

INF016 – Arquitetura de Software

11 – Projetando para Propriedades Não-Funcionais

Sandro Santos Andrade
sandroandrade@ifba.edu.br

Instituto Federal de Educação, Ciência e Tecnologia da Bahia
Departamento de Tecnologia Eletro-Eletrônica
Graduação Tecnológica em Análise e Desenvolvimento de Sistemas



Introdução

- Como os componentes, conectores e configurações impactam na:
 - Eficiência
 - Complexidade
 - Escalabilidade e Heterogeneidade
 - Adaptação
 - Dependability

Eficiência

Eficiência: qualidade que reflete a habilidade de um sistema atender seus requisitos de desempenho e, ao mesmo tempo, minimizar o uso dos recursos do seu ambiente computacional. Em outras palavras, eficiência é uma medida da economia de uso dos recursos do sistema

Eficiência

- Componentes e Eficiência:
 - Mantenha os componentes pequenos
 - Mantenha as interfaces dos componentes simples e compactas
 - Permita múltiplas interfaces para a mesma funcionalidade
 - Separe componentes de processamento dos componentes de dados
 - Separe os dados dos meta-dados

Eficiência

- Conectores e Eficiência:
 - Selecione os conectores cuidadosamente
 - Use conectores *broadcast* com cuidado
 - Faça uso de interações assíncronas sempre que possível
 - Use transparência de localização criteriosamente

Eficiência

- Configurações e Eficiência:
 - Mantenha os componentes que interagem frequentemente próximos um do outro
 - Selecione e localize os conectores na arquitetura de forma cuidadosa
 - Considere como os estilos e padrões arquiteturais impactam na eficiência

Complexidade

Complexidade: grau de dificuldade de compreensão e verificação do projeto/implementação de um sistema ou de um dos seus componentes

Complexidade: propriedade de um software proporcional ao tamanho do sistema, ao número de elementos constituintes, ao tamanho e estrutura interna de cada elemento e ao número e natureza das inter-dependências entre componentes

Complexidade

- Componentes e Complexidade:
 - Separe os *concerns* em componentes diferentes
 - Mantenha somente funcionalidade dentro dos componentes – não interação
 - Mantenha os componentes com alta coesão
 - Esteja ciente do impacto de componentes *COTS* na complexidade
 - Isole os componentes de processamento das mudanças no formato dos dados

Complexidade

- Conectores e Complexidade:
 - Trate os conectores de forma explícita
 - Mantenha dentro dos conectores somente facilidades de interação
 - Separe diferentes *concerns* de interação em diferentes conectores
 - Limite as interações disponibilizadas por cada conector
 - Esteja ciente do impacto de conectores *COTS* na complexidade

Complexidade

- Configurações e Complexidade:
 - Elimine dependências desnecessárias
 - Gerencie todas as dependências de forma explícita
 - Use decomposição hierárquica

Escalabilidade / Heterogeneidade

Escalabilidade: capacidade de um sistema ser adaptado para atender novos requisitos de tamanho e escopo

Heterogeneidade: qualidade de um sistema de *software* consistir de elementos diferentes ou funcionar em ambientes computacionais diferentes

Escalabilidade / Heterogeneidade

- Componentes e Escalabilidade/Heterogeneidade
 - Defina para cada componente um propósito único e claro
 - Defina para cada componente uma interface única e de fácil compreensão
 - Não atribua responsabilidades de interação aos componentes
 - Evite heterogeneidade desnecessária
 - Distribua as fontes de dados
 - Replique os dados quando necessário

Escalabilidade / Heterogeneidade

- Conectores e Escalabilidade/Heterogeneidade
 - Use conectores explícitos
 - Atribua a cada conector uma responsabilidade claramente definida
 - Escolha o conector mais simples que seja adequado para a tarefa
 - Esteja ciente das diferenças entre dependências diretas e indiretas
 - Não introduza funcionalidade de aplicação dentro dos conectores
 - Utilize conectores explícitos para suportar escalabilidade

Escalabilidade / Heterogeneidade

- Configurações e Escalabilidade/Heterogeneidade
 - Evite gargalos
 - Faça uso de processamento paralelo
 - Posicione as fontes de dados próximas dos consumidores de dados
 - Tente tornar a distribuição de dados transparente
 - Use os estilos arquiteturais apropriados

Adaptação

Adaptação: habilidade do *software* satisfazer novos requisitos e se ajustar a novas condições operacionais durante o seu ciclo de vida

Adaptação

- Componentes e Adaptação
 - Atribua a cada componente um propósito único e claro
 - Minimize as inter-dependências entre componentes
 - Evite sobrecarregar os componentes com responsabilidades de interação
 - Separe o processamento dos dados
 - Separe os dados dos meta-dados

Adaptação

- Conectores e Adaptação
 - Atribua a cada conector uma responsabilidade clara
 - Torne os conectores flexíveis
 - Suporte composição de conectores
 - Esteja ciente das diferenças entre dependências diretas e indiretas

Adaptação

- Configurações e Adaptação
 - Use conectores explícitos
 - Torne a distribuição transparente
 - Use estilos arquiteturais apropriados

Dependability

Um sistema é **tolerante a falhas** se é capaz de responder de forma não-abrupta à ocorrência de falhas em *run-time*

Dependability

- Componentes e *Dependability*
 - Controle cuidadosamente as dependências externas entre componentes
 - Disponibilize capacidades de reflexão aos componentes
 - Disponibilize mecanismos de tratamento de exceção apropriados
 - Especifique as invariantes chave de cada componente

Dependability

- Conectores e *Dependability*
 - Utilize conectores que controlam de forma estrita a dependência entre componentes
 - Disponibilize garantias apropriadas de interação aos componentes
 - Suporte *dependability* através de conectores avançados

Dependability

- Configurações e *Dependability*
 - Evite pontos únicos de falha
 - Utilize *backups* dos dados e funcionalidades críticas
 - Suporte o monitoramento não-intrusivo do sistema
 - Suporte adaptação dinâmica

INF016 – Arquitetura de Software

11 – Projetando para Propriedades Não-Funcionais

Sandro Santos Andrade
sandroandrade@ifba.edu.br

Instituto Federal de Educação, Ciência e Tecnologia da Bahia
Departamento de Tecnologia Eletro-Eletrônica
Graduação Tecnológica em Análise e Desenvolvimento de Sistemas

