UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

UNIDADE ACADÊMICA DE SERRA TALHADA

Prototipação e Desenvolvimento Incremental

Ygor Amaral <ygor.amaral@ufrpe.br>

Disciplina: Processo de Desenvolvimento de Software

Curso: Sistemas de Informação (2016.1)

Lidando com mudanças

UFRPE - UAST

- A mudança é inevitável em todos os grandes projetos de software

- Os requisitos do sistema mudam...
  - O negócio que adquiriu o sistema responde a pressões externas!
  - Prioridades são alteradas ao longo do tempo

Lidando com mudanças

UFRPE - UAST

- Além disso, com a disponibilidade de novas tecnologias:
  - Emergem novos projetos e possibilidades de implementação
- É essencial que os projetos possam acomodar mudanças no software em desenvolvimento:
  - Qualquer que seja o modelo do processo de desenvolvimento...

Lidando com mudanças

UFRPE - UAST

- A mudança aumenta os custos de desenvolvimento de software!
  - Geralmente significa que o trabalho deve ser refeito (retrabalho)

- Ex:
  - Pode ser necessário reprojetar o sistema de acordo com os novos requisitos
  - Ou seja, mudar qualquer programa que tenha sido desenvolvido e testar novamente o sistema

Lidando com mudanças

UFRPE - UAST

- Existem duas abordagens que podem ser adotadas para a redução de custos de retrabalho:
  - Prevenção de mudanças
  - Tolerância a mudanças

- Analisaremos essas duas abordagens a seguir

Prevenção de mudanças

UFRPE - UAST

- O processo de software inclui atividades capazes de antecipar as mudanças possíveis
  - Para que não seja necessário qualquer retrabalho
- Ex:
  - Um protótipo de sistema pode ser desenvolvido para mostrar algumas características-chave do sistema
  - Eles podem experimentar o protótipo e refinar seus requisitos
    - Isso acontece antes de se comprometer com elevados custos de produção de software

Tolerância a mudanças

UFRPE - UAST

- Nesse caso, o processo foi projetado para que as mudanças possam ser acomodadas a um custo relativamente baixo

- Isso normalmente envolve alguma forma de desenvolvimento incremental

Tolerância a mudanças

UFRPE - UAST

- As alterações propostas podem ser aplicadas em incrementos que ainda não foram desenvolvidos

- Se isso for impossível...
  - Então apenas um incremento (uma pequena parte do sistema) deve ser alterado para incorporar

Prototipação

Prototipação

UFRPE - UAST

- É uma abordagem proposta para lidar com mudanças de forma mais eficaz
  - Quando comparado ao modelo *waterfall*

- Propósito:
  - Desenvolver rapidamente uma versão do sistema para verificar as necessidades do cliente e a viabilidade de algumas decisões de projeto
    - Essa versão é um “protótipo”
    - Abordagem contrária ao propósito do *waterfall*

Prototipação

UFRPE - UAST

- O que é um protótipo de software?

Prototipação

UFRPE - UAST

- O que é um protótipo de software?
  - É uma versão inicial de um sistema de software
- Qual será a utilidade de um protótipo?

Prototipação

UFRPE - UAST

- O que é um protótipo de software?
  - É uma versão inicial de um sistema de software
- Qual será a utilidade de um protótipo?
  - Demonstrar conceitos
  - Experimentar opções de projeto
  - Descobrir mais sobre o problema e suas possíveis soluções
    - Entender melhor o domínio do problema

Prototipação

UFRPE - UAST

- O desenvolvimento rápido e iterativo do protótipo é essencial
  - Dessa forma os custos são controlados mais eficientemente
  - Aplicado da forma correta faz com que os *stakeholders* possam experimentá-lo no início do processo de software
    - No *waterfall* isso acontece próximo do fim

Prototipação

UFRPE - UAST

- Um protótipo pode ser usado para ajudar a antecipar as mudanças que podem ser requisitadas:
  - Pode ajudar na elicitação e validação de requisitos de sistema
  - Pode ser usado para estudar soluções específicas do software

Prototipação

UFRPE - UAST

- Na sua forma ideal:
  - O protótipo atua como um mecanismo para identificar os requisitos do software

- Normalmente, um protótipo não é operacional em produção
  - Ou seja, o cliente não deve utilizar o protótipo em um ambiente real

Prototipação

UFRPE - UAST

- Existem dois tipos de protótipos:
  - Protótipos descartáveis
    - São apenas para demonstrar uma ideia e que não poderão ser incorporados ao sistema final
    - Recomenda-se que se jogue fora
  - Protótipos evolucionários
    - Evoluem lentamente até se transformarem no sistema real

Prototipação

UFRPE - UAST

- Protótipos do sistema permitem aos usuários ver quão bem o sistema dá suporte a seu trabalho

- Por que você acha que isso acontece?

