



Fundamentos de Arquitetura de Software

Rodrigo Oliveira Spínola

rodrigo@sqlmagazine.com.br



Doutorando em Engenharia de Sistemas e Computação (COPPE/UFRJ). Mestre em Engenharia de Software (COPPE/UFRJ, 2004). Bacharel em Ciências da Computação (UNIFACS, 2001). Colaborador da Kali Software (www.kalisoftware.com), tendo ministrado cursos na área de Qualidade de Produtos e Processos de Software, Requisitos e Desenvolvimento Orientado a Objetos. Consultor para implementação do MPS.BR. Atua como Gerente de Projeto e Analista de Requisitos em projetos de consultoria na COPPE/UFRJ. É Colaborador Engenharia de Software Magazine.

Rafael Ferreira Barcelos

rbarcelos@gmail.com



É Mestre na área de Engenharia de Software da COPPE/UFRJ, atualmente trabalha como Software Development Engineer in Test na Microsoft em Redmond/USA. Possui 5 anos de experiência em desenvolvimento de software tanto para sistemas de informação quanto para sistemas específicos, como por exemplo celular.

De que se trata o artigo:

Este artigo apresenta os fundamentos da arquitetura de software. São descritas a importância e o papel da arquitetura de software no processo de desenvolvimento. Também são identificadas as principais atividades realizadas durante o processo de especificação arquitetural.

Para que serve:

Quando tentamos solucionar um problema, é possível identificar diversas soluções que poderiam ser utilizadas visando resolvê-lo. Contudo, outros fatores como custo e eficiência influenciam na escolha da solução a ser adotada. No contexto do de-

senvolvimento de software, o mesmo pode ser observado ao se analisar os requisitos visando a construção de um software: várias soluções computacionais podem ser definidas para atender a esses requisitos, mas uma análise deve ser feita para definir a mais adequada ao contexto de desenvolvimento da aplicação. Para se representar essas soluções, a arquitetura de software é uma das abordagens que podem ser usadas.

Em que situação o tema é útil:

No entendimento dos fundamentos da arquitetura de software. Conhecimento este fundamental na elaboração da arquitetura de aplicações em projetos reais.

Quando tentamos solucionar um problema, é possível identificar diversas soluções que poderiam ser utilizadas visando resolvê-lo. Contudo, outros fatores como custo e eficiência influenciam na escolha da solução a ser adotada. No contexto do desenvolvimento de software, o mesmo pode ser observado ao se analisar os requisitos visando a construção de um software: várias soluções computacionais podem ser definidas para atender a esses requisitos, mas uma análise deve ser

feita para definir a mais adequada ao contexto de desenvolvimento da aplicação.

Para se representar essas soluções, a arquitetura de software é uma das abordagens que podem ser usadas. Com isso, para se obter a arquitetura (solução) mais adequada para atender aos requisitos do software (problema), uma avaliação dessa estrutura deve ser realizada.

A arquitetura consiste em um modelo de alto nível que possibilita um entendimento e uma análise mais fácil do software a

ser desenvolvido. O uso de arquitetura para representar soluções de software foi incentivada principalmente por duas tendências (GARLAN e PERRY, 1995; KAZMAN, 2001): (1) o reconhecimento por parte dos projetistas que o uso de abstrações facilita a visualização e o entendimento de certas propriedades do software, e (2) a exploração cada vez maior de frameworks visando diminuir o esforço de construção de produtos através da integração de partes previamente desenvolvidas.

Outra propriedade da arquitetura é a possibilidade de usá-la como ferramenta para comunicar a solução projetada aos diversos stakeholders que participam do processo de desenvolvimento do software (GARLAN, 2000). Contudo, para que essa comunicação seja possível, a arquitetura deve ser representada através de um documento, conhecido como documento arquitetural.

Para se construir a arquitetura de um software, e por consequência o documento arquitetural que a representa, os requisitos são as principais informações usadas. Durante o processo de especificação arquitetural (**Figura 1**), além dos requisitos, outras fontes de conhecimento podem ser utilizadas para definir os elementos arquiteturais e a forma como eles devem estar organizados. Entre essas fontes de conhecimento se destacam principalmente a experiência do arquiteto, o raciocínio sobre os requisitos, e os estilos e as táticas arquiteturais.

Contudo, existe uma falta de consenso na comunidade em relação tanto aos conceitos e definições básicas quanto à forma de representar uma arquitetura de software (BUSCHMANN et al., 1996; CLEMENTS et al., 2004). Portanto, na próxima seção são descritos os termos aqui adotados e seus respectivos conceitos associados. Além disso, são descritas a importância e o papel da arquitetura de software no processo de desenvolvimento, e, por fim, são identificadas as principais atividades realizadas durante o processo de especificação arquitetural.

Definição dos conceitos relacionados à arquitetura de software

Nessa seção, são definidos os termos utilizados neste trabalho, evitando ambigüidades, visto que terminologias in-

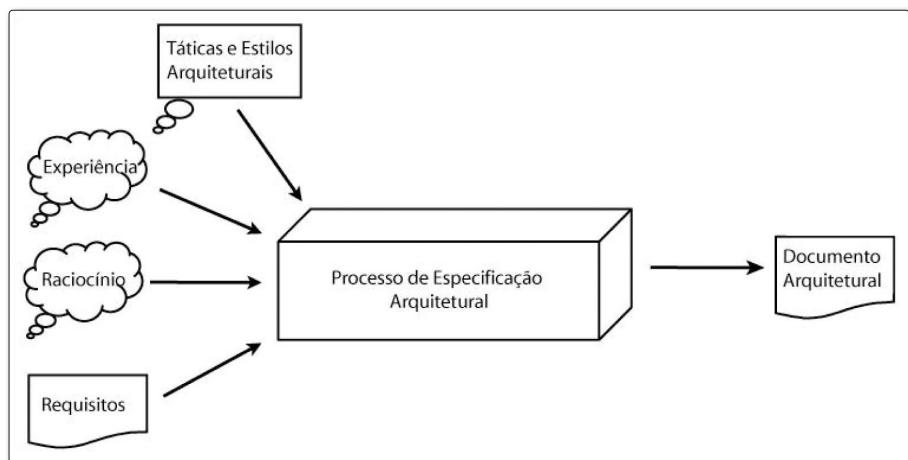


Figura 1. Elementos usados na construção de uma arquitetura.

consistentes sobre estes termos podem ser encontradas na literatura.

Arquitetura de software representa a estrutura, ou conjunto de estruturas, que compreende os elementos de software, suas propriedades externamente visíveis e seus relacionamentos (BASS et al., 2003).

Para criar essa estrutura, grande parte dos autores concorda que três tipos de elementos básicos podem ser usados (DIAS e VIEIRA, 2000):

- Elementos de software, podendo também ser chamados de módulos ou componentes, são as abstrações responsáveis por representar as entidades que implementam funcionalidades especificadas;
- Conectores, podendo ser chamados de relacionamentos ou interfaces, são as abstrações responsáveis por representar as entidades que facilitam a comunicação entre os elementos de software;
- Organização ou configuração que consiste na forma como os elementos de software e conectores estão organizados.

Além disso, essa estrutura e as entidades que a compõem devem ser representadas de uma forma que permita utilizar a arquitetura projetada para seus devidos fins, a essa representação é dado o nome de documento arquitetural. Esse documento é composto por um conjunto de modelos e informações que descrevem principalmente a estrutura do software especificado para atender aos requisitos. Para compor um documento arquitetural, podemos nos basear, por exemplo, nas recomendações descritas no padrão IEEE-1471 (IEEE, 2000).

Contudo, mesmo existindo padrões que indicam o tipo de informação que deve ser descrito em um documento arquitetural, não é definido exatamente o nível de abstração que deve ser usado na descrição dessas informações.

A arquitetura de um software começa a ser construída nos estágios iniciais de um processo de desenvolvimento de software com o objetivo de definir e visualizar a solução computacional que será implementada. Neste momento, esse artefato é conhecido como arquitetura inicial, pertence ao escopo do problema, tem como principal característica descrever a solução em um elevado nível de abstração e é utilizado por vários stakeholders como base para tomada de decisões.

Contudo, ao longo do desenvolvimento do software, a arquitetura sofre refinamentos que diminuem o nível de abstração e permitem, por exemplo, a representação dos relacionamentos entre os elementos arquiteturais e os arquivos de código fonte responsáveis por implementá-los (CLEMENTS et al., 2004). Neste momento, a arquitetura passa a pertencer ao escopo da solução e incorpora também informações relacionadas às decisões de projeto, como elementos específicos à tecnologia que será usada para implementar a solução.

O fato da arquitetura representar informações em diferentes níveis de abstração ao longo do processo de desenvolvimento é um dos motivos que leva à falta de consenso na comunidade, pois ainda não se padronizou a granularidade que deve ser usada para descrever esse artefato.

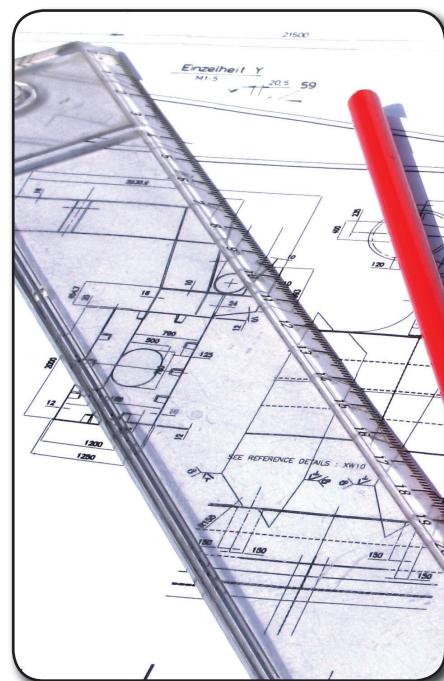
No contexto desse artigo, iremos trabalhar somente com a arquitetura inicial, ou seja, a que representa a estrutura em um elevado nível de abstração. Acreditamos que o uso de arquitetura para representar a solução em um baixo nível de abstração não é adequado devido à existência de diversos tipos de representação de projeto de baixo nível, como diagramas de classe e de seqüências, que permitem uma representação mais completa desse tipo de informação.

A partir de agora, identificaremos os papéis que a arquitetura possui no processo de desenvolvimento de software e os benefícios que podem ser obtidos ao avaliá-la.

Papel da arquitetura em um processo de desenvolvimento de software e os benefícios de sua avaliação

Ao revisar um artefato de software vários benefícios para o projeto e para a melhoria da qualidade do software podem ser obtidos. Contudo, para que essa atividade seja realizada, recursos devem ser alocados, o que pode aumentar o custo final do projeto.

Portanto, antes de realizar a revisão de um artefato, é imprescindível que a importância desse artefato dentro do processo de desenvolvimento seja identificada, permitindo definir o custo/benefício de sua revisão.



A principal motivação para avaliar a arquitetura de um software está relacionada ao seu papel dentro do processo de desenvolvimento.

Possuindo o documento arquitetural do sistema, os stakeholders podem utilizá-lo como artefato de entrada na realização de algumas atividades do processo ou então como base para tomada de decisões no contexto do projeto. Para cada stakeholder, a arquitetura do software é utilizada com diferentes propósitos (GACEK, 1995; XAVIER, 2001; CLEMENTS et al., 2004):

- Cliente. O cliente é a pessoa ou empresa que contrata uma equipe de desenvolvimento para a construção de um sistema de sua necessidade. Na fase inicial do projeto, esse stakeholder necessita de uma estimativa de certos fatores, normalmente econômicos, que podem ser obtidos após a definição da estrutura principal do software. O cliente, por exemplo, tem interesse em estimativas de custo, confiabilidade e manutenibilidade do software que podem ser obtidos principalmente através de uma análise da arquitetura. Portanto, é de extrema importância para o cliente que a arquitetura atenda os requisitos do software de forma a representar suas reais expectativas em relação ao que foi especificado.

- Gerentes. A arquitetura permite aos gerentes tomarem certas decisões de projeto por possibilitar a sumarização

das diversas características do sistema. Um gerente pode, por exemplo, usar a arquitetura como base para definir as equipes de desenvolvimento de acordo com os elementos arquiteturais que estão identificados na arquitetura e que devem ser construídos.

- Desenvolvedor. Da arquitetura de um software, o desenvolvedor busca uma especificação que descreva a solução com detalhes suficientes e que satisfaça os requisitos do cliente, mas que não seja tão restritiva a ponto de limitar a escolha das abordagens para a sua implementação. Os desenvolvedores usam a arquitetura como uma referência para a composição e o desenvolvimento dos elementos do sistema, e para a identificação e reutilização de elementos arquiteturais já construídos.

- Testadores. A arquitetura fornece, numa visão de caixa preta, informações aos testadores relacionadas ao correto comportamento dos elementos arquiteturais que se integram. Sendo assim, este artefato pode ser um dos artefatos bases utilizados durante o planejamento e execução de testes de integração e de sistema.

- Mantenedor. A descrição arquitetural do software fornece aos mantenedores uma estrutura central da aplicação que idealmente não deve ser violada. Qualquer mudança deve preservá-la, buscando, se possível, uma

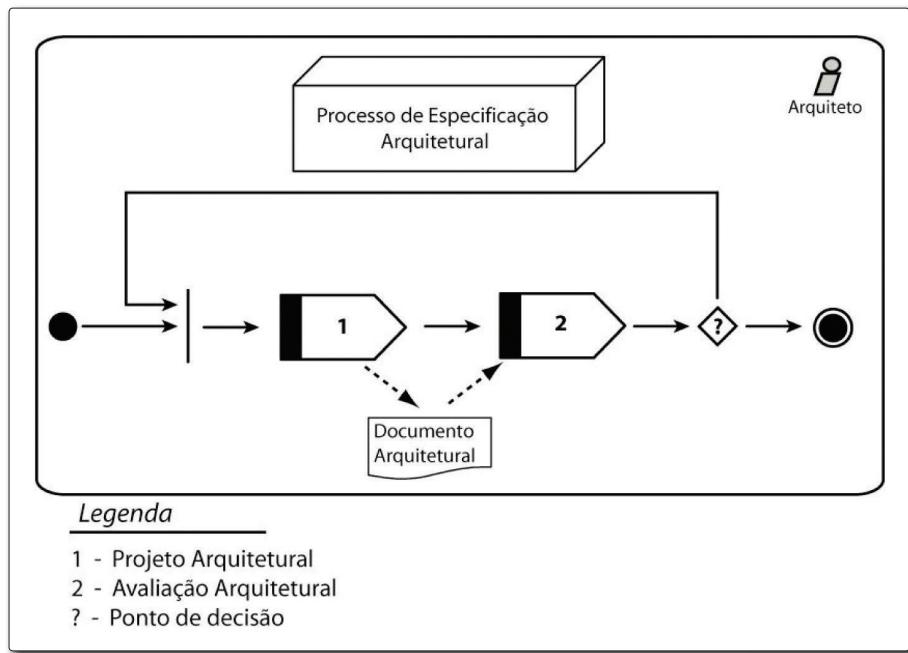


Figura 2. Processo genérico de especificação arquitetural



modificação puramente dos elementos arquiteturais e não da forma como estão organizados.

Visto como os principais stakeholders podem utilizar a arquitetura de um software, percebemos que o principal papel desse artefato é servir como instrumento para comunicar a solução proposta (GARLAN, 2000).

Sendo assim, o principal benefício em se avaliar um documento arquitetural está na diminuição das chances de um stakeholder utilizar um documento defeituoso nas atividades subsequentes do processo de desenvolvimento de software.

Contudo, para permitir uma melhor compreensão sobre como e o que deve ser avaliado em um documento arquitetural, devemos primeiro entender como esse artefato é criado.

Processo de especificação arquitetural

Existem na literatura diversas abordagens que objetivam a especificação de arquiteturas de software. Após avaliar algumas das principais abordagens (GACEK, 1995; SHAW e GARLAN, 1996; BOSCH e MOLIN, 1999; BACHMANN et al., 2000; BASS et al., 2003) pode-se perceber um processo genérico de especificação arquitetural.

Esse processo é composto principalmente pelos seguintes elementos (**Figura 2**): duas macro-atividades (projeto e avaliação arquitetural) e a tarefa de documentação da arquitetura. O que diferencia essas abordagens é principalmente a forma como cada um desses elementos são realizados.

Nesse processo, a característica comum às duas macro-atividades identificadas é a presença da tarefa de documentação responsável por criar e atualizar o documento que representa a arquitetura de software. Esse documento arquitetural é criado

durante a macro-atividade de projeto arquitetural e é responsável por registrar as decisões e os elementos arquiteturais.

Após identificarem que a solução descrita na arquitetura atende a todos os requisitos especificados, os arquitetos dão início à atividade de avaliação arquitetural que utiliza como principal artefato de entrada o documento arquitetural.

Após a avaliação, dependendo da qualidade do documento arquitetural e por consequência da arquitetura projetada, o arquiteto decide se o artefato será reavaliado visando atingir a qualidade desejada ou então se o processo de especificação arquitetural será finalizado.

A seguir, é mostrado para cada um dos elementos do processo de especificação arquitetural que tipo de informações e abordagens podem ser utilizadas para realizá-los.

Projeto Arquitetural

O projeto arquitetural consiste na atividade em que a solução computacional e, por consequência, a arquitetura do software são definidas. Durante essa atividade, o raciocínio sobre os requisitos é realizado e decisões arquiteturais são tomadas, visando identificar e organizar os elementos arquiteturais para que os requisitos especificados possam ser atendidos.

Ao se analisar como essa atividade é realizada nas principais abordagens de especificação arquitetural, observamos a importância dos requisitos de qualidade no projeto de uma arquitetura e a existência de várias abordagens que podem ser utilizadas para atendê-los.

Requisitos de Qualidade

Os requisitos de um software podem ser classificados, de forma geral, como requisitos funcionais e os não-funcionais.

Os requisitos funcionais são responsáveis por descreverem as funcionalidades que o software deve apresentar. Já os não-funcionais descrevem características que o software deve apresentar, muitas vezes podem ser enxergadas como restrições ou especialidades do produto final. Os requisitos podem ter várias subcategorias como requisitos de qualidade, requisitos legais e etc.

Dentre os diferentes tipos de requisitos, tanto funcionais quanto não-funcionais, os requisitos de qualidade são os que mais influenciam na construção da arquitetura. Isso ocorre visto que, diferente dos requisitos funcionais onde na maioria dos casos uma modificação ocasiona alterações em um conjunto específico de elementos arquiteturais, alterações em um requisito de qualidade podem implicar na total reestruturação da arquitetura (BASS et al., 2003).

Contudo, nem todos os requisitos de qualidade são relevantes a nível arquitetural, pois determinados tipos de requisitos podem ser atendidos somente durante a etapa de codificação ou disponibilização (XAVIER, 2001). Um requisito de inteligibilidade, por exemplo, só poderá ser implementado no momento da definição da interface do sistema com o usuário.

Existem diferentes taxonomias para se classificar requisitos de qualidade (ISO/IEC, 1998; BASS et al., 2003). No contexto desse artigo, adotamos a taxonomia descrita por BASS et al. (2003) visto que ela identifica os tipos de requisitos de qualidade que são relevantes a nível arquitetural, ou seja, quais os tipos de requisitos de qualidade que influenciam na construção da arquitetura de um software.

Portanto, de acordo com BASS et al. (2003), esses tipos de requisitos são:

- Desempenho: Descrevem o comportamento do sistema em relação a restrições de tempo e de recurso computacional;
- Disponibilidade: Descrevem o comportamento de determinada parte do sistema em caso de falha;
- Modificabilidade: Descrevem quais as prováveis modificações que podem acontecer no sistema e as flexibilidades que devem estar nele presentes para que essas modificações sejam facilmente realizadas;
- Segurança: Descrevem o comportamento de determinada parte do sistema em relação ao acesso de seus dados ou funcionalidades;
- Testabilidade: Descrevem o comportamento de determinada parte do sistema em relação às facilidades que elas devem fornecer para a realização de testes;
- Usabilidade: Requisitos desse tipo, em um contexto arquitetural, descrevem facilidades que o sistema deve possuir, mas que não são consideradas funcionalidades do sistema. Exemplo dessas facilidades são operações de *undo* e *redo*.

Atendendo os requisitos de qualidade

Durante o projeto de uma arquitetura, para atender aos requisitos de qualidade, as principais abordagens utilizam

diversas fontes de conhecimento, tanto tácito quanto explícito para definir quais serão os elementos arquiteturais e com estarão organizados. Um exemplo de conhecimento tácito seria a experiência do arquiteto, e em relação ao conhecimento explícito teríamos os estilos e as táticas arquiteturais.

A experiência de um arquiteto é uma característica importante para o sucesso do projeto de uma arquitetura, pois a partir de suas lições aprendidas, o arquiteto consegue facilmente identificar que elementos arquiteturais devem ser criados e como eles devem ser organizados. Mas, por ser um conhecimento tácito, é difícil de ser externalizado e utilizado por terceiros.

Por outro lado, estilos e táticas arquiteturais são conhecimentos explícitos amplamente difundidos na literatura e bastante utilizados por arquitetos de software (SHAW, 1995; BUSCHMANN et al., 1996; BASS et al., 2003).

Um estilo arquitetural, ou padrão arquitetural, consiste em um conhecimento que pode ser diretamente aplicado pelo arquiteto na identificação dos elementos arquiteturais. Isso é possível por ele ser composto por um conjunto de regras que permitem a identificação dos ti-

pos de componentes e de conectores que serão usados na composição do software levando em conta as restrições impostas (SHAW e GARLAN, 1996).

Na literatura, existe um outro conceito, chamado de padrões de projeto, que é muito semelhante ao conceito de estilos arquiteturais. Em BUSCHMANN et al. (1996), é feita a diferenciação entre padrões arquiteturais e padrões de projeto. Essa diferença encontra-se principalmente no nível de abstração onde cada um desses padrões atua. Os padrões de projeto são utilizados somente durante a fase de definição do projeto de baixo-nível, onde refinamentos são feitos nos elementos arquiteturais que formam a arquitetura, e que foram definidos com base nos padrões arquiteturais. Contudo, muitos dos conceitos presentes em padrões arquiteturais e padrões de projeto são semelhantes, mas o que os diferencia é o fato de serem utilizados em níveis de abstração diferentes.

No contexto desse artigo abordaremos somente padrões arquiteturais pois são eles que possuem os principais conceitos relevantes a nível arquitetural. Para evitar confusões, utilizaremos a denominação de estilos arquiteturais quando abordarmos esses conceitos.

Com isso, uma característica particular aos estilos arquiteturais é que o uso de um único estilo possibilita o atendimento a vários tipos de requisitos de qualidade. XAVIER (2001), por exemplo, descreve uma abordagem que, a partir dos tipos de requisitos de qualidade que devem ser atendidos pelo software, permite identificar os estilos arquiteturais mais adequados que devem ser usados na construção desse software.

Além dos estilos, outro tipo de conhecimento explícito que pode ser utilizado no projeto arquitetural são as táticas arquiteturais. Uma tática arquitetural consiste em um conhecimento mais abstrato, utilizado principalmente para auxiliar o atendimento a um tipo de requisito de qualidade. Portanto, por serem mais abstratas, essas táticas descrevem principalmente possíveis características que uma arquitetura deve apresentar para atender a um determinado tipo de requisito.

Em BASS et al. (2003), essas táticas são identificadas e categorizadas em grupos, de acordo com os atributos de qualidade que elas influenciam.

