UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

UNIDADE ACADÊMICA DE SERRA TALHADA

Modelo espiral e Desenvolvimento orientado a reúso

Ygor Amaral <ygor.amaral@ufrpe.br>

Disciplina: Processo de Desenvolvimento de Software

Curso: Sistemas de Informação (2016.1)

Modelo Espiral

Modelo Espiral

UFRPE - UAST

- É um framework de processo de software dirigido a riscos
  - O processo de software é representado como uma espiral!
    - Ao invés de uma sequência de atividades com alguns retornos de uma para outra...
  - Cada volta na espiral representa uma fase do processo de software

Modelo Espiral

UFRPE - UAST

- Por exemplo:
  - A volta mais interna pode preocupar-se com a viabilidade do sistema
  - O ciclo seguinte, com definição de requisitos
  - O seguinte, com o projeto do sistema, e assim por diante...

Modelo Espiral

UFRPE - UAST

- Esse modelo combina:
  - Prevenção e tolerância a mudanças
- Assume que mudanças são um resultado de riscos de projeto
  - Inclui atividades explícitas de gerenciamento de riscos para sua redução

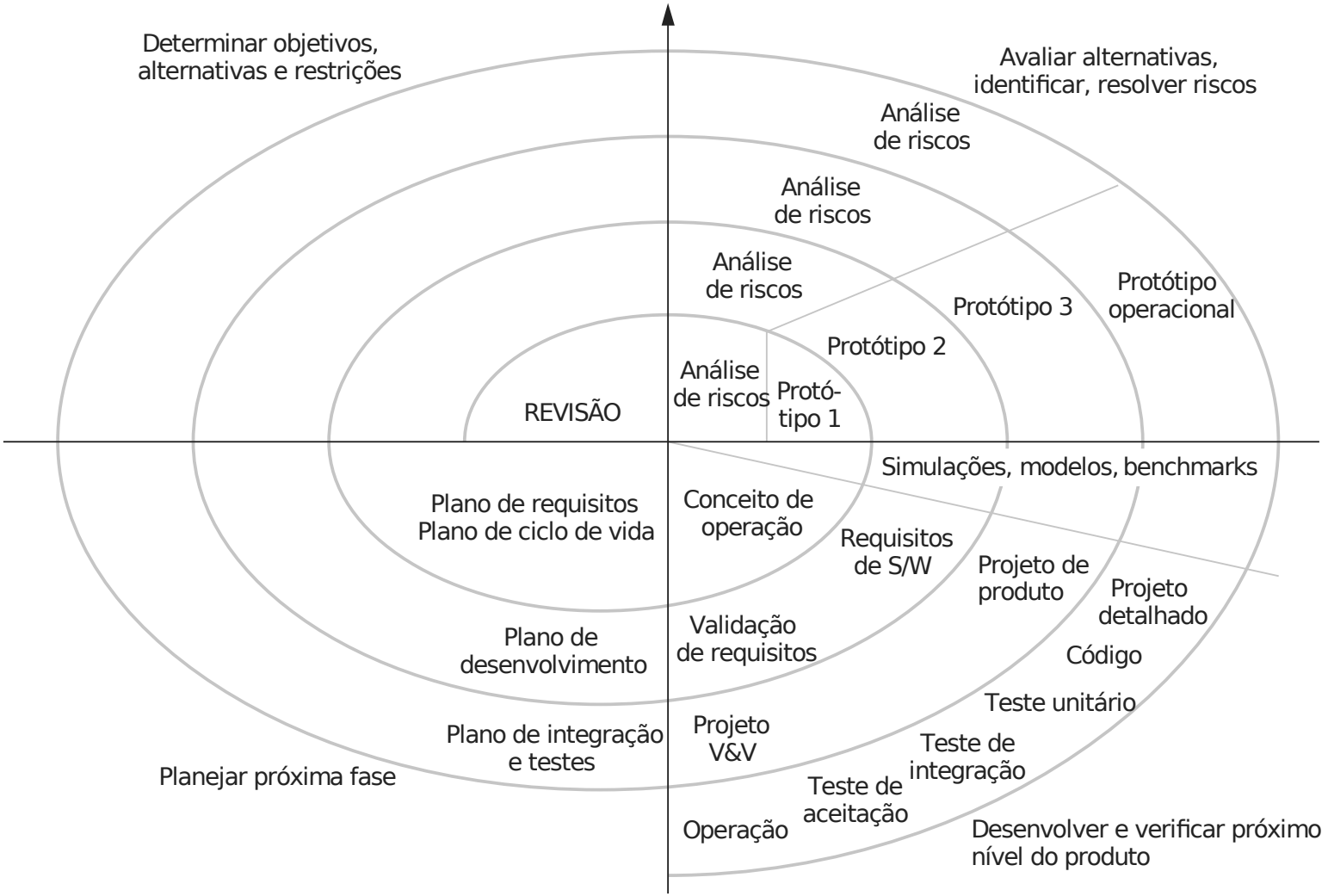
Modelo Espiral

UFRPE - UAST

- Cada volta da espiral é dividida em quatro setores:
  - Definição de objetivos
  - Avaliação e redução de riscos
  - Desenvolvimento e validação
  - Planejamento da próxima fase