Prototipação

UFRPE - UAST

- Porque ajuda a obter novas ideias para requisitos e encontrar pontos fortes e fracos do software
  - Ou seja, ajuda a propor novos requisitos do sistema
  - Requisitos que são mais próximos do que o usuário acredita ser o ideal

Prototipação

UFRPE - UAST

- O desenvolvimento do protótipo pode revelar erros e omissões nos requisitos propostos!
  - Isso acontece ainda em fases iniciais do projeto

- A função descrita em uma especificação pode parecer útil e bem definida inicialmente
  - Entretanto, quando essa função é combinada com outras, os usuários muitas vezes percebem que sua visão inicial foi incorreta ou incompleta

Prototipação

UFRPE - UAST

- Com a criação de protótipos no desenvolvimento do software:
  - A especificação do sistema pode então ser modificada para refletir o entendimento dos requisitos alterados
  - Para tentar obter sempre a satisfação do cliente

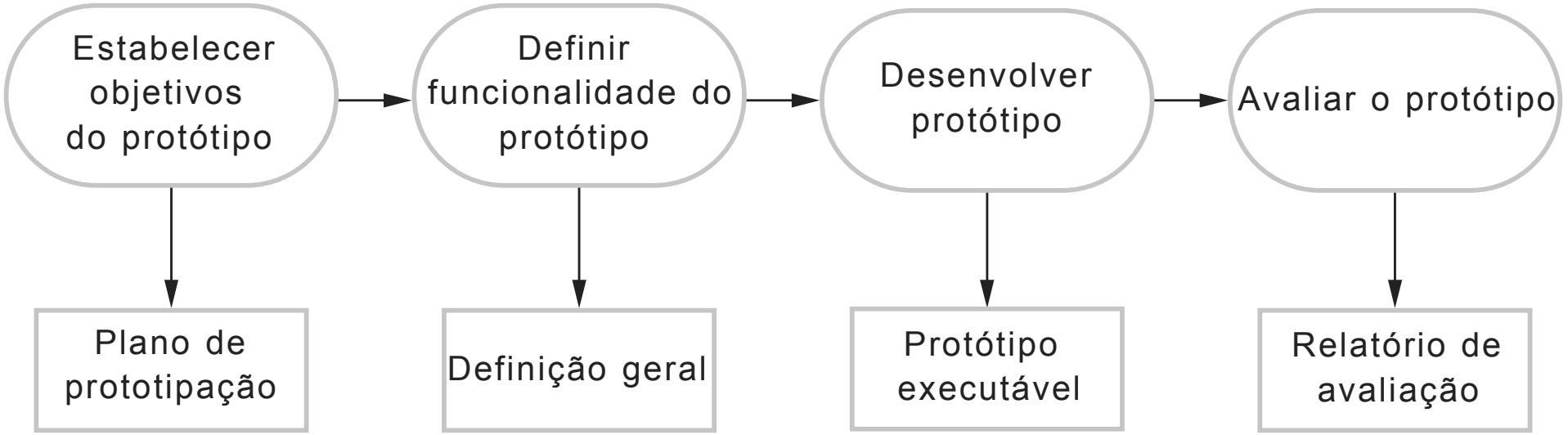
Prototipação

UFRPE - UAST

- Prototipação também é uma parte essencial do processo de projeto da interface de usuário
  - Descrições textuais e diagramas não são bons o suficiente para expressar seus requisitos
    - “Devido à natureza dinâmica de tais interfaces”
  - A prototipação com envolvimento do usuário final é uma boa maneira de desenvolver interfaces gráficas

