

## RESENHA — BIG BALL OF MUD (Foote & Yoder, 1997)

O artigo “Big Ball of Mud” explora um fenômeno comum no desenvolvimento de software: sistemas que, apesar de funcionarem, tornam-se estruturalmente desorganizados, caóticos e difíceis de manter. Os autores argumentam que esse tipo de arquitetura — embora indesejada — é extremamente comum, surgindo não por ignorância, mas por pressões reais do ambiente de desenvolvimento, como prazos curtos, mudanças de requisitos e falta de investimento em arquitetura.

A obra descreve o “Big Ball of Mud” como um sistema sem uma arquitetura clara, composto por código que cresce de forma improvisada, com alto acoplamento, duplicação e ausência de documentação. Em vez de tratar o fenômeno como uma falha moral do desenvolvedor, o artigo revela por que ele ocorre: é a resposta prática e rápida às demandas imediatas, sacrificando organização por entrega contínua.

Os autores apresentam padrões que explicam essas forças. “Throwaway Code” mostra como protótipos rápidos, nunca descartados, evoluem para sistemas permanentes e frágeis. “Piecemeal Growth” discute como alterações incrementais inevitáveis deterioram a arquitetura ao longo do tempo. “Keep It Working” destaca a prioridade da manutenção contínua sobre reformas profundas, reforçando a sobrevivência do sistema mesmo quando sua estrutura já está comprometida. Já “Sweeping It Under the Rug” descreve tentativas de esconder o caos atrás de interfaces limpas, sem resolver a raiz do problema. Por fim, “Reconstruction” trata do momento em que o sistema se torna tão complexo que a única saída viável é reconstruí-lo do zero.

A principal contribuição do artigo é provocar uma reflexão madura: Big Balls of Mud não surgem porque os desenvolvedores querem, mas porque o ambiente de desenvolvimento favorece soluções rápidas e pragmáticas. O texto propõe que, ao compreender essas forças, equipes podem aprender a reconhecer quando estão caminhando para o caos e como corrigir a rota por meio de refactoring, investimento em arquitetura e aprendizado contínuo sobre o domínio da aplicação.

Assim, o artigo é essencial para profissionais e estudantes de software, pois revela a realidade prática dos sistemas no mundo real e ensina que o objetivo não é evitar completamente o “lodo”, mas entender como evoluir dele para arquiteturas mais sólidas e sustentáveis.