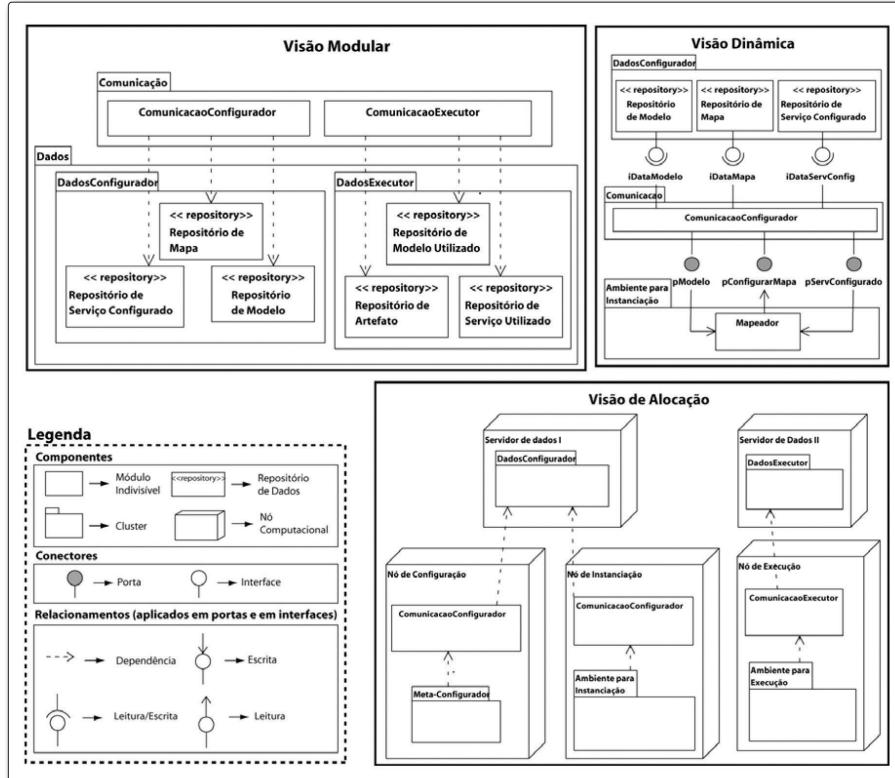


Figura 3. Exemplo de visões arquiteturais

Uma característica particular a essas táticas é que quando agrupadas e especializadas, podem ser usadas como base para a criação de estilos arquiteturais. ZHU (2004), por exemplo, realizou uma análise dos principais estilos arquiteturais por eles utilizados e identificou as táticas arquiteturais que os compõem.

Sendo assim, a partir do uso desse tipo de conhecimento, o arquiteto consegue definir a estrutura principal da arquitetura. Essa estrutura é em seguida povoada com elementos arquiteturais identificados principalmente a partir da análise dos requisitos funcionais.

Documentação Arquitetural

Uma característica única em Engenharia de Software em relação às outras áreas de engenharia é que os produtos por ela construídos não são completamente materializáveis. Diferente de um engenheiro civil que pode inspecionar, por exemplo, as partes de um prédio, um engenheiro de software não consegue inspecionar um pedaço do software em si. Para isso ele deve utilizar representações desse software (LAITENBERGER e ATKINSON, 1999).

A arquitetura é um exemplo da parte de um software que não é materializável. Durante uma inspeção, por exemplo, é o documento arquitetural que deve ser revisado, por impossibilidade de se inspecionar diretamente a arquitetura projetada. Sendo assim, durante o seu projeto, a arquitetura tem que ser documentada para que ela possa ser usada para os seus devidos fins.

A arquitetura é uma entidade complexa que não pode ser descrita de uma forma unidimensional (CLEMENTS et al., 2004). Uma forma efetiva de lidar com essa complexidade é descrevendo-a a partir de diferentes perspectivas, também conhecidas como visões arquiteturais.

Em cada visão, a forma como os elementos arquiteturais e seus relacionamentos são documentados coloca em evidência propriedades distintas do software que eles representam. De acordo com EGYED e MEDVIDOVIC (1999), ao criar uma visão arquitetural, os desenvolvedores conseguem reduzir a quantidade de informação que são obrigados a lidar em um determi-

nado momento. Portanto, essas visões representam um aspecto parcial da arquitetura que mostram propriedades específicas do software.

Na **Figura 3**, podemos identificar três visões arquiteturais usadas para descrever um conjunto de elementos arquiteturais. Independente da notação gráfica utilizada, é possível notar as diferentes propriedades que cada visão permite identificar.

Existe um grande número de visões arquiteturais propostas na literatura que propõem soluções similares para a representação de uma arquitetura (KRUCHTEN, 1995; HOFMEISTER et al., 2000; CLEMENTS et al., 2004). As principais visões são:

- Visão Modular: Esta perspectiva representa os elementos que compõem a arquitetura, responsáveis por realizar um conjunto de funcionalidades, e as dependências entre eles. Para isso, um conjunto de diagramas pode ser criado para representar através de diferentes níveis de abstração, os elementos, seus elementos internos (caso haja) e como eles se relacionam entre si.
- Visão Dinâmica: Esta perspectiva procura descrever o comportamento dos elementos arquiteturais durante a realização dos diferentes fluxos de execução que pertencem ao sistema.
- Visão de Alocação: Esta perspectiva busca representar o mapeamento das unidades de software para elementos físicos do ambiente (hardware, arquivos do sistema, equipe de desenvolvimento).
- Visão de contexto geral: Esta perspectiva tem como objetivo representar uma visão geral dos principais componentes que formam a arquitetura do software e de como ele se relaciona com os elementos externos ao seu contexto (atores e sistemas externos).

