

Padrões e Desenho de Software

Resumos
2015/2016

João Alegria | 68661

Introdução ao Design de Software

O que é o Design de Software?

→ Design centrado nos objetivos propostos pelo utilizador (*Design bridges the gap*).

- Saber o que é necessário (requisitos de software)
- Para programar um código eficaz e eficiente (fase de construção)



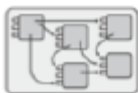
O design é necessário em diferentes níveis do sistema:



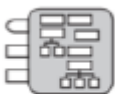
Sistema



Subsistemas ou pacotes: interface com o utilizador, armazenamento de dados, criação de classes e gráficos.



Classes dentro de pacotes, relações entre classes, interface de cada classe, métodos públicos



Atributos, métodos privados, classes internas



Implementação de métodos no código fonte

Importância do Design Software

O processo de Design pode tornar-se mais sistemático e previsível através de métodos, técnicas e padrões, todos aplicados de acordo com os princípios e heurísticas.

Importância da Gestão de Complexidade

- Programas mal concebidos são difíceis de entender e alterar;
- Quanto maior for o programa, mais visíveis são as consequências de um fraco design.

Tipos de Complexidade em Software

- Complexidades essenciais: complexidades em que são inerentes os programas;
- Complexidades acidentais/incidentais: complexidades que são os artefactos da solução.

- A soma total da complexidade numa solução de software é:
Complexidades essenciais + complexidades acidentais

→ O principal propósito do Design é controlar a complexidade

Objetivo: Gerir a complexidade essencial, evitando a introdução de complexidades acidentais adicionais

Lidar com a Complexidade do Software

Modularidade: Subdividir a solução em componentes mais pequenas, para uma fácil gestão (dividir e conquistar).

Abstração: Usar abstrações para suprimir detalhes em lugares onde são desnecessárias.

Esconder Informação: Ocultar detalhes e complexidades atrás de interfaces simples.

Herança: Componentes gerais podem ser reutilizados para definir elementos mais especificados.

Composição: Reutilização de outros componentes para construir uma nova solução.

Características do Design de Software

Não-Determinístico: Não existem dois designs que tendem a produzir o mesmo resultado.

Heurístico: Técnicas de Design tendem a confiar em heurísticas e “regras-de-ouro” ao invés de processos repetidos.

Emergente: O design final evolui/constrói-se a partir de experiência e feedback. O design é um processo iterativo e incrementável, onde um sistema complexo surge de interações simples.

Processo Geral de Design

- Compreender o problema (requisitos de software);
- Construir um modelo de “caixa-preta” da solução (especificação do sistema)
 - Especificações do sistema são tipicamente representadas com “use cases” (especificamente em POO);
- Procurar por soluções já existentes (por exemplo: arquiteturas e padrões) que cobram alguns ou todos os problemas de design de software identificados;
- Considerar a construção de protótipos;
- Documentar e rever o design;
- Iterar sobre a solução (*refactor*);
 - Evoluir o projeto até que cumpra os requisitos funcionais e maximize os requisitos não funcionais.

Inputs para o processo de Design

- Requisitos e especificações do sistema
 - Incluir qualquer restrição sobre as opções de design e implementação
- Conhecimento do domínio
 - Por exemplo, se for uma aplicação de assistência médica o designer precisará de alguns conhecimentos dos termos e conceitos da área da saúde
- Conhecimento da implementação
 - Capacidade e limitações do eventual ambiente de execução

Características Internas para um Design desejável

- **Complexidade mínima:** Manter tudo simples. Talvez não seja preciso grande generalidade
- **Acoplamento fraco:** Minimizar as dependências entre módulos (uso de funções ou classes nos programas deve ser minimizado de maneira a torna-lo simples e independente)
- **Facilidade de Manutenção:** O código será lido com mais regularidade depois de escrito
- **Extensibilidade:** Projete para hoje mas com o olho no amanhã.
Nota: Esta característica pode estar em conflito com “minimizar complexidade”. Engenharia consiste no equilíbrio de objetivos conflituosos.
- **Reutilização:** Reutilizar é uma marca registada de uma grande disciplina de engenharia
- **Portabilidade:** Funciona ou pode ser facilmente desenvolvido para funcionar noutros ambientes.
- **Magreza:** Em caso de dúvida, deixe de fora. O custo de adicionar “uma linha de código a mais” é certamente muito mais do que os minutos que demorou a ser escrito.
- **Estratificação:** Em camadas. Se todo o sistema não segue o estilo de arquitetura por camadas, os componentes individuais podem.
- **Técnicas Padrão:** Por vezes é bom ser conformista! Chato é bom. A implementação de código não é um lugar de experiências

Métodos de Software de Design



- Os métodos de design fornecem uma descrição sobre o procedimento para a obtenção de uma solução.
- A maioria dos métodos incluem:
 - Representação/Notação de problemas e formas intermédias de solução do problema (UML, pseudocódigo)
 - Procedimentos a seguir no desenvolvimento da solução
 - Heurísticas: Diretrizes e boas práticas para a tomada de decisões e avaliação de resultados intermédios e finais. Lembre-se, o design não é determinista.

Design – Formas de Representação

- Diagramas de classe para estruturas estáticas
- Diagramas de sequência para um comportamento dinâmico
- “Use Cases” são usados para criar e validar análises e projetar formas de representação
- Outros métodos UML são também úteis para a compreensão do problema e conceptualizar uma solução (diagramas de máquinas de estado, diagrama de atividade)

Métodos e Padrões

- Métodos e modelos são as principais técnicas para lidar com os desafios do design
- São úteis para:
 - Documentar e projetar um design
 - Transferência de conhecimento do design e experiência entre profissionais

Padrões

- Um padrão de design (*design pattern*) é uma solução reutilizável para um problema de design que ocorre comumente.
- Os padrões de design são adaptados para as características únicas do problema particular
- Assim, como existem níveis de design, existem níveis de Padrões de Design:
 - Estilos/Padrões de Arquitetura
 - Padrões de Design
 - Linguagens de Programação