

ANÁLISE DO ