

O artigo apresenta um panorama abrangente da evolução da arquitetura de software como disciplina e lança um olhar crítico sobre os caminhos futuros da área. O texto é relevante não apenas pela revisão histórica que oferece, mas principalmente por antecipar discussões que se tornaram centrais nas duas últimas décadas, como a padronização de arquiteturas, a integração de sistemas heterogêneos e o avanço da computação pervasiva.

Logo no início, Garlan ressalta que a arquitetura de software consolidou-se como uma das áreas mais importantes da engenharia de software, pois é responsável por organizar sistemas complexos em componentes que interagem entre si. Essa organização permite garantir atributos de qualidade como desempenho, confiabilidade, escalabilidade e interoperabilidade. De maneira simples e objetiva, o autor reforça que uma boa arquitetura pode ser a chave para o sucesso de um sistema, enquanto uma arquitetura mal concebida pode comprometer todo o projeto. Nesse sentido, a arquitetura atua como ponte entre requisitos e implementação, assumindo papéis fundamentais na compreensão, construção, análise, manutenção e gestão de sistemas de software.

Ao revisitar o passado, o autor lembra que, até a década de 1990, a arquitetura de software era frequentemente tratada de maneira informal, muitas vezes restrita a diagramas de “caixa e linha” que raramente eram mantidos ou analisados com rigor. Os arquitetos, quando existiam, desenvolviam suas habilidades de forma empírica, sem metodologias consolidadas. Esse cenário começou a mudar à medida que a complexidade dos sistemas cresceu e se tornou evidente a necessidade de princípios mais sólidos, padrões de projeto e linguagens que apoiassem a descrição arquitetural. Assim, surgiram vocabulários técnicos, estilos arquiteturais e ferramentas de apoio, estabelecendo as bases para que a área se afirmasse como disciplina.

No presente da época em que o artigo foi escrito, Garlan observa uma mudança significativa. A arquitetura de software passa a ser reconhecida formalmente, com cargos específicos em empresas, revisões arquiteturais e práticas cada vez mais estruturadas. Entre os avanços mais importantes, ele destaca o desenvolvimento das Architecture Description Languages (ADLs), que permitem representar e analisar arquiteturas de forma formal; a disseminação de linhas de produto (product lines), que favorecem a reutilização e a padronização em diferentes sistemas; e a consolidação de estilos arquiteturais bem documentados, como pipe-and-filter, client-server e

event-based. Tais avanços ajudaram a transformar a arquitetura em um campo mais disciplinado, mas também revelaram novos problemas, especialmente no que diz respeito à integração de sistemas heterogêneos, vindos de diferentes fornecedores e com pressupostos incompatíveis.

O ponto mais instigante do artigo está em sua projeção para o futuro. Garlan identifica três tendências tecnológicas que moldariam a evolução da arquitetura de software: a mudança na relação entre construir (build) e comprar (buy), a ascensão da computação em rede (network-centric computing) e o avanço da computação pervasiva (pervasive computing). A primeira revela um mundo em que empresas passam a se comportar mais como integradoras do que como desenvolvedoras, exigindo padrões arquiteturais de maior nível para garantir compatibilidade. A segunda reflete a transição do modelo centrado no computador pessoal para um modelo centrado na rede, em que sistemas distribuídos, serviços dinâmicos e coalizões de software autônomas se tornam comuns. Já a terceira aponta para a multiplicação de dispositivos inteligentes conectados, demandando arquiteturas capazes de lidar com recursos limitados, reconfiguração dinâmica e mobilidade dos usuários.

Na conclusão, Garlan reforça que, embora a arquitetura de software tenha amadurecido consideravelmente, ela ainda está em evolução. Consolidar práticas já existentes é importante, mas os desafios impostos por novos cenários tecnológicos exigem inovação constante. O texto termina com uma mensagem clara: a arquitetura de software é um campo vivo, que precisa acompanhar e até se antecipar às transformações do mundo digital.