

Resenha do artigo “Big Ball of Mud” – Felipe Giannetti Fontenelle

Introdução

O artigo Big Ball of Mud, escrito por Brian Foote e Joseph Yoder, aborda um tema muito comum na área de Engenharia de Software: a diferença entre a arquitetura ideal ensinada na teoria e a arquitetura que realmente existe na maioria dos sistemas usados no dia a dia. Os autores chamam essa arquitetura predominante de Big Ball of Mud, que pode ser entendida como um sistema grande, confuso e sem uma estrutura bem definida.

O texto é relevante porque foge de uma visão excessivamente idealizada do desenvolvimento de software e tenta explicar, de forma realista, por que tantos sistemas acabam se tornando difíceis de entender e manter. Em vez de apenas criticar, o artigo busca compreender as causas desse fenômeno.

O que é Big Ball of Mud

Segundo os autores, um Big Ball of Mud é um sistema que cresce de forma desorganizada, sem um planejamento arquitetural claro. Ele costuma apresentar código duplicado, alto acoplamento entre partes do sistema, uso excessivo de variáveis globais e funções que fazem várias coisas ao mesmo tempo. Como consequência, o sistema se torna difícil de manter, evoluir e compreender.

Apesar dessas características negativas, o artigo destaca que esse tipo de sistema é extremamente comum porque ele funciona. Muitas vezes, mesmo sendo mal estruturado, o software atende às necessidades imediatas dos usuários e do negócio, o que faz com que ele continue sendo usado e ampliado.

Por que sistemas se tornam um Big Ball of Mud

O artigo aponta várias razões pelas quais sistemas acabam assumindo essa forma. Uma das principais é a pressão por tempo. Em muitos projetos, há prazos curtos e a prioridade é entregar algo que funcione rapidamente. Com isso, decisões arquiteturais acabam sendo deixadas de lado.

Outro fator importante é o custo. Investir em uma boa arquitetura exige tempo, profissionais experientes e planejamento, o que nem sempre é visto como prioridade pelas organizações. Além disso, no início de um projeto, a equipe ainda não entende completamente o domínio do problema, o que dificulta a criação de uma arquitetura sólida desde o começo.

A mudança constante de requisitos também contribui para esse cenário. Novas necessidades surgem, clientes mudam suas demandas e o sistema precisa se adaptar rapidamente. Nessas situações, a solução mais comum é fazer ajustes pontuais, mesmo que isso prejudique a estrutura geral do sistema.

Padrões relacionados ao Big Ball of Mud

Foote e Yoder apresentam alguns padrões que ajudam a explicar como o Big Ball of Mud surge e se mantém. Um deles é o Throwaway Code, que ocorre quando um código criado como protótipo ou solução temporária acaba sendo reaproveitado como sistema definitivo. Isso é bastante comum em projetos reais, especialmente quando o protótipo funciona bem.

Outro padrão é o Piecemeal Growth, que representa o crescimento incremental do sistema. Pequenas mudanças são feitas ao longo do tempo, sem uma revisão global da arquitetura. Embora esse tipo de crescimento permita que o sistema evolua rapidamente, ele pode levar à degradação da estrutura.

Há também o padrão Keep It Working, que prioriza manter o sistema sempre funcionando, evitando grandes mudanças que possam causar falhas. Embora essa estratégia seja segura no curto prazo, ela pode dificultar melhorias arquiteturais no longo prazo.

Relação com experiências profissionais

Os conceitos apresentados no artigo podem ser facilmente observados em ambientes de trabalho reais. Em minha experiência com desenvolvimento de software, já vi sistemas que começaram como soluções simples para resolver um problema específico e, com o tempo, passaram a receber novas funcionalidades sem uma reorganização adequada.

Normalmente, o foco era entregar rapidamente o que era pedido, sem tempo para refatorar ou repensar a arquitetura. Como o sistema continuava funcionando, as melhorias estruturais eram constantemente adiadas. Com o passar do tempo, o código se tornava cada vez mais difícil de entender, caracterizando um Big Ball of Mud. Um exemplo de experiência é o meu trabalho interdisciplinar 2, desenvolvido no segundo semestre do curso de Engenharia de Software.

Esse tipo de situação é comum em empresas onde há poucos desenvolvedores, prazos apertados e mudanças frequentes de prioridade. Nesses casos, a teoria da arquitetura bem planejada entra em conflito direto com as necessidades práticas do dia a dia.

Conclusão

O artigo Big Ball of Mud apresenta uma visão realista sobre o desenvolvimento de software, mostrando que sistemas mal estruturados não surgem apenas por erro ou falta de conhecimento, mas como resultado de pressões reais enfrentadas pelas equipes. O texto ajuda a entender que esse tipo de arquitetura pode ser aceitável em determinados momentos, principalmente nas fases iniciais de um projeto.

No entanto, os autores deixam claro que o problema surge quando não há preocupação em melhorar ou reorganizar o sistema ao longo do tempo. Assim, a principal lição do artigo é a importância de reconhecer quando um sistema está se tornando um Big Ball of Mud e buscar estratégias para evitar que ele se torne insustentável.