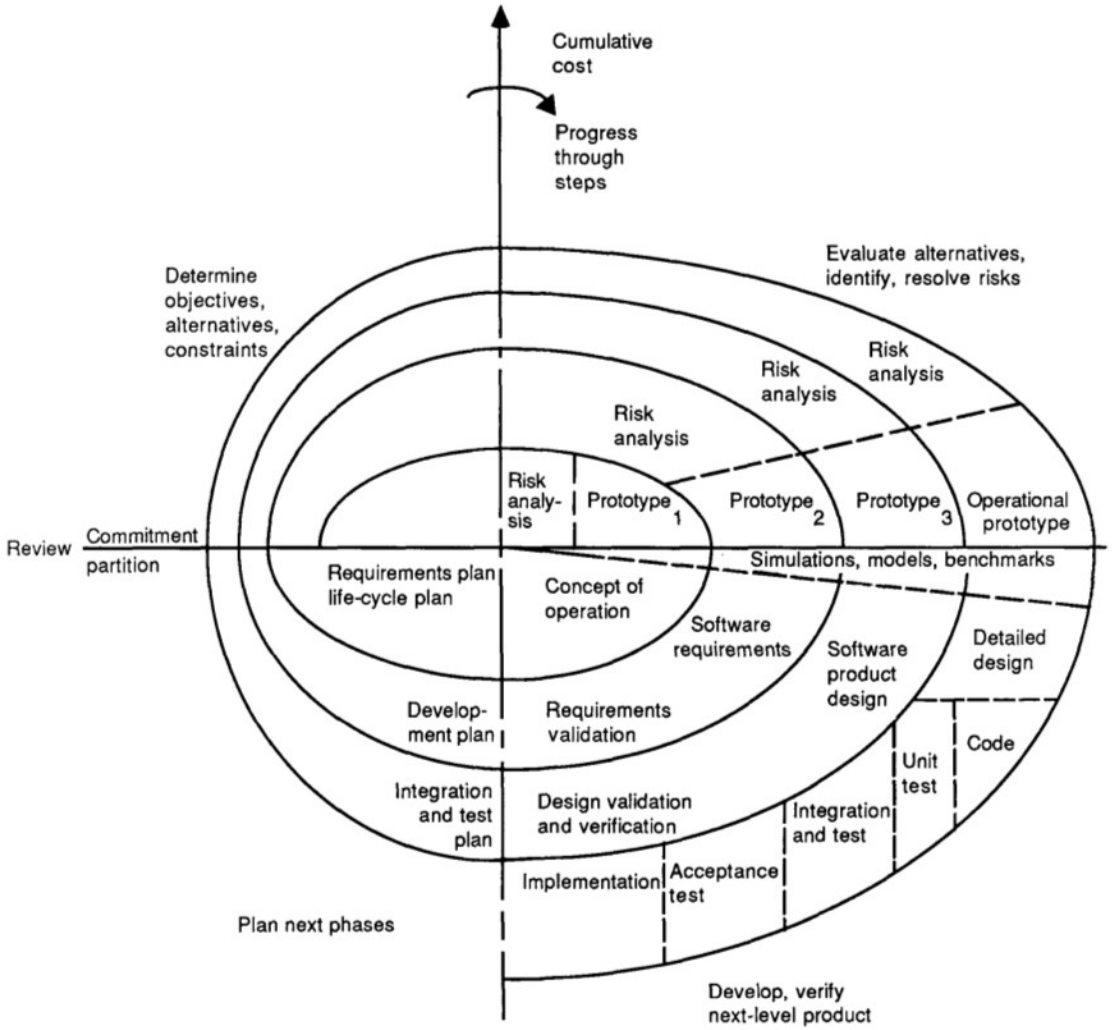
Modelo Espiral

UFRPE - UAST



Modelo Espiral

UFRPE - UAST



Modelo Espiral

UFRPE - UAST

- Setor 1 → Definição de objetivos:
  - Objetivos específicos para essa fase do projeto são definidos
  - Restrições ao processo e ao produto são identificadas
  - Um plano de gerenciamento detalhado é elaborado
  - Os riscos do projeto são identificados
  - *Podem ser planejadas estratégias alternativas em função desses riscos

Modelo Espiral

UFRPE - UAST

- Setor 2 → Avaliação e redução de riscos:
  - Para cada um dos riscos identificados do projeto, é feita uma análise detalhada
    - Um risco significa, simplesmente, algo que pode dar errado
    - Riscos podem levar a mudanças no software e problemas de projeto, como estouro de prazos e custos
  - Medidas para redução do risco são tomadas
    - Tentar reduzir os riscos antes de se tornarem problemáticos
    - Por ex: se houver risco dos requisitos serem inadequados, um protótipo de sistema pode ser desenvolvido

Modelo Espiral

UFRPE - UAST

- Setor 3 → Desenvolvimento e validação:
  - Após a avaliação dos riscos, “uma parte” do software precisa finalmente ser desenvolvida
  - Para passar por esse setor, o que foi desenvolvido precisa ser validado

Modelo Espiral

UFRPE - UAST

- Setor 4 → Planejamento:
  - O projeto é revisado!
