

Aplicação da Checklist de Arquitetura

Lista de verificação: arquitetura

Aqui está uma lista de problemas que uma boa arquitetura deve tratar. A lista não se destina a ser um guia abrangente de arquitetura, mas uma maneira prática de avaliar o conteúdo nutritivo do que você obtém no lado do programador da cadeia alimentar do desenvolvimento de *software*. Use esta lista de verificação como ponto de partida para sua própria lista. Assim como acontece com a lista de verificação de requisitos, se você estiver trabalhando em um projeto informal, encontrará alguns itens que nem mesmo precisa pensar a respeito. Mas se você estiver trabalhando em um projeto maior, a maioria dos itens lhe será útil.

Tópicos arquitetônicos específicos

- ☐ A organização global do *software* está clara, incluindo um bom panorama arquitetônico e uma justificativa?
- ☐ Os principais blocos de construção estão bem-definidos, incluindo suas áreas de responsabilidade e suas interfaces com outros blocos?
- ☐ Todas as funções listadas nos requisitos foram avaliadas de forma sensata, não possuindo blocos de construção a mais nem a menos?
- ☐ As classes mais importantes estão descritas e justificadas?
- ☐ O projeto dos dados está descrito e justificado?
- ☐ A organização e o conteúdo do banco de dados estão especificados?
- ☐ Todas as regras de negócio importantes estão identificadas e seu impacto no sistema está descrito?
- ☐ Está descrita uma estratégia para o projeto da interface com o usuário?
- ☐ A interface com o usuário é modularizada de forma que as alterações sobre ela não afetem o restante dos programas?
- ☐ Está descrita e justificada uma estratégia para tratamento de entrada e saída?
- ☐ 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 da rede, etc.?
- ☐ Os requisitos de segurança da arquitetura estão descritos?
- ☐ A arquitetura define a provisão de espaço e velocidade para cada classe, subsistema ou área de funcionalidade?
- ☐ A arquitetura descreve como a extensibilidade será obtida?
- ☐ A arquitetura trata da interoperabilidade?
- ☐ Está descrita uma estratégia de internacionalização/localização?
- ☐ Foi fornecida uma estratégia coerente para tratamento de erros?

- ☐ A estratégia de tolerância a falhas está definida (se houver necessidade de uma)?
- ☐ A praticabilidade técnica de todas as partes do sistema foi estabelecida?
- ☐ Está especificada uma estratégia de superengenharia?
- ☐ Todas as decisões de compra ou construção necessárias foram incluídas?
- ☐ A arquitetura descreve como o código reutilizado se adaptará aos outros objetivos arquitetônicos?
- ☐ A arquitetura está projetada para conciliar as prováveis alterações?

Qualidade arquitetônica geral

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

O Documento de Arquitetura é feito informalmente e com pouca complexidade, tal como todo o trabalho. Com isso temos itens da lista de verificação que não podem ser atendidos e não são relevantes.

Tópicos arquitetônicos específicos

Atende:

- ☐ A organização global do software está clara, incluindo um bom panorama arquitetônico e uma justificativa?
Sim, o software tem seus requisitos e definição de utilidade bem definidos, bem como suas fronteiras com relação ao que o mesmo deve e não deve fazer.
- ☐ Todas as funções listadas nos requisitos foram avaliados de forma sensata, não possuindo blocos de construção a mais nem a menos?
Sim, foram listados somente os requisitos necessários para construção do software, requisitos que foram identificados como fora do escopo ou de impossível implementação já foram retirados.
- ☐ A interface com o usuário é modularizada de forma que as alterações sobre ela não afetem o restante dos programas?
Sim, pelo fato de utilizarmos MVC, as funcionalidades, bem como suas responsabilidades estão bem divididas, sendo assim temos um baixo acoplamento no nosso software.
- ☐ Todas as regras de negócio importantes estão identificadas e seu impacto no sistema está descrito?
Sim. Temos demonstrado isso no pacote de design modelo.
- ☐ A arquitetura descreve como a extensibilidade será obtida?
Sim, temos pacotes do projeto que já estão preparados para receber novas mudanças e adição de novas funcionalidades.
- ☐ As classes mais importantes estão descritas e justificadas?
Sim.
- ☐ A arquitetura está projetada para conciliar as prováveis alterações?
Sim. A organização da arquitetura MVC nos permite isso.
- ☐ Os principais blocos de construção estão bem definidos, incluindo suas áreas de responsabilidade e suas interfaces com outros blocos?
Sim.

Qualidade arquitetônica geral

Atende:

- ☐ As motivações para todas as principais decisões foram fornecidas?
Sim. A cada explicitação de arquitetura principal, temos uma breve explicação sobre.
- ☐ Você no lugar do programador que implementará o sistema, está a vontade com a arquitetura?

Sim. A arquitetura deixa bem explícita as principais funcionalidade que o sistema deve ter.

- ☐ O projeto de alto nível é independente da máquina e da linguagem que serão usadas para implementá-lo?

Sim.

- ☐ A arquitetura com um todo é conceitualmente consistente?

Sim.

- ☐ A arquitetura leva em conta todos os requisitos?

Sim.