Processo de Prototipação

UFRPE - UAST



O processo de desenvolvimento de protótipo

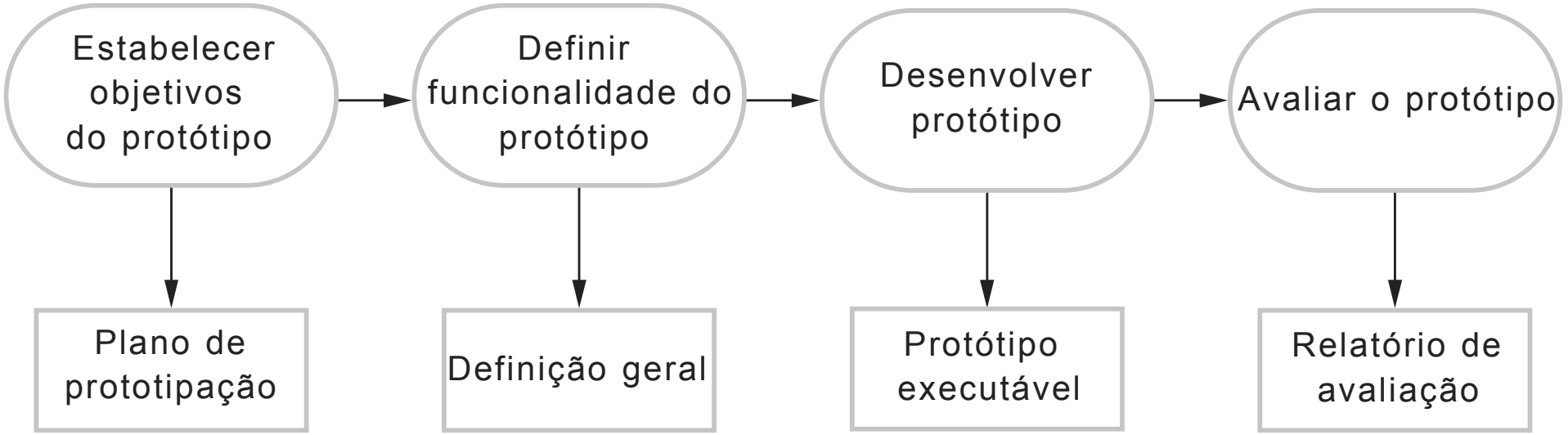
Processo de Prototipação

UFRPE - UAST

- Estabelecer os objetivos do protótipo:
  - Os objetivos da prototipação devem ser explicitados desde o início do processo
  - Estes podem ser o desenvolvimento de um sistema para:
    - Prototipar a interface de usuário
    - Validar os requisitos
    - Demonstrar aos gerentes a viabilidade da aplicação
  - O mesmo protótipo não pode cumprir todos esses objetivos
    - Se os objetivos não são declarados, os *stakeholders* podem não entender a função do protótipo