    - E uma decisão é tomada a respeito da continuidade do modelo com mais uma volta da espiral
  - Caso se decida pela continuidade, planos são elaborados para a próxima fase do projeto

Modelo Espiral

UFRPE - UAST

- A primeira volta na espiral pode resultar no desenvolvimento de uma especificação de produto

- Passagens subsequentes em torno da espiral podem ser usadas para desenvolver um protótipo
  - Progressivamente, versões cada vez mais sofisticadas do software

Modelo Espiral

UFRPE - UAST

- Cada passagem pela região de planejamento resulta em ajustes no planejamento do projeto

- Custo e cronograma são ajustados de acordo com o *feedback* obtido do cliente após entrega
  - A entrega pode (deveria) acontecer ao final de cada volta da espiral

Modelo Espiral

UFRPE - UAST

- O software será desenvolvido em uma série de versões evolucionárias
  - Versões parciais ficando completas com o passar do tempo
  - Nas primeiras iterações, a versão pode consistir em um modelo ou em um protótipo
  - Nas iterações posteriores, são produzidas versões cada vez mais completas do sistema que passa pelo processo de engenharia

Modelo Espiral

UFRPE - UAST

- A principal diferença entre o modelo espiral e outros modelos de processo de software é seu reconhecimento explícito do risco

Modelo Espiral

UFRPE - UAST

- O modelo espiral pode ser adaptado para ser aplicado ao longo da vida do software
  - Manutenção...

