

Aluno: Bernardo de Resende Marcelino  
Disciplina: Projeto de software  
Professor: João Paulo Aramuni

### Resenha Crítica "Microservices - Martin Fowler"

o Artigo "Microservices" de Martin Fowler busca compreender a arquitetura e definir o que são microsserviços, quando devem ser usados, comparações com modelos monólitos, que tipo de problemas podem ser resolvidos através desse modelo arquitetural e quais suas vantagens e desvantagens durante o desenvolvimento e o ciclo de vida do software

Martin busca compreender as motivações para se adotar esse tipo de arquitetura comparando com monólitos, código sem modularização, onde o autor identifica problemas como dificuldade de fazer deploys, onde pequenas mudanças necessitam de fazer deploy na aplicação como um todo, dificultando o processo de alteração no código, além de abrir espaço para pequenas vulnerabilidades, que por estarem em um sistema monólito, podem comprometer o funcionamento do sistema como um todo.

Além disso, são observados problemas de escalabilidade, pelos mesmos motivos de deploys, que nesse modelo, podem necessitar alterações de escala global na aplicação, mesmo que nem todos recursos necessitem de ter sua escalabilidade aumentada

Diante desse cenário, o artigo atribui a essas dificuldades o surgimento de uma nova arquitetura nos últimos anos, os microsserviços, que permitem deploy individual, escalabilidade de cada serviço de forma separada, produção de serviços em diferentes linguagens, dando maior flexibilidade de desenvolvimento diante das necessidades do software além de permitir que diferentes times trabalhem em serviços separadamente, trazendo maior modularidade e independência na construção e manutenção do software, aumentando ainda mais a compreensão de software como produto e não como projeto, onde times de desenvolvimento se tornam diretamente responsáveis por serviços individuais envolvidos diretamente em todas etapas da implementação, testes e manutenção deste módulo.

No entanto, Martin destaca também as dificuldades de manter esse tipo de arquitetura, como overhead de comunicação, dificuldades de se manter padrão de dados, formas de armazenamento, e como compartilhar esses dados entre os componentes sem inconsistências pode ser desafiador, como a utilização de requests HTTP/REST que pode ser mais simples e eficiente que pipes de ESB para comunicação com roteamento complexo e os serviços passam a depender menos da infraestrutura nesse aspecto.

Dessa forma, o artigo traz aspectos positivos e negativos importantes sobre arquitetura moderna de software, cuidados importantes ao implementar e por que pode ser uma boa alternativa trocar um grande monólito por microsserviços, no entanto, o autor se preocupa em ressaltar como nem sempre microsserviços são a resposta para todos os problemas, tendo suas limitações e cuidados específicos exigindo análise do cenário individual do software e produto que está sendo desenvolvido.