A escolha das visões a serem documentadas deve ser feita com base nas características de qualidade que se deseja por em evidência, uma vez que diferentes visões expõem características de qualidade distintas.

Para CLEMENTS et al. (2004), documentar uma arquitetura consiste em documentar as visões arquiteturais relevantes, explicar como essas visões se relacionam e como um stakeholder deve utilizar esse material.

No contexto desse artigo, é importante ressaltar algumas das recomendações definidas pelo padrão IEEE-1471, que abordam a descrição arquitetural de sistemas de software, para definir as principais informações que devem ser descritas em um documento arquitetural. Sendo assim, um documento arquitetural deve:

- Identificar os elementos arquiteturais que compõem a solução a ser construída, assim como a forma que esses elementos estão organizados;
- Descrever o papel de cada elemento dentro da arquitetura;
- Identificar como cada requisito relevante a nível arquitetural está sendo atendido através da arquitetura documentada. Essa identificação pode ser feita principalmente através do rastreamento de que requisito está sendo atendido e quais requisitos justificam a criação de determinado elemento arquitetural.
- Representar o software através de diferentes perspectivas, por exemplo, através do uso de visões arquiteturais.

Avaliação Arquitetural

A avaliação arquitetural consiste em caracterizar e avaliar os documentos arquiteturais através de métodos ou procedimentos sistemáticos (BAHSOON e EMMERICH, 2003). Essa avaliação verifica principalmente se as informações descritas no documento estão consistentes e se a arquitetura nele representada atende aos requisitos especificados para o produto.

Visto que são os requisitos de qualidade os que mais influenciam a construção de uma arquitetura, portanto, é principalmente sob a perspectiva desse tipo de requisitos que a avaliação deve ser realizada (DOBRICA e NIEMELA, 2002; BABAR et al., 2004).

A realização da atividade de avaliação é de extrema importância para a melhoria da qualidade do produto de software e para o sucesso do projeto. Esta afirmação é fortalecida se for considerado que (1) a avaliação da arquitetura impede que seus defeitos se propaguem para os demais artefatos, como diagramas de projeto e código fonte, e (2) o custo de correção desses defeitos é bem menor se for realizada durante os primeiros estágios do projeto (BOEHM, 1981).

Além dos benefícios listados anteriormente, MARANZANO et al. (2005) identificaram os seguintes benefícios, após aplicar a avaliação arquitetural em diversos projetos no contexto da empresa em que trabalha, que podem ser obtidos através dessa prática:

- Permite um melhor aproveitamento do conhecimento de seus especialistas, pois são alocados em avaliações arquiteturais que analisam arquiteturas de projetos em que não tiveram participação, utilizando assim suas experiências e conhecimentos para auxiliá-los;
- Permite um melhor gerenciamento dos fornecedores de componentes de software da empresa;
- Permite que a alta gerência tenha uma maior compreensão de problemas, principalmente de ordem técnica, que ocorrem durante a gerência dos projetos da empresa;
- Possibilita a identificação de necessidades de treinamentos ao nível de projeto ou organizacional com base em tipos de problema freqüentemente identificados durante as avaliações. Por exemplo, fornecer cursos em otimização de sistemas quando as avaliações identificarem principalmente problemas arquiteturais relacionados à característica de desempenho.

A avaliação de documentos arquiteturais é um tema que tem sido bastante discutido no contexto de vários grupos de pesquisa, como no grupo do Software Engineering Institute (SEI) (KAZMAN et al., 1994; CLEMENTS et al., 2002), por exemplo.

Conclusão

Ao longo deste artigo foram descritos os principais conceitos em relação à arquitetura de software, dando ên-

fase principalmente nas atividades que estão relacionadas ao seu processo de especificação.

Através da análise desses conceitos e processos, foi possível identificar (1) a importância da arquitetura dentro do processo de desenvolvimento de software, (2) como esse artefato é construído e principalmente (3) que informações devem estar representadas nesse artefato e que devem ser analisadas durante o processo de avaliação para que se determine a corretude do documento arquitetural. ●



Dê seu feedback sobre esta edição!

A Engenharia de Software Magazine tem que ser feita ao seu gosto.

Para isso, precisamos saber o que você, leitor, acha da revista!