- Nessa forma, a espiral permanece em operação até que o software seja retirado de operação
  - Há casos em que o processo fica inativo, porém, toda vez que uma mudança é iniciada, começa no ponto de partida apropriado da espiral

Modelo Espiral

UFRPE - UAST

- É uma abordagem realista para o desenvolvimento de sistemas e de software em larga escala
  - Não é ideal para pequenos projetos, assim com o *waterfall*
- Nesse modelo, o software evolui à medida que o processo avança
  - O desenvolvedor e o cliente compreendem e reagem melhor aos riscos em cada nível evolucionário
  - Essa característica torna o processo de desenvolvimento do projeto mais visível para o cliente

Modelo Espiral

UFRPE - UAST

- Esse modelo usa a prototipação como mecanismo de redução de riscos

- Mantém a abordagem em etapas, de forma sistemática, sugerida pelo ciclo de vida clássico
  - Entretanto, incorpora uma metodologia iterativa que reflete mais realisticamente o mundo real

Modelo Espiral

UFRPE - UAST

- Se a gerência ou cliente quiser um desenvolvimento com orçamento fixo
  - “Geralmente uma ideia ruim”
  - A espiral pode ser um problema nesse caso

- À medida que cada volta for realizada, o custo do projeto será repetidamente revisado

UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

UNIDADE ACADÊMICA DE SERRA TALHADA

Desenvolvimento Orientado a Reúso

Desenvolvimento orientado a reúso

UFRPE - UAST

- Na maioria dos projetos de software, há algum reúso de software

- Isso acontece muitas vezes informalmente
  - Quando as pessoas envolvidas no projeto sabem de projetos ou códigos semelhantes ao que é exigido
    - Buscam, fazem as modificações necessárias e incorporam-nos a seus novos sistemas

Desenvolvimento orientado a reúso

UFRPE - UAST

- O reúso informal ocorre independentemente do processo de desenvolvimento que se use

- No entanto, processos de desenvolvimento de software com foco no reúso de software existente foram criados

Desenvolvimento orientado a reúso

UFRPE - UAST

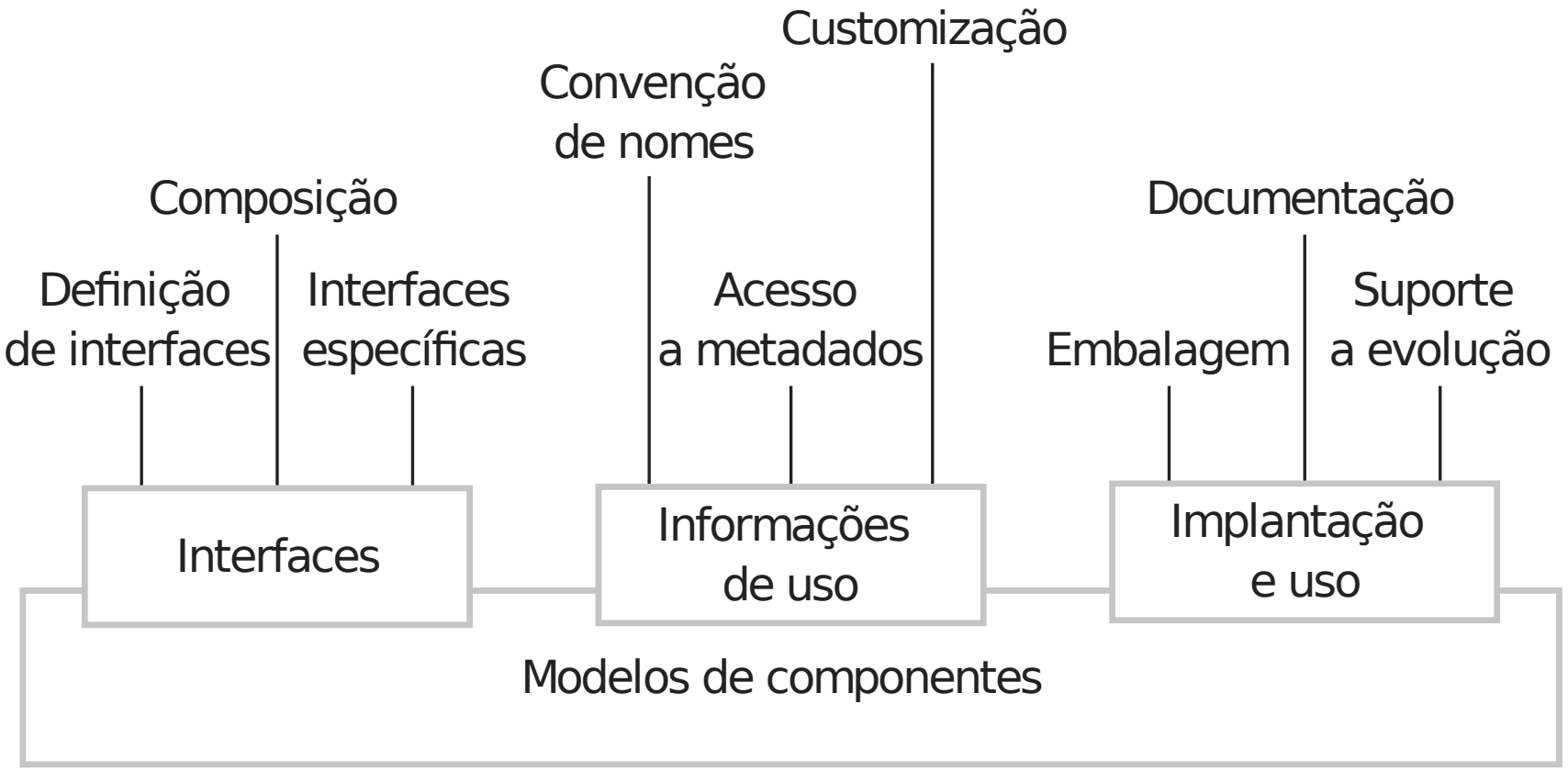
- Abordagens orientadas a reúso dependem de uma ampla base de componentes reusáveis

