PUCRS
Pontifícia Universidade Católica
do Rio Grande do Sul

ESCOLA
POLITÉCNICA

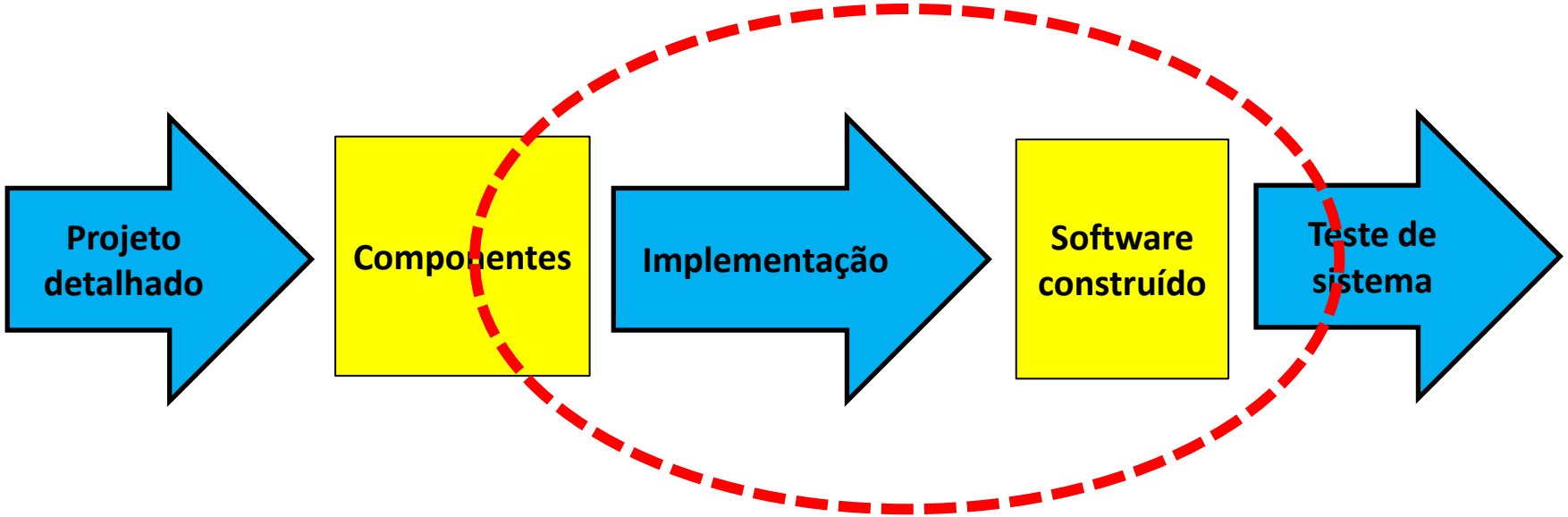
Construção de Software
Conceitos

46504-04 - Construção de Software

Prof. Msc. Eduardo Arruda
eduardo.arruda@pucrs.br

Material original elaborado pelo Prof. Dr.
Marcelo Yamaguti

Construção



- Construção:
  - Criação de versão do software:
    - Desenvolvimento de software com linguagens de programação; e/ou
    - Customização/adaptação de software de prateleira; e/ou
    - Integração de componentes de software.

Construção

- Fundamentos de construção
  - Minimizar a complexidade
  - Antecipar a mudança
  - Construir para a verificação
  - Padrões na construção
  - Reúso

Fundamentos de Construção de Software

- Minimizar a complexidade
  - Limitação de lidar com muita informação ao mesmo tempo
  - Minimizar a complexidade auxilia o teste
  - Buscar código simples e legível em vez de código astuto
  - Como obter:
    - Projeto modular
    - Técnicas de codificação
    - Técnicas de qualidade de construção

