

Arquitetura de software

Origem: Wikipédia, a enciclopédia livre.

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 reúso do projeto dos componentes e padrões entre projetos.

Índice

Introdução
História
Descrevendo arquiteturas
Linguagem de descrição de arquitetura
Visões
Padrões de arquitetura
Exemplos de arquitetura de software
Notas e Referências
Ver também
Ligações externas

Introdução

O campo da ciência da computação tem lidado com problemas associados, como a complexidade da informação, desde sua criação.^[1] 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 vêm 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 – freqüentemente 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 ideia da redução da complexidade através da abstração e separação de interesses. O glossário do site oficial SOFTWARE ENGINEERING INSTITUTE (Instituto de Engenharia de Software) ^[3] 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 de software é ainda uma composição de arte e ciência. Os aspectos de *arte* da arquitetura de software são devidos ao fato que os sistemas de software comerciais suportam alguns aspectos de um negócio ou missão. Assim como o direcionamento de negócio chave para o suporte os sistemas são descritos nos cenários como requisitos não-funcionais de sistema, também conhecidos como atributos de qualidade, que determinam como um sistema irá se comportar.^[4] Cada sistema é único devido à natureza do negócio que ele suporta, tal que o nível dos atributos de qualidade exigidos de um sistema como compatibilidade, extensibilidade, confiabilidade, manutenabilidade, disponibilidade, segurança, usabilidade, dentre outros – irão variar para cada aplicação sendo desenvolvida.^[4]

Para trazer a perspectiva do usuário para dentro da arquitetura de software, pode-se dizer que essa disciplina dá a direção dos passos que serão tomados e as tarefas envolvidas em cada área de especialidade e interesse do usuário, por exemplo, os *stakeholders* de sistemas de software, os desenvolvedores de software, o grupo de suporte ao software do sistema operacional, os testadores e os usuários de negócio 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 o 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.^[5] 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 de software, linguagens de descrição de arquitetura de software, documentação de arquitetura de software, e métodos formais.^[6] 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 direcionados para os estilos de arquitetura, descrições de linguagens de arquitetura, e arquiteturas dinâmicas.

ANSI/IEEE 1471-2000: Recommended Practice for Architecture Description of Software-Intensive Systems^[1] (http://en.wikipedia.org/wiki/IEEE_1471) 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. Várias LDAs distintas 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,^[7] as quais são análogas aos diferentes tipos de plantas utilizadas no estabelecimento da arquitetura. Na Ontologia estabelecida pela ANSI/IEEE 1471-2000, *visões* são instâncias de *pontos de vista*, onde cada 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/retorno

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 de representação deve ser adotado. Alguns acreditam que a UML irá 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 provê uma 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 os padrões.

Padrões de arquitetura

- DODAF [2] (http://en.wikipedia.org/wiki/Department_of_Defense_Architecture_Framework)
- MODAF [3] (<http://en.wikipedia.org/wiki/MODAF>)
- TOGAF [4] (<http://en.wikipedia.org/wiki/TOGAF>)
- Zachman framework [5] (http://en.wikipedia.org/wiki/Zachman_framework)
- Federal Enterprise Architecture [6] (http://en.wikipedia.org/wiki/Federal_Enterprise_Architecture)

Exemplos de arquitetura de software

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
- MVC

Notas e Referências

1. University of Waterloo (2006). «A Very Brief History of Computer Science» (<http://www.cs.uwaterloo.ca/~shallit/Courses/134/history.html>). Consultado em 23 de setembro de 2006
2. IEEE Transactions on Software Engineering (2006). «Introduction to the Special Issue on Software Architecture» (<http://csdl2.computer.org/persagen/DLAbsToc.jsp?resourcePath=/dl/trans/ts/&toc=comp/trans/ts/1995/04/e4toc.xml&DOI=10.1109/TSE.1995.10003>). Consultado em 23 de setembro de 2006
3. SEI (2010). «Glossary?» (<http://www.sei.cmu.edu/architecture/start/glossary>). Consultado em 31 de janeiro de 2010
4. SoftwareArchitectures.com (2006). «Intro to Software Quality Attributes» (<http://www.softwarearchitectures.com/one/Designing+Architecture/78.aspx>). Consultado em 23 de setembro de 2006
5. SEI (2006). «Origins of Software Architecture Study» (<http://www.sei.cmu.edu/architecture/roots.html>). Consultado em 25 de setembro de 2006
6. Garlan & Shaw (2006). «An Introduction to Software Architecture» (http://www.cs.cmu.edu/afs/cs/project/able/ftp/intro_softarch/intro_softarch.pdf) (PDF). Consultado em 25 de setembro de 2006
7. Clements, Paul; Felix Bachmann; Len Bass; David Garlan; James Ivers; Reed Little; Robert Nord; Judith Stafford (2003). *Documenting Software Architectures: Views and Beyond* 2 ed. Boston: Addison-Wesley. pp. 13–15. 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 \(http://www.sei.cmu.edu/architecture/definitions.html\)](http://www.sei.cmu.edu/architecture/definitions.html)
 - [Software architecture vs. software design \(http://eden-study.org/articles/2003/icse03.pdf\)](http://eden-study.org/articles/2003/icse03.pdf)
 - [Worldwide Institute of Software Architects \(http://www.wwisa.org/\)](http://www.wwisa.org/)
 - [Grady Booch's \(http://www.booch.com/architecture/index.jsp\)Handbook of Software Architecture](http://www.booch.com/architecture/index.jsp) project
 - [SoftwareArchitectures.com \(http://www.SoftwareArchitectures.com\)](http://www.SoftwareArchitectures.com)Independent resource of information on the discipline
 - [International Association of Software Architects \(http://www.iasahome.org/\)](http://www.iasahome.org/)
 - [Microsoft Architecture Journal \(http://www.architecturejournal.net/\)](http://www.architecturejournal.net/)
 - [Pangea - Professional and Academic Network to the Growing and Evolution of Architecture \(Portuguese\) \(http://pangeanet.org/\)](http://pangeanet.org/)
 - [Blog da OCTO Technology Brasil, empresa de consultoria em Arquitetura de Sistemas de Informação \(http://blog.octo.com/pt-br/\)](http://blog.octo.com/pt-br/)
-

Obtida de "https://pt.wikipedia.org/w/index.php?title=Arquitetura_de_software&oldid=50440529"

Esta página foi editada pela última vez às 18h35min de 10 de novembro de 2017.

Este texto é disponibilizado nos termos da licença [Atribuição-CompartilhaIgual 3.0 Não Adaptada \(CC BY-SA 3.0\)](#) da [Creative Commons](#); pode estar sujeito a condições adicionais. Para mais detalhes, consulte as [condições de utilização](#).