Processo de Prototipação

UFRPE - UAST



O processo de desenvolvimento de protótipo

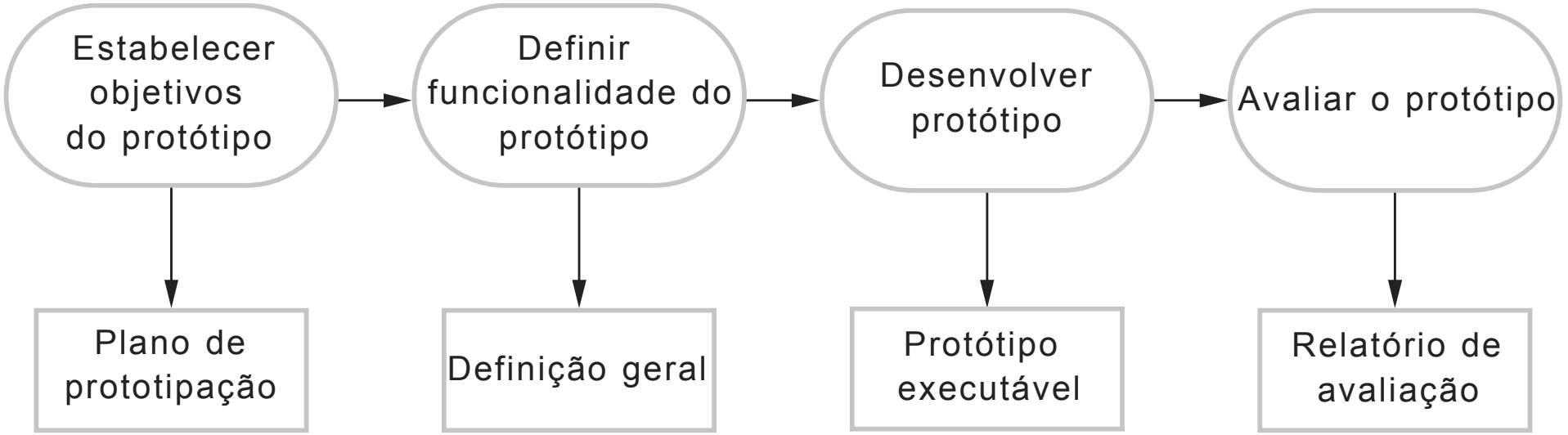
Processo de Prototipação

UFRPE - UAST

- Definir funcionalidade do protótipo:
  - É importante decidir o que colocar e o que deixar de fora do protótipo
  - O protótipo não pode ter custos elevados e nem demorar para ser desenvolvido
    - Pode-se deixar alguma funcionalidade fora do protótipo
    - Pode-se optar por relaxar os requisitos não funcionais
      - Como o tempo de resposta e utilização de memória
  - Padrões de confiabilidade e qualidade de programa podem ser reduzidos

Processo de Prototipação

UFRPE - UAST



O processo de desenvolvimento de protótipo

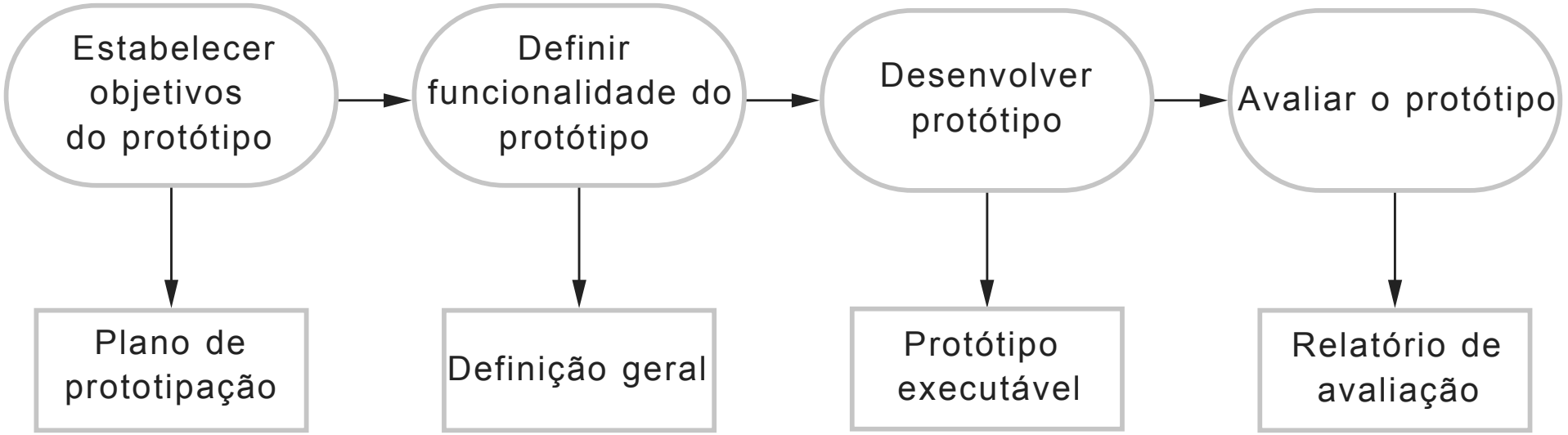
Processo de Prototipação

UFRPE - UAST

- Desenvolver o protótipo
  - Autoexplicativo =)

Processo de Prototipação

UFRPE - UAST



O processo de desenvolvimento de protótipo

Processo de Prototipação

UFRPE - UAST

- Avaliar o protótipo
  - Os objetivos do protótipo devem ser usados para derivar um plano de avaliação
  - Os usuários necessitam de um tempo para se sentir confortáveis
    - Uma vez que estejam usando o sistema normalmente, eles descobrem erros e omissões de requisitos

Problemas da Prototipação

UFRPE - UAST

- O protótipo pode não ser necessariamente usado da mesma forma como no sistema final
- O testador do protótipo pode não ser um usuário típico do sistema
- Às vezes, os desenvolvedores são pressionados pelos gerentes para entregar protótipos descartáveis
  - Especialmente quando há atrasos no desenvolvimento do software