- Esses componentes são capazes de fornecer uma funcionalidade específica

Desenvolvimento orientado a reúso

UFRPE - UAST

Elementos básicos de um modelo de componentes



Desenvolvimento orientado a reúso

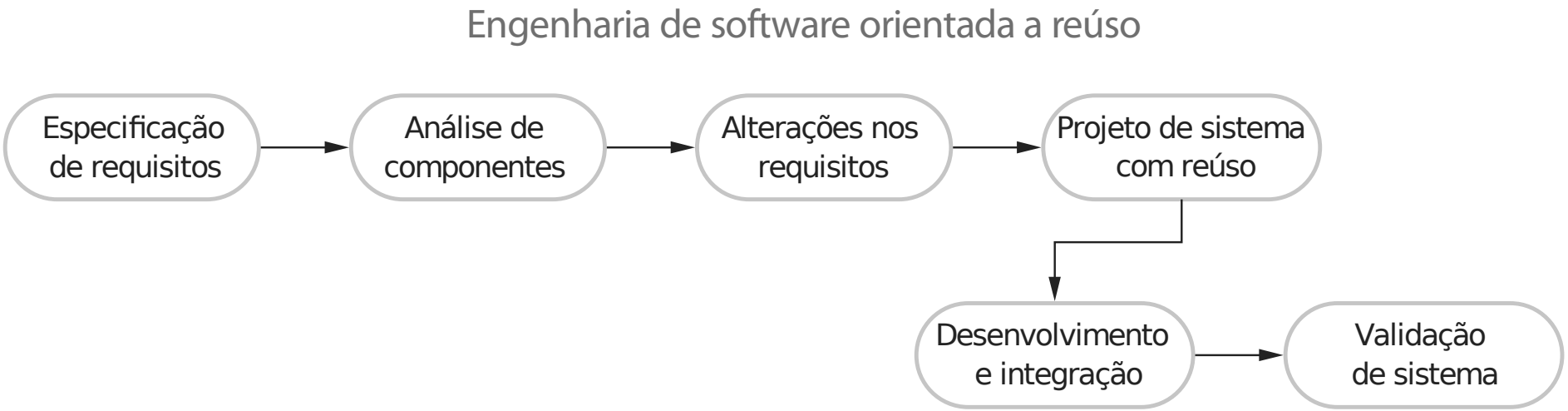
UFRPE - UAST

- Embora os estágios iniciais e finais sejam comparáveis a outros processos de software...
  - Os estágios intermediários em um processo orientado a reúso são diferentes!

