de modelagem de processos

Objetivo: quando uma organização trabalha para aprimorar um processo de negócio (ou de software), ela precisa, primeiramente, compreendê-lo. As ferramentas de modelagem de processos (também chamadas ferramentas de tecnologia de processos ou ferramentas de gerenciamento de processos) são usadas para representar elementos-chave de um processo para que possa ser mais bem compreendido. Essas ferramentas podem também oferecer "links" para descrições de processos, ajudando os envolvidos no processo a compreender as ações e tarefas necessárias para realizá-lo. As ferramentas de modelagem de processos fornecem links para outras ferramentas que oferecem suporte para atividades de processos definidas.

Mecânica: as ferramentas nesta categoria permitem a uma equipe de desenvolvimento definir os elementos de um modelo

FERRAMENTAS DO SOFTWARE

único de processo (ações, tarefas, artefato, pontos de garantia da qualidade de software), dar orientação detalhada sobre o conteúdo ou descrição de cada elemento de um processo e, então, gerenciar o processo conforme ele for conduzido. Em alguns casos, as ferramentas de tecnologia de processos incorporam tarefas padronizadas de gerenciamento de projeto como estimativa de custos, cronograma, acompanhamento e controle.

Ferramentas Representativas:

Igrafx Process Tools — ferramentas que capacitam uma equipe a mapear, medir e modelar o processo de software (www.micrografx.com)

Adeptia BPM Server — projetada para gerenciar, automatizar e otimizar processos de negócio (www.adeptia.com)

SpeedDev Suite — conjunto de seis ferramentas com forte ênfase no gerenciamento das atividades de comunicação e modelagem (www.speeddev.com)



Processo do Produto

- Se o processo for fraco, certamente produto final sofrerá consequências. Porém, uma confiança excessiva e obsessiva no processo é igualmente perigosa.
- As pessoas obtêm satisfação tanto do processo criativo quanto do produto final. Um artista sente prazer tanto de suas pinceladas quanto do resultado geral do seu quadro.
- Como profissional de software criativo, você também deve extrair tanta satisfação do processo como do produto final.
- A dualidade produto e processo é um elemento importante para manter pessoas criativas engajadas à medida que a engenharia de software continua a evoluir.