Fundamentos de Construção de Software

- Projeto modular (características desejáveis):
  - **Complexidade mínima**: cuidado com soluções muito “engenhosas”.
  - **Alta coesão**: manter o mínimo de funcionalidade necessária.
  - **Baixo acoplamento**: manter inter-relacionamento entre módulos baixo.
  - **Alto fan-in**: considerar um alto número de “clientes”, sem sacrificar a coesão.
  - **Baixo fan-out**: manter no máximo 7 chamadas a outros módulos.
  - **Facilidade de manutenção**: pense numa solução a partir do ponto de vista de manutenção.
  - **Extensibilidade**: avaliar a possibilidade da solução ser extensível.
  - **Reutilização**: considerar se a solução pode ser reutilizada em outros sistemas.
  - **Portabilidade**: prever a que possa ser facilmente portado para outra plataforma.
  - **Estratificação**: manter organização entre módulos (ex.: camadas).
  - **Técnicas-padrão**: evite utilizar estratégias exóticas.

Fundamentos de Construção de Software

- Antecipar a mudança
  - O software pode evoluir com o passar do tempo
  - Como obter
    - Projetar uma solução extensível a partir de uma estrutura básica inicial
    - Aplicar princípios de construção de código

Fundamentos de Construção de Software

- Construir para verificação
  - Foco na construção que facilite a localização de defeitos caso haja falhas
  - Como obter:
    - Padrões de codificação para revisão de código
    - Teste unitário
    - Teste automatizado
    - Limitar o uso de estruturas complexas de uma linguagem

Fundamentos de Construção de Software

- Reuso
  - Pode ocorrer em vários níveis:
    - Reuso de experiência
    - Reuso de código
    - Reuso de projeto
    - Reuso de sistemas (COTS – *Commercial-Off-The-Shelf*, software de prateleira)
  - Pode ter dois focos:
    - Implementar **com** reúso: visa a reutilização de software existente.
    - Implementar **para** reúso: visa a criação de software que possa ser reutilizado.

Fundamentos de Construção de Software

- Padrões na construção:
  - Utilizar padrões de desenvolvimento para obter maior eficiência, qualidade, custo e segurança.
  - Como obter:
    - Utilizar versões padronizadas de linguagens de programação (ex.: ANSI)
    - Utilizar padrões para especificação (ex.: UML), comunicação (documentos) e para codificação
    - Buscar utilizar soluções padronizadas (ex.: OMG, ISO, IEEE)

Construção

- Gerenciamento:
  - As atividades de construção devem ser gerenciadas
    - Modelo de ciclo de vida
    - Planejamento
    - Medição

Construção

- Considerações práticas
  - Projeto (*design*) da solução
  - Escolha da linguagem de programação
  - Boas práticas de codificação
  - Teste da solução
  - Garantia da qualidade

Construção

- Ferramentas de construção
  - IDE (*Integrated Development Enviroment*). Ex: Visual Studio Code, BlueJ, Eclipse, Visual Studio
  - Geradores de GUI (*Graphical User Interface*)
  - Ferramentas de teste unitário
  - Ferramentas de depuração (*debugging*)
  - Geradores de código
  - Geradores de aplicação (*MDA-Model Driven Architecture*)
  - Geradores de documentação
  - Gerenciador de configuração (versionamento, integração, rastreabilidade)

Referências

- Estude para aprofundamento no conteúdo:
  - SOMMERVILLE, I. **Engenharia de software**. 9ª ed. São Paulo: Pearson Brasil, 2011. – Capítulos 7 e 24
  - PFLEEGER, S. L. **Engenharia de Software**: teoria e prática. 2ª ed. São Paulo: Prentice Hall, 2004. – Capítulo 7.
  - IEEE Computer Society. **Guide to the Software Engineering Body of Knowledge (SWEBOK)**: Version 3.0. IEEE Computer Society Press, 2014. – Chapters 2 e 3.
  - McConnell, Steve. **Code complete**: a practical handbook of software construction. Redmont: Microsoft Press, 1993. – Capítulos 2 a 7.

Dúvidas?