- Os estágios intermediários são:
  - Análise de componentes
  - Alterações nos requisitos
  - Projeto de sistema com reúso
  - Desenvolvimento e integração

Desenvolvimento orientado a reúso

UFRPE - UAST



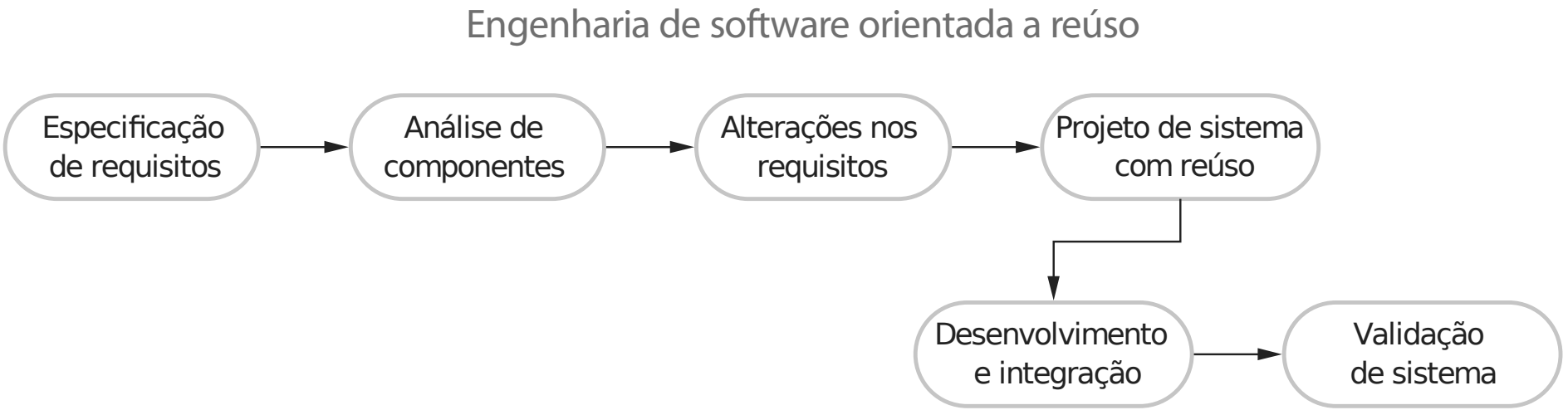
Desenvolvimento orientado a reúso

UFRPE - UAST

- Estágio: Análise de componentes
  - Dada a especificação de requisitos, é feita uma busca por componentes para implementar essa especificação
  - É comum não existir uma correspondência exata
    - Os componentes podem ser usados apenas para fornecer alguma funcionalidade específica

