

# Construção de Software

Prof. Rubens de Castro Pereira

## Lista de Verificação – Arquitetura

Referência:

- Code Complete – Um guia prático para a construção de software, 2ª ed., Steven McConnell, Bookman – Cap. 3.

Lista de problemas que uma boa arquitetura deve tratar.

### 1. Tópicos Arquitetônicos Específicos

- 1.1. A organização global do *software* (sistema) está clara, incluindo um bom panorama arquitetônico e uma justificativa?
- 1.2. Os principais blocos de construção estão bem definidos, incluindo suas áreas de responsabilidade e suas interfaces com outros blocos?
- 1.3. Todas as funções listadas nos requisitos foram avaliadas de forma sensata, não possuindo blocos de construção a mais nem a menos?
- 1.4. As classes mais importantes estão descritas e justificadas?
- 1.5. O projeto de dados está descrito e justificado?
- 1.6. A organização e o conteúdo do banco de dados estão especificados?
- 1.7. Todas as regras de negócio importantes estão identificadas e seu impacto no sistema está descrito?
- 1.8. Está descrita uma estratégia para o projeto de interface com o usuário?
- 1.9. A interface com o usuário é modularizada de forma que as alterações que as alterações sobre ela não afetam o restante dos programas?
- 1.10. Está descrita e justificada uma estratégia para tratamento de entrada e saída?
- 1.11. As estimativas de uso de recursos e uma estratégia para o seu gerenciamento estão descritas e justificadas para recursos escassos, como *threads*, conexões de banco de dados, *handles*, largura de banda de rede, etc?
- 1.12. Os requisitos de segurança da arquitetura estão descritos?
- 1.13. A arquitetura define provisão de espaço e velocidade para cada classe, subsistema ou área de funcionalidade?
- 1.14. A arquitetura descreve como a extensibilidade (capacidade do sistema crescer/expandir nas funcionalidades) será obtida?
- 1.15. A arquitetura trata da interoperabilidade?
- 1.16. Está descrita uma estratégia de internacionalização / localização?
- 1.17. Foi fornecida uma estratégia coerente para tratamento de erros?
- 1.18. A estratégia de tolerância a falhas está definida?
- 1.19. A praticabilidade técnica de todas as partes do sistema foi estabelecida?
- 1.20. Está especificada uma estratégia de superengenharia (extrema robustez)?

- 1.21. Todas as decisões de compra ou construção necessárias foram incluídas?
- 1.22. A arquitetura descreve como o código reutilizado se adaptará aos outros objetivos arquitetônicos?
- 1.23. A arquitetura está projetada para conciliar as prováveis alterações?

## **2. Qualidade Arquitetônica Geral**

- 2.1. A arquitetura leva em conta todos os requisitos?
- 2.2. Alguma parte foi dotada de superarquitetura ou subarquitetura? As expectativas nessa área estão especificadas explicitamente?
- 2.3. A arquitetura como um todo é conceitualmente consistente?
- 2.4. O projeto de alto nível é independente da máquina e da linguagem que serão usados para implementá-lo?
- 2.5. As motivações para todas as principais decisões foram fornecidas?
- 2.6. Você, no lugar do programador que implementará o sistema, está à vontade com a arquitetura?