

A **arquitetura de software** de um sistema consiste na definição dos componentes de software, suas propriedades externas, e seus relacionamentos com outros softwares. O termo também se refere à documentação da arquitetura de software do sistema. A documentação da arquitetura do software facilita: a comunicação entre os stakeholders, registra as decisões iniciais acerca do projeto de alto-nível, e permite o reuso do projeto dos componentes e padrões entre projetos.

### Índice

- 1                    Introdução
- 2                    História
- 3                    Descrevendo arquiteturas
- 3.1                Linguagem de descrição de arquitetura
- 3.2                Visões
- 3.3                Padrões de arquitetura
  
- 4                    Exemplos de arquitetura
- 5                    Notas e Referências
- 6                    Ver também
- 7                    Ligações externas

### Introdução

O campo da ciência da computação tem lidado com problemas associados, como a complexidade da informação, desde sua criação.<sup>[1]</sup> Os primeiros problemas de complexidade foram resolvidos pelos desenvolvedores através da escolha da estrutura de dados, do desenvolvimento de algoritmos e pela aplicação de conceitos de separação de escopos.

Embora o termo *arquitetura de software* seja relativamente novo na indústria, os princípios fundamentais deste campo vem sendo aplicados esporadicamente pela engenharia de software desde o início dos anos 80. As primeiras tentativas de capturar e explicar a arquitetura de software do sistema foram imprecisas e desorganizadas – frequentemente caracterizadas por um conjunto de diagramas.

[2]

Durante o decorrer da década de 90 houve um esforço concentrado para definir e codificar os aspectos fundamentais desta disciplina. Inicialmente um conjunto de padrões de projeto, estilo, melhores práticas, descrição de linguagens, e lógica formal foram desenvolvidas durante este período.

A disciplina de arquitetura de software é centrada na idéia da redução da complexidade através da abstração e separação de interesses. O glossário do site oficial SOFTWARE ENGINEERING INSTITUTE (Instituto de Engenharia de Software) <sup>[3]</sup> descreve que arquitetura de software é a estrutura ou estruturas de um sistema, com todos os elementos de software vendo e tendo suas propriedades vistas por todos os outros elementos e relacionamentos.

Sendo a arquitetura de sistema uma disciplina em maturação, sem regras claras, a ação do arquiteto é ainda uma composição de arte e ciência. Os aspectos de *arte* da arquitetura de software são devido ao fato que os sistemas de software comerciais suportam alguns aspectos de um negócio ou missão. Como o direcionamento de negócio chave para o suporte sistemas são descrito nos cenários como requisitos não-funcionais de sistema, também conhecidos como atributos de qualidade, que determinam como um sistema ira se comportar.

[4]

Cada sistema é único devido a natureza do negócio que ele suporta, tal que o nível dos atributos de qualidade exigidos de um sistema como compatibilidade, extensibilidade, confiabilidade, manutenibilidade, disponibilidade, segurança, usabilidade, dentre outros – irão variar com cada aplicação.

[4]

Para trazer a perspectiva do usuário para dentro da arquitetura de software, pode-se dizer que a arquitetura de software dá a direção dos passos que serão tomados e as tarefas envolvidas em cada área de especialidade e interesse do usuário, por exemplo, *stakeholders* de sistemas de software, os desenvolvedores de software, o grupo de suporte ao software do sistema operacional, aos testadores e usuários de negocio final. Neste sentido, a arquitetura de software se torna a ligação das múltiplas perspectivas que um sistema traz nele embutido. O fato de que estas várias perspectivas diferentes possam ser postas juntas em uma arquitetura de software padrão justifica e valida a necessidade de criação da arquitetura de software antes do desenvolvimento do software para que projeto alcance a maturidade.

## História

A origem da arquitetura de software como um conceito foi primeiramente identificado no trabalho de pesquisa de Edsger Dijkstra em 1968 e David Parnas no início de 1970. Estes cientistas enfatizaram a importância das estruturas de um sistema de software e a criticidade da identificação da sua estrutura.<sup>[5]</sup> O estudo deste campo aumentou de popularidade desde o início de 1990 com os trabalhos de pesquisa concentrando-se nos padrões de estilo de arquitetura, linguagens de descrição de arquitetura, documentação de arquitetura, e métodos formais.

<sup>[6]</sup> Muitas instituições de pesquisa tais como a Carnegie Mellon University e a University of California, Irvine estavam realizando muitas pesquisas no campo da arquitetura de software. Mary Shaw e David Garlan da Carnegie

Mellon escreveram um livro intitulado

*Software*

*Architecture: Perspectives on an Emerging Discipline*

em 1996, o qual trazia a tona conceitos da arquitetura de software, tais como componentes, conexões, estilos, etc. Os esforços do UCI's (Institute for Software Research) na pesquisa da arquitetura de software foram inicialmente direcionado para os estilos de arquitetura, descrições de linguagens arquitetura, e arquiteturas dinâmicas.