Desenvolvimento orientado a reúso

UFRPE - UAST



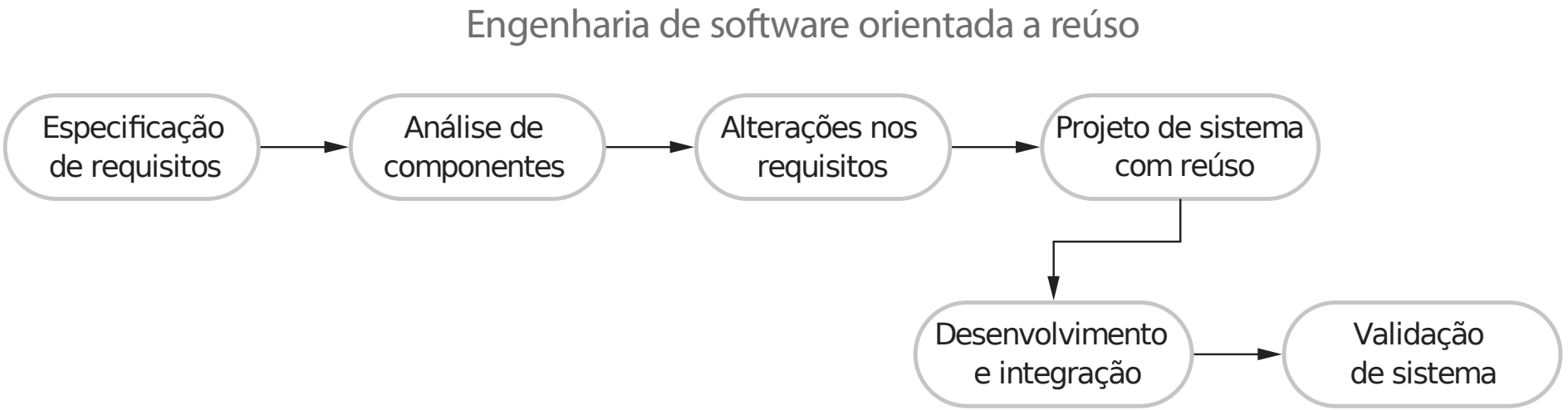
Desenvolvimento orientado a reúso

UFRPE - UAST

- Estágio: Alterações nos requisitos
  - Os requisitos são analisados usando-se informações sobre os componentes que foram descobertos
  - Em seguida, os requisitos serão modificados para refletir os componentes disponíveis
    - No caso de modificações impossíveis, a atividade de análise dos componentes pode ser reinserida na busca por soluções alternativas

Desenvolvimento orientado a reúso

UFRPE - UAST



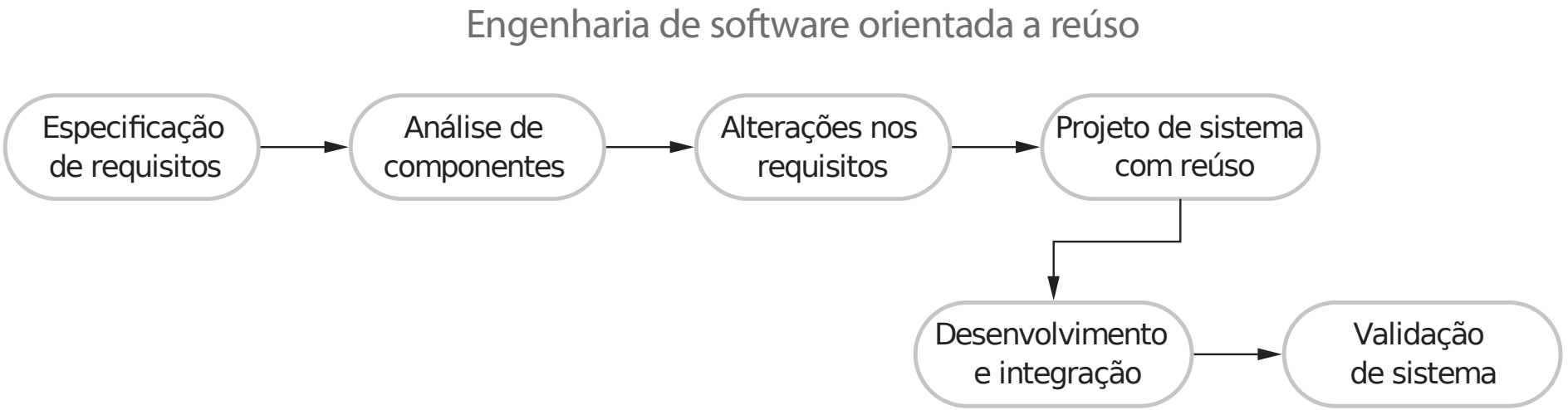
Desenvolvimento orientado a reúso

UFRPE - UAST

- Estágio: Projeto de sistema com reúso
  - O framework do sistema é projetado ou algo existente é reusado
  - Os projetistas têm em mente os componentes que serão reusados e organizam o framework para reúso
    - Alguns softwares novos podem ser necessários, se componentes reusáveis não estiverem disponíveis

Desenvolvimento orientado a reúso

UFRPE - UAST



Desenvolvimento orientado a reúso

UFRPE - UAST

- Estágio: Desenvolvimento e integração
  - Softwares (componentes) que não podem ser adquiridos externamente são desenvolvidos
  - Posteriormente, todos os componentes são integrados para criar o novo sistema
    - A integração de sistemas, nesse modelo, pode ser parte do processo de desenvolvimento
      - Ao invés de uma atividade separada