Problemas da Prototipação

UFRPE - UAST

- Resista à pressão de estender um protótipo grosseiro a um produto final

- Quase sempre, como resultado, a qualidade fica comprometida

Problemas da Prototipação

UFRPE - UAST

- Protótipos descartáveis costumam ser desaconselháveis:

  1) Pode ser impossível ajustar o protótipo para atender aos requisitos não funcionais

  - “desempenho, proteção, robustez e confiabilidade”
  - Normalmente ignorados durante o desenvolvimento do protótipo

Problemas da Prototipação

UFRPE - UAST

- Protótipos descartáveis costumam ser desaconselháveis:

  2) O protótipo não é documentado

  - A única especificação de projeto é o código do protótipo
  - Para a manutenção a longo prazo, isso não é bom o suficiente

Problemas da Prototipação

UFRPE - UAST

- Protótipos descartáveis costumam ser desaconselháveis:

  3) Mudanças durante o desenvolvimento do protótipo provavelmente terão degradado a estrutura do sistema

  - O sistema resultante do protótipo será difícil e custoso de ser mantido

Problemas da Prototipação

UFRPE - UAST

- Protótipos descartáveis costumam ser desaconselháveis:
  - 4) Falta de padrões de qualidade
    - Padrões geralmente são relaxados para o desenvolvimento do protótipo

Considerações finais sobre prototipação

UFRPE - UAST

- Protótipos não precisam ser executáveis para serem úteis
- Interface de usuário do sistema pode ser ilustrada em papel
  - Ou em ferramentas de "*mockup*"
- Estes são muito baratos de se desenvolver e podem ser construídos em poucos dias

UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

UNIDADE ACADÊMICA DE SERRA TALHADA

Desenvolvimento e Entrega incremental

Desenvolvimento incremental

UFRPE - UAST

- É baseado na seguinte ideia:
  - Desenvolver uma implementação inicial
  - Expô-la aos comentários dos usuários
  - Continuar por meio da criação de várias versões até que um sistema adequado seja desenvolvido