Dê seu voto sobre este artigo, através do link:

www.devmedia.com.br/esmag/feedback



Referências

- BABAR, M.A., ZHU, L., JEFFERY, R., 2004, "A framework for classifying and comparing software architecture evaluation methods". In: Proceedings of the Australian Software Engineering Conference, pp. 309-318, Melbourne, Australia, April.
- BACHMANN, F., BASS, L., CHASTEK, G., DONOHOE, P., PERUZZI, F., 2000, The Architecture Based Design Method, CMU/SEI, Relatório Técnico, CMU/SEI-2000-TR-001.
- BAHSOON, R., EMMERICH, W., 2003, "Evaluating software architectures: development, stability, and evolution". In: Book of Abstracts of the ACS/IEEE International Conference on Computer Systems and Applications, pp. 47, Tunis, Tunisia, July.
- BASS, L., CLEMENTS, P., KAZMAN, R., 2003, Software Architecture in Practice, Second Edition, Addison Wesley.
- BOEHM, B.W., 1981, Software Engineering Economics, Prentice-Hall.
- BOSCH, J., MOLIN, P., 1999, "Software Architecture Design: Evaluation and Transformation". In: Proceedings of the IEEE Engineering of Computer Based Systems Symposium (ECBS'99), pp. 4, Nashville, TN, USA, March.
- BUSCHMANN, F., MEUNIER, R., ROHNERT, H., SOMMERLAD, P., STAL, M., 1996, Pattern-Oriented Software Architecture: A System of Patterns, Jon Wiley and Sons.
- CLEMENTS, P., BACHMANN, F., BASS, L., GARLAN, D., IVERS, J., LITTLE, R., NORD, R., STAFFORD, J., 2004, Documenting Software Architectures, Addison-Wesley.
- DIAS, M.S., VIEIRA, M.E.R., 2000, "Software architecture analysis based on statechart semantics". In: International Workshop on Software Specification and Design, pp. 133-137, Washington, DC, USA.
- DOBRIĆA, L., NIEMELÄ, E., 2002, "A survey on software architecture analysis methods", IEEE Transactions on Software Engineering, v. 28, n. 7, pp. 638-653.
- EGYED, A., MEDVIDOVIC, N., 1999, "Extending Architectural Representation in UML with View Integration". In: Proceedings of the 2nd International Conference on the Unified Modeling Language. Beyond the Standard (UML'99), v. 1723, pp. 2-16, Fort Collins, USA, October.
- GACEK, C., 1995, On the Definition of Software System Architecture, University of Southern California, Relatório Técnico, USC/CSE-95-TR-500.
- GARLAN, D., 2000, "Software architecture: a roadmap". In: Proceedings of The Conference on The Future of Software Engineering, pp. 91-101.
- GARLAN, D., PERRY, D., 1995, "Introduction to the Special Issue on Software Architecture". In: IEEE Transactions on Software Engineering, v. 21, April.
- HOFMEISTER, C., NORD, R.L., SONI, D., 2000, A study on agreement between participants in an architecture assessment, Addison-Wesley.
- IEEE, 2000, "IEEE Recommended Practice For Architectural Description Of Software-Intensive Systems - IEEE Standard 1471-2000", Institute of Electrical and Electronics Engineers.
- ISO/IEC, 1998, "International Technology - Software Product Evaluation - ISO/IEC 9126 Part 1: Quality Model".
- KAZMAN, R., 2001, "Handbook of Software Engineering and Knowledge Engineering". In: CHANG, S.K. (eds), World Scientific Publishing.
- KAZMAN, R., BASS, L., ABOWD, G., WEBB, M., 1994, "SAAM: a method for analyzing the properties of software architectures". In: Proceedings of the International conference on Software Engineering (ICSE), pp. 81-90.
- KRUCHTEN, P., 1995, "Architectural Blueprints - The 4+1 View Model of Software Architecture". In: IEEE Software, v. 12, pp. 42-50, November.
- LAITENBERGER, O., ATKINSON, C., 1999, "Generalizing Perspective-based Inspection to handle Object-Oriented Development Artifacts". In: Proceedings of the International conference on Software Engineering (ICSE).
- MARANZANO, J.F., ROZYPAL, S.A., ZIMMERMAN, G.H., WARNKEN, G.W., WIRTH, P.E., WEISS, D.M., 2005, "Architecture reviews: practice and experience", IEEE Software, v. 22, n. 2, pp. 34-43.
- SHAW, M., 1995, "Some Patterns for Software Architectures".
- SHAW, M., GARLAN, D., 1996, Software Architecture - Perspectives on an Emerging Discipline, Prentice Hall.
- XAVIER, J.R., 2001, Criação e Instanciação de Arquiteturas de Software Específicas de Domínio no Contexto de uma Infra-estrutura de Reutilização, Dissertação de Mestrado, Programa de Engenharia de Sistemas e Computação - COPPE/UFRJ.
- ZHU, L., BABAR, M.A., JEFFERY, R., 2004, "Mining patterns to support software architecture evaluation". In: Proceedings of the 4th Working IEEE/IFIP Conference on Software Architecture, pp. 25 - 34, June.