Desenvolvimento orientado a reúso

UFRPE - UAST

- Existem três tipos de componentes de software que podem ser usados em um processo orientado a reúso:
  - Web services desenvolvidos de acordo com os padrões de serviço e que estão disponíveis para invocação remota
    - Muito comum hoje em dia....
  - Coleções de objetos que são desenvolvidas como um pacote a ser integrado com um framework de componentes
    - Bastante comum em .NET ou Java EE
  - Sistemas de software stand-alone configurados para uso em um ambiente particular
    - Ex: MaryTTS

Desenvolvimento orientado a reúso

UFRPE - UAST

- O desenvolvimento de software orientado a reúso tem a vantagem óbvia de reduzir a quantidade de software a ser desenvolvido
  - Consequentemente reduzir os custos e riscos
  - Geralmente, também proporciona a entrega mais rápida do software
  - Uma componente específico provavelmente terá sido desenvolvido melhor do que você fará em um prazo estabelecido
    - Redes neurais
    - Acesso a banco de dados
    - Relatórios

    ...

Desenvolvimento orientado a reúso

UFRPE - UAST

- Componentes devem ser avaliados com cuidado e seriedade

- Compromissos com os requisitos são inevitáveis!
  - Isso pode levar a um sistema que não atende às reais necessidades dos usuários

- Além disso:
  - Algum controle sobre a evolução do sistema é perdido
  - Novas versões dos componentes reusáveis não estão sob o controle da organização que os está utilizando

Desenvolvimento orientado a reúso

UFRPE - UAST

- Tornou-se o paradigma de desenvolvimento dominante para sistemas de informação
  - Sejam baseados na Web ou sistemas corporativos

Benefícios do reúso de software

UFRPE - UAST

- Confiança aumentada
  - Os softwares reusados são experimentados e testados em diversos sistemas em funcionamento
  - Provavelmente são mais confiáveis do que um novo software desenvolvido por você

- Risco de processo reduzido
  - Reduz a margem de erro de estimativa de custos de projeto
    - O custo do software existente já é conhecido
    - Os custos de desenvolvimento são sempre uma questão que envolve um novo julgamento

Benefícios do reúso de software

UFRPE - UAST

- Uso eficaz de especialistas
  - Desenvolver novas soluções ao invés de reinventar a roda
  - Especialistas em aplicações podem desenvolver softwares reusáveis que encapsulem seu conhecimento

Benefícios do reúso de software

UFRPE - UAST

- Conformidade com padrões
  - Alguns padrões, como os de interface de usuário, podem ser implementados como um conjunto de componentes reusáveis
    - Swing, JavaFX, Qt, GTK...
      - Se vários programas tiverem interfaces de usuário implementadas usando componentes reusáveis, todas as aplicações apresentarão os mesmos formatos para os usuários

Benefícios do reúso de software

UFRPE - UAST

- Desenvolvimento acelerado
  - O reúso de um software pode acelerar a produção do sistema, pois pode reduzir o tempo de desenvolvimento e validação
    - Mesmo nos projetos que os custos gerais de desenvolvimento não são tão importantes

Problemas com reúso

UFRPE - UAST

- Maiores custos de manutenção
  - Nem sempre os códigos fontes dos componentes reusáveis estão disponíveis
    - Os custos de manutenção podem ser maiores
  - Os elementos reusados do sistema podem tornar-se cada vez mais incompatíveis com as alterações do sistema

Problemas com reúso

UFRPE - UAST

- Síndrome de ‘não-inventado-aqui’
  - ISSO É MUITO COMUM!
  - Alguns engenheiros de software preferem reescrever componentes, pois acreditam poder melhorá-los
    - Isso tem a ver, parcialmente, com aumentar a confiança
  - O fato de que escrever softwares originais é considerado mais desafiador do que reusar softwares de outras pessoas
    - “Eu sou o cara!”
    - “Entendo 100% do código fonte!”

Problemas com reúso

UFRPE - UAST

- Encontrar, compreender e adaptar os componentes reusáveis
  - Componentes de software precisam ser descobertos e compreendidos
    - É comum precisar adaptar algum componente para trabalhar em um novo ambiente
  - Os engenheiros precisam estar confiantes de que encontrarão, na biblioteca, um componente desejado
    - Antes de incluírem a pesquisa de componente como parte de seu processo normal de desenvolvimento