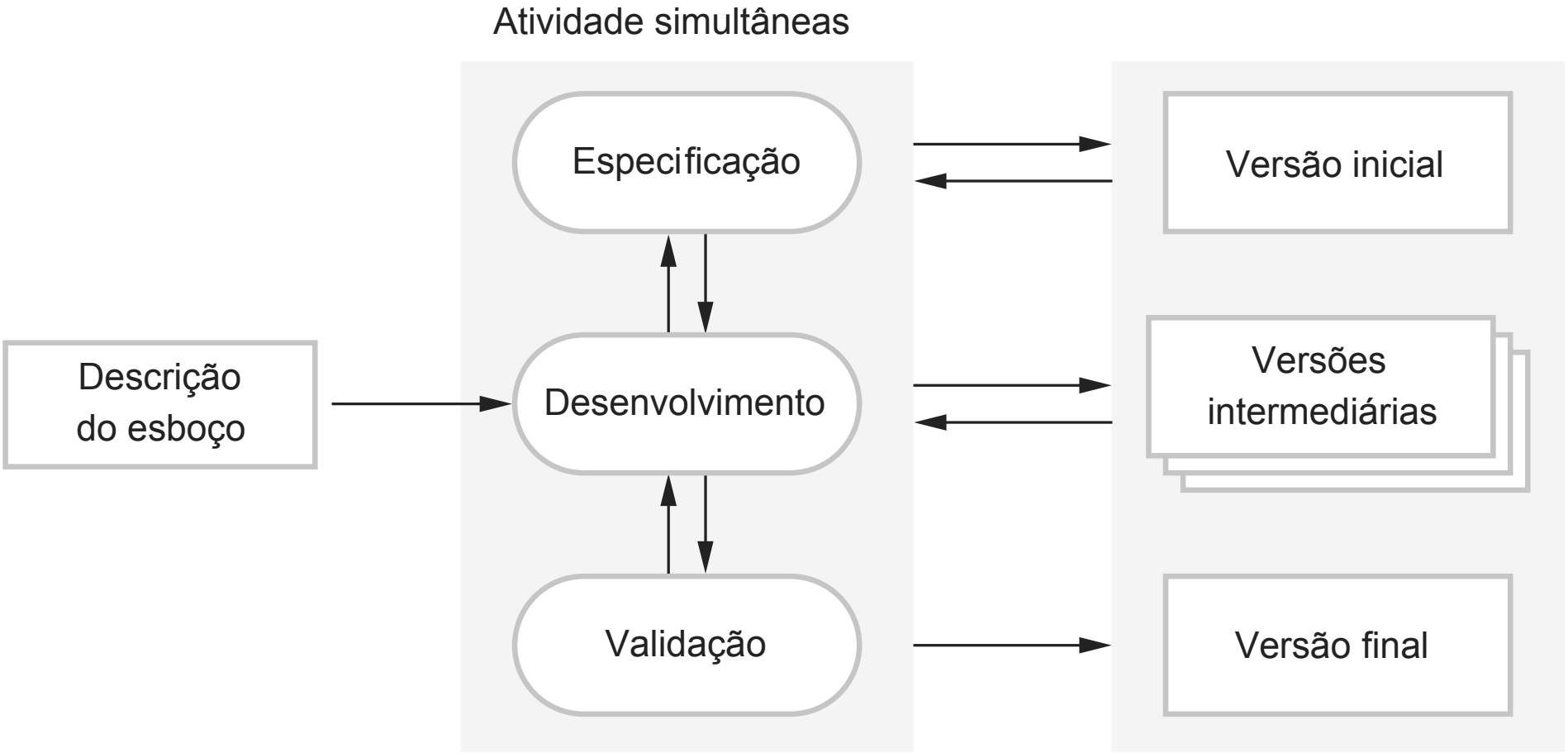
Desenvolvimento incremental

UFRPE - UAST

- Atividades de:
  - Especificação
  - Desenvolvimento
  - Validação
- São *intercaladas*, e não separadas...
  - Ao contrário do *waterfall*...
- Com rápido *feedback* entre todas as atividades

