“4+1” View Model of Software Architecture

Leonardo Rodrigues Marques - 178610

1. 4 + 1 é um modelo de visualização projetado por Philippe Kruchten em 1995 para "descrever a arquitetura de sistemas com uso intensivo de software, com base no uso de múltiplas visualizações simultâneas"[1] . Esse uso de múltiplas visões permite abordar separadamente os interesses de vários “stakeholders” de uma arquitetura: usuários finais, desenvolvedores, engenheiros de sistemas, gerentes de projetos, etc. e também manusear separadamente os requisitos funcionais e não-funcionais.

Na introdução da publicação de Philippe, ele questiona sobre os esforços dos livros e autores em querer capturar a essência de uma arquitetura de sistema em um único diagrama. Em busca de uma abordagem alternativa e mais clara, ele propõe organizar a descrição de uma arquitetura de software usando várias visualizações concorrentes, cada qual aborda um conjunto específico de interesses.[1]

O modelo de visualização de software 4+1 com o fim de tratar arquiteturas grandes e desafiadoras, é feito de cinco principais visualizações[2]:

* Visão de lógica: relacionada à funcionalidade que o sistema fornece aos usuários finais. Os diagramas UML usados para representar a visualização lógica incluem diagrama de classes, diagrama de comunicação e diagrama de sequência.
* Visão de desenvolvimento: descreve a organização estática do software em seu ambiente de desenvolvimento. Usa o diagrama de componentes ou diagrama de pacotes.
* Visão de processo: captura os aspectos de concorrência e sincronização de design. O diagrama de atividades é um exemplo abordado nesta visão.
* Visão de física: descreve o mapeamento do software no hardware e reflete seu aspecto distribuído. Os diagramas UML usados para descrever esta visão incluem o diagrama de implantação.
* Cenários ou visão de casos de uso: a descrição de uma arquitetura é ilustrada usando pequenos casos de uso. Os cenário descrevem sequências de interações entre processos e objetos, portanto são usados para identificar elementos arquitetônicos e para ilustrar e validar o design da arquitetura. Englobam os diagramas de casos de uso.

1. As visões podem ser aplicadas a Agenda Médica Online:

* Visão lógica: os diagramas de classes Funcionário, Consulta, Médico, Pessoa, Paciente abstraídos com seus métodos representam as funcionalidades que o sistema fornece ao usuário final. Por exemplo, o diagrama de sequência retrata a o processo de marcar uma consulta, uma funcionalidade oferecida ao cliente.
* Visão de desenvolvimento: não foi tratado especificamente na Agenda Online, mas é possível observar elementos nos diagramas de atividade. Para cada tipo de usuário, um login no sistema é pedido e as opções são mostradas de acordo com o propósito de cada integrante. Observe que há um desenvolvimento processual para cada ambiente: login representa a fronteira, a validação do login resume-se ao controle fornecido ao tipo de usuário e por fim, a entidade do usuário.
* Visão de processo: essa visão é representado no diagrama de atividades. É notável que uma sequência de operações pode levar ao desencadeamento de novos caminhos ou diversas operações podem resultar em único retorno. As barras são exemplo dessa sincronização: divergem-se ou convergem-se a medida que as atividades vão sendo selecionadas.
* Visão de física: não foi abordada na construção da Agenda, entretanto não é difícil descrevê-la. A visão física é prototipada na interação do software no hardware. Um cliente, para acessar sua conta, precisa de um navegador habilitado em um sistema operacional. Os dados do sistema precisam ficar armazenados em um banco de dados na nuvem e solicitações de acesso precisam ser atendidas por servidores. O aplicativo do médico deve ser compatível com arquitetura ARM(celular) ou X86(computador).
* Cenários: esse aspecto foi tratado no diagrama de casos de uso. Temos três autores que relacionam-se de forma coerente com as funções que eles desempenham na clínica. A partir dessa colocação, as visões lógica, processual, física e desenvolvimento serão arquitetadas a fim de satisfazer as divergências das necessidades de cada autor.

Referências

[1] - Kruchten, Philippe (1995, November). [Architectural Blueprints — The “4+1” View Model of Software Architecture.](http://www.cs.ubc.ca/~gregor/teaching/papers/4+1view-architecture.pdf) IEEE Software 12 (6), pp. 42-50.

[2] - Mikko Kontio (2008, July) [Architectural manifesto: Designing software architectures, Part 5](https://www.scribd.com/document/84656947/Architectural-manifesto-Designing-software-architectures-Part-5)