

As melhores práticas de engenharia de software: modelar visualmente (UML) e usar arquitetura baseada em componentes

APRESENTAR OS CONCEITOS RELACIONADOS À MODELAGEM VISUAL E À ARQUITETURA BASEADA EM COMPONENTES.

O que é modelagem visual?

Modelagem visual é o uso de notações gráficas e textuais para capturar designs de software.

Uma notação, como UML, por exemplo, permite o aumento do nível de abstração, enquanto mantém uma rigorosa sintaxe e semântica, o que melhora a comunicação dentro da equipe de profissionais, durante o desenvolvimento e revisão do projeto.

Por que modelamos?

Um modelo é uma visão simplificada de um sistema. Os modelos mostram o essencial de um sistema por meio de uma perspectiva particular, ocultando os detalhes não essenciais.

Pense por exemplo no projeto de uma casa. Há vários modelos que ajudam na construção da casa, cada um deles apresentando uma visão diferente e importante do projeto. Por exemplo, modelo (ou planta) estrutural, modelo (ou planta) elétrico, modelo (ou planta) hidráulico, etc.

Modelos podem nos ajudar da seguinte maneira:

- Auxiliam na compreensão de sistemas complexos.
- Exploram e comparam projetos alternativos a um baixo custo.
- Fornecem uma base para a implementação.
- Capturam requisitos precisamente.
- Comunicam decisões claramente.

A importância dos modelos cresce proporcionalmente com a complexidade dos sistemas. Por exemplo, uma pequena aplicação que deve ser construída por uma única pessoa em poucos dias pode ser compreendida facilmente em toda sua extensão.

Já um sistema de e-commerce com dezenas de milhares de linhas de código, ou um sistema de controle de tráfego aéreo com centenas de milhares de linhas de código podem não ser facilmente compreendidos por uma única pessoa.

A construção de modelos, nesses casos, permite aos desenvolvedores focar no problema inteiro, entender como os vários componentes do sistema interagem e identificar falhas fatais.

Portanto, a modelagem é importante porque auxilia os desenvolvedores a visualizar, construir e documentar a estrutura e comportamento dos sistemas, sem acarretar a perda de complexidade.

O que significa arquitetura baseada em componentes?

Podemos definir arquitetura de um software como a organização ou estruturação dos seus componentes, interagindo por meio de interfaces. Os componentes são grupos coesos de código, em forma de código fonte ou executável, com interfaces e comportamentos bem definidos e que fornecem forte encapsulamento de seus conteúdos, podendo ser, portanto, substituíveis.

Arquiteturas baseadas em componentes tendem a reduzir o tamanho efetivo e a complexidade da solução, gerando, portanto, sistemas mais robustos e resistentes.

Ênfase na arquitetura

A arquitetura de um sistema é importante por várias razões:

- Ela permite que você obtenha o controle intelectual do projeto, auxiliando no gerenciamento de sua complexidade e na manutenção da integridade do sistema. Um sistema complexo é mais que a soma de suas partes, mais do que a sucessão de pequenas decisões táticas independentes. Deve haver uma estrutura unificadora e coerente para organizar suas partes sistematicamente. Além disso, deve haver regras precisas sobre como o sistema crescerá sem que sua complexidade "exploda" além da compreensão humana. A arquitetura fornece os meios para melhorar a comunicação e compreensão ao longo do projeto, estabelecendo um conjunto de referências e vocabulário comuns, com o qual é possível discutir os assuntos relativos ao projeto.
- Ela é uma base efetiva para o reuso em larga escala.

Por articular claramente os principais componentes e as interfaces críticas entre eles, uma arquitetura facilita o reuso – seja o reuso interno, que é a identificação de partes comuns, seja o reuso externo, que é a incorporação de componentes prontos para uso ou componentes de prateleira (off-the-shelf).

- Ela melhora a base para o gerenciamento de projetos

O desenvolvimento é particionado em um conjunto de equipes pequenas, cada uma tendo responsabilidades por uma ou várias partes do sistema.

Desenvolvimento baseado em componentes

Um componente de software pode ser definido como uma parte dele, um módulo, um pacote ou um subsistema, o qual desempenha uma função bem definida, tendo uma fronteira clara, e que pode ser integrado em uma arquitetura bem definida. Um componente é uma "realização física" de uma abstração do seu projeto.

Componentes podem vir de diferentes locais:

- Ao definir uma arquitetura modular, você identifica, isola, projeta, desenvolve e testa componentes, os quais podem ser individualmente testados e gradualmente integrados para formar um sistema completo.
- Alguns componentes podem ser desenvolvidos para serem reusados, especialmente os que fornecem soluções comuns a uma grande variedade de problemas. Esses componentes reusáveis formam a base do reuso dentro da organização, aumentando a produtividade e a qualidade no desenvolvimento de software.
- Recentemente, o advento de infraestruturas comerciais de componentes, tais como CORBA, ACTIVEX e JAVABEANS, desencadeou toda uma indústria de componentes de prateleira para vários domínios, que lhe permite comprar e integrar componentes ao invés de desenvolvê-los internamente na empresa.

Na próxima aula, continuaremos a abordar as melhores práticas de Engenharia de Software, tratando da verificação contínua da qualidade e do gerenciamento de configuração e mudanças.

Agora que você já estudou esta aula, resolva os exercícios e verifique seu conhecimento. Caso fique alguma dúvida, leve a questão ao Fórum e divida com seus colegas e professor.

EXERCÍCIOS (https://ead.uninove.br/ead/disciplinas/impressos/_g/pdsoft80_100/a10ex01_pdsoft80_100.pdf)

REFERÊNCIA

PRESSMAN, R. S. *Engenharia de software*. 7. ed. São Paulo: McGraw-Hill, 2010.

SOMMERVILLE, Ian. *Engenharia de software*. São Paulo: Addison-Wesley, 2007.