Desenvolvimento incremental

UFRPE - UAST



Desenvolvimento incremental

Desenvolvimento incremental

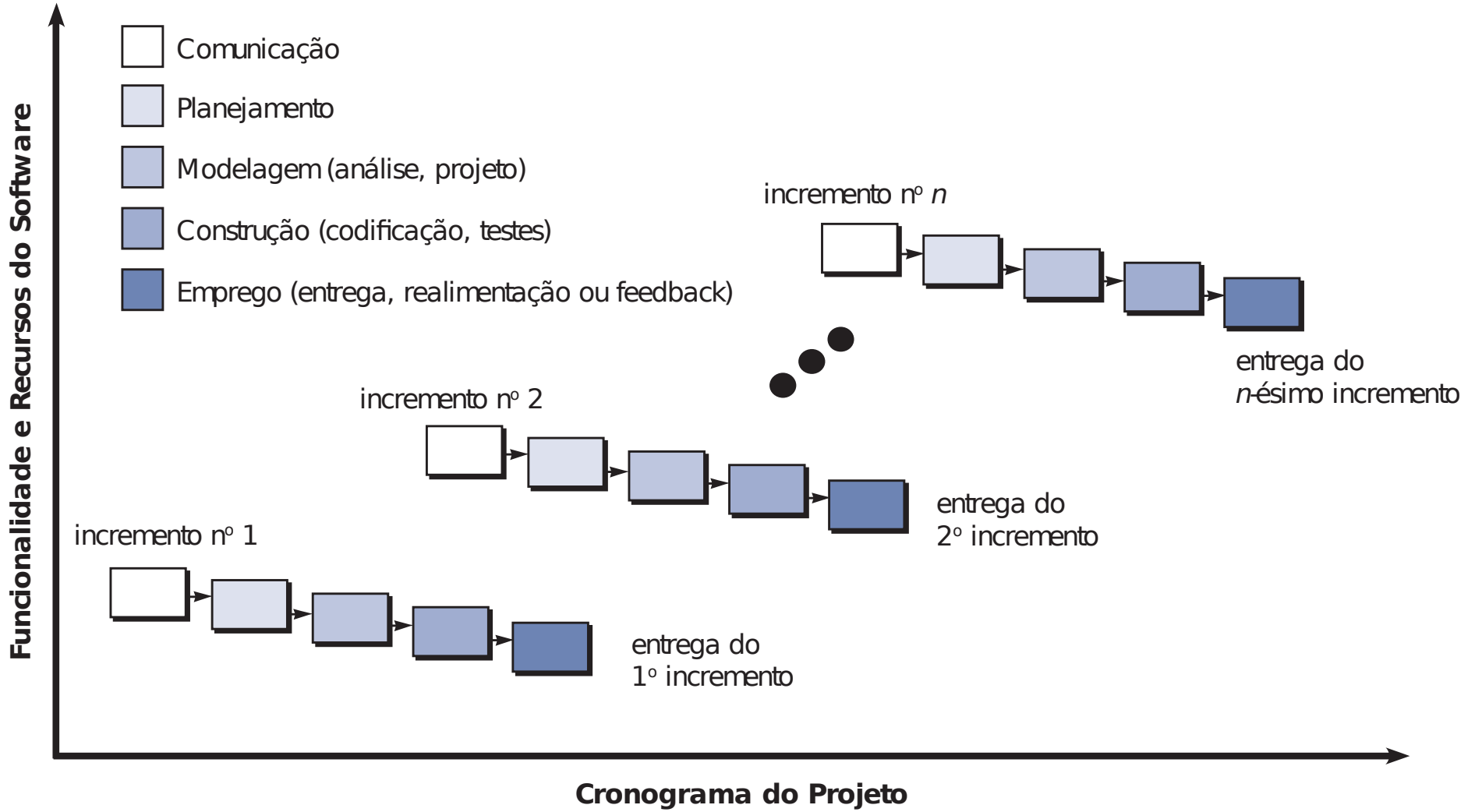
UFRPE - UAST

- Combina elementos dos fluxos de processos lineares e paralelos
  - Ou seja, aplica sequências lineares, de forma escalonada, à medida que o tempo vai avançando

- Cada sequência linear gera “incrementos” (entregáveis/aprovados/liberados) do software

Desenvolvimento incremental

UFRPE - UAST



Desenvolvimento incremental

UFRPE - UAST

- Frequentemente, o primeiro incremento é um produto essencial
  - Requisitos básicos (críticos) são atendidos
- Porém, muitos recursos complementares (alguns conhecidos, outros não) ainda não são entregues
  - Foco voltado para a entrega de um produto operacional com cada incremento

Desenvolvimento incremental

UFRPE - UAST

- Desenvolvimento incremental de software, é uma parte fundamental das abordagens ágeis!
  - É melhor do que uma abordagem *waterfall* para a maioria dos sistemas!

- Essa abordagem reflete a maneira como resolvemos os problemas
  - Raramente elaboramos uma completa solução do problema com antecedência
  - Realizamos passo a passo em direção a uma solução
    - Recuando quando percebemos que cometemos um erro!

Desenvolvimento incremental

UFRPE - UAST

- Ao desenvolver um software de forma incremental:
  - É mais barato e mais fácil fazer mudanças no software durante seu desenvolvimento

Desenvolvimento incremental

UFRPE - UAST

- Cada incremento ou versão do sistema incorpora alguma funcionalidade necessária para o cliente

- Os incrementos iniciais incluem a funcionalidade mais importante...
  - Ou mais urgente!

Desenvolvimento incremental

UFRPE - UAST

- O cliente pode avaliar o sistema em um estágio relativamente inicial
  - Para ver se ele oferece o que foi requisitado!
  - Em caso negativo, só o incremento que estiver em desenvolvimento no momento precisará ser alterado/refeito

Desenvolvimento incremental

UFRPE - UAST

- O desenvolvimento incremental tem três vantagens importantes quando comparado ao modelo *waterfall*:

  1) O custo de acomodar as mudanças nos requisitos do cliente é reduzido

  - A quantidade de análise e documentação a ser refeita é muito menor do que o necessário no modelo *waterfall*

Desenvolvimento incremental

UFRPE - UAST

- O desenvolvimento incremental tem três vantagens importantes quando comparado ao modelo *waterfall*:

  2) É mais fácil obter *feedback* dos clientes sobre o desenvolvimento que foi feito

  - Os clientes podem fazer comentários sobre as demonstrações do software e ver o quanto foi implementado
  - Os clientes têm dificuldade em avaliar a evolução por meio de documentos de projeto de software
    - Abordagem utilizada no modelo *waterfall*