*ANSI/IEEE 1471-2000: Recommended Practice for Architecture Description of Software-Intensive Systems[1]* foi a primeira norma padrão na área de arquitetura de software, e foi recentemente adotada pelo ISO como *ISO/IEC DIS 25961*.

## **Descrevendo arquiteturas**

### **Linguagem de descrição de arquitetura**

As Linguagens de descrição de arquitetura (LDAs) são usadas para descrever a arquitetura de software. Vários diferentes LDAs foram desenvolvidas por diferentes organizações, incluindo Wright (desenvolvido por Carnegie Mellon), Acme (desenvolvido por Carnegie Mellon), xADL (desenvolvido por UCI), Darwin (desenvolvido por Imperial College London), DAOP-ADL (desenvolvido pela University of Málaga). Elementos comuns de uma LDA são componente, conexão e configuração.

### **Visões**

A arquitetura de software é normalmente organizada em visões<sup>[7]</sup>, as quais são análogas aos diferentes tipos de plantas utilizadas no estabelecimento da arquitetura. Na Ontologia estabelecida pela ANSI/IEEE 1471-2000,

*visões*

são instancias de

*pontos de vista*

, onde um ponto de vista existe para descrever a arquitetura na perspectiva de um conjunto de stakeholders e seus consortes.

Algumas possíveis visões são:

- Visão funcional/lógica
- Visão de código.
- Visão de desenvolvimento/estrutural
- Visão de concorrência/processo/*thread*

- Visão física/evolutiva
- Visão de ação do usuário/*feedback*

Várias linguagens para descrição da arquitetura de software foram inventadas, mas nenhum consenso foi ainda alcançado em relação a qual conjunto de símbolos ou sistema representação deve ser adotado. Alguns acreditam que a UML ira estabelecer um padrão para *representação de arquitetura de software*. Outros acreditam que os desenvolvimentos efetivos de software devem contar com a compreensão única das restrições de cada problema, e notações tão universais são condenadas a um final infeliz porque cada uma prove um notação diferenciada que necessariamente torna a notação inútil ou perigosa para alguns conjuntos de tarefas. Eles apontam a proliferação de linguagens de programação e a sucessão de tentativas falhas para impor uma simples 'linguagem universal' na programação, como uma prova da tendência do software para a diversidade e não para padrões.

### Padrões de arquitetura

- DODAF [2]
- MODAF [3]
- TOGAF [4]
- Zachman framework [5]
- Federal Enterprise Architecture [6]

### Exemplos de arquitetura

Há muitas formas comuns de projetar módulos de software de computador e suas comunicações, entre elas:

- Cliente-Servidor
- Computação distribuída
- P2P
- Quadro Negro
- Criação implícita
- Pipes e filtros
- Plugin
- Aplicação monolítica
- Modelo em três camadas
- Análise de sistema estruturada (baseada em módulos, mas usualmente monolíticas em dentro dos módulos)
  - Arquitetura orientada a serviço
  - Arquitetura orientada a busca

## Notas e Referências

1. ↑ University of Waterloo (2006). *A Very Brief History of Computer Science*. Página visitada em 2006-09-23.
2. ↑ IEEE Transactions on Software Engineering (2006). *Introduction to the Special Issue on Software Architecture*. Página visitada em 2006-09-23.
3. ↑ SEI (2010). *Glossary?*. Página visitada em 2010-01-31.
4. ↑ <sup>a</sup> <sup>b</sup> SoftwareArchitectures.com (2006). *Intro to Software Quality Attributes*. Página visitada em 2006-09-23.
5. ↑ SEI (2006). *Origins of Software Architecture Study*. Página visitada em 2006-09-25.
6. ↑ Garlan & Shaw (2006). *An Introduction to Software Architecture*. Página visitada em 2006-09-25.
7. ↑ Clements, Paul. *Documenting Software Architectures: Views and Beyond*. 2 ed. Boston: Addison-Wesley, 2003. pp. 13-15 p. ISBN 0201703726

## Ver também

- Padrões de projeto de software
- Antipadrões de software
- Modelagens de dados padrões
- Matriz de estrutura de dependência
- Arquitetura de negócio
- Arquitetura de dados
- Arquitetura de processo
- Arquiteto de software

## Ligações externas

- Software architecture definitions at Carnegie Mellon University Software Engineering Institute
- Software architecture vs. software design
- Worldwide Institute of Software Architects
- Grady Booch's Handbook of Software Architecture project
- SoftwareArchitectures.com Independent resource of information on the discipline
- International Association of Software Architects
- Microsoft Architecture Journal
- Pangea - Professional and Academic Network to the Growing and Evolution of Architecture (Portuguese)
- Blog da OCTO Technology Brasil, empresa de consultoria em Arquitetura de Sistemas de Informação

Obtida de  
"http://pt.wikipedia.org/w/index.php?title=Arquitetura\_de\_software&oldid=27486721" Categoria:

- Engenharia de software

fonte.wikipedia