Desenvolvimento incremental

UFRPE - UAST

- O desenvolvimento incremental tem três vantagens importantes quando comparado ao modelo *waterfall*:

  3) É possível obter entrega e implementação rápida de um software útil ao cliente

  - Partes incompletas, porém funcionais são entregas periodicamente aos clientes
  - Os clientes podem usar e obter ganhos a partir do software inicial antes do que é possível com um processo *waterfall*

Desenvolvimento incremental

UFRPE - UAST

- Esse modelo é atualmente a abordagem mais comum para o desenvolvimento de softwares

- Existe três formas de uso:
  - Dirigida a planos
  - Ágil
  - Uma mescla dessas abordagens
    - O mais desempenhado na prática

Desenvolvimento incremental

UFRPE - UAST

- Em uma abordagem dirigida a planos:
  - Os incrementos do sistema são identificados previamente
- Em uma abordagem ágil:
  - Os incrementos iniciais são identificados
    - Mas o desenvolvimento de incrementos posteriores depende do progresso e das prioridades dos clientes

Problemas do desenvolvimento incremental

UFRPE - UAST

- Do ponto de vista do gerenciamento, a abordagem incremental tem dois problemas:

  1) O processo não é visível

     - Os gerentes precisam de entregas regulares para mensurar o progresso

     - Não é economicamente viável produzir documentos que reflitam cada um dos incrementos (versões do sistema)

Problemas do desenvolvimento incremental

UFRPE - UAST

- Do ponto de vista do gerenciamento, a abordagem incremental tem dois problemas:

  2) A estrutura do sistema tende a se degradar com a adição dos novos incrementos

  - As constantes mudanças no código fonte tendem a degradar a sua estrutura

  - Incorporar futuras mudanças do software torna-se cada vez mais difícil e oneroso
    - A menos que tempo e dinheiro sejam dispendidos em refatoração para melhoria do software

Problemas do desenvolvimento incremental

UFRPE - UAST

- Os problemas são particularmente críticos para os sistemas de vida-longa, grandes e complexos
  - Necessitam de uma arquitetura estável
- Normalmente, tais sistemas são planejados com antecedência
  - Não sendo desenvolvido de forma incremental

Problemas do desenvolvimento incremental

UFRPE - UAST

- O desenvolvimento iterativo também pode ser difícil quando um sistema substituto está sendo desenvolvido
  - Usuários querem toda a funcionalidade do sistema antigo! (entregues de uma só vez)
    - E não apenas os requisitos mais críticos
  - Por tanto, muitas vezes, ficam relutantes em experimentar um novo sistema que está incompleto

Vantagens da entrega incremental

UFRPE - UAST

- Quais são as vantagens você enxerga?

Vantagens da entrega incremental

UFRPE - UAST

- Os clientes podem usar os incrementos iniciais como protótipos e ganhar experiência
  - Melhora o entendimento do que é desejado
    - Domínio do problema
  - A qual informa seus requisitos para incrementos posteriores do sistema
- Ao contrário de protótipos, trata-se, aqui, de partes do sistema real
  - Os requisitos não são relaxados nas versões parciais

Vantagens da entrega incremental

UFRPE - UAST

- Os clientes não necessitam esperar até a versão final
  - Desde o início o cliente pode obter ganhos com versões parciais
  - Incrementos iniciais satisfazem os requisitos mais críticos
    - Possibilitando usar o software em estágios iniciais do projeto

Vantagens da entrega incremental

UFRPE - UAST

- Os requisitos mais críticos recebem a maioria dos testes
  - Pois são desenvolvidos em estágios muitos iniciais
    - Onde normalmente ainda não ocorreram atrasos significativos no cronograma
  - Também significa que a probabilidade dos clientes encontrarem falhas nas partes críticas é menor
    - Afinal, esses requisitos são testados com um maior rigor

Considerações finais sobre desenvolvimento incremental

UFRPE - UAST

- É uma abordagem na qual alguns dos incrementos desenvolvidos são entregues ao cliente
  - Isso acontece mesmo no início do projeto

- Cada incremento proporciona um subconjunto da funcionalidade do sistema
  - “Versões parciais”
  - Os incrementos depende da ordem de prioridade do cliente
    - Os requisitos de mais alta prioridade são implementados e entregues em primeiro lugar

Referências

UFRPE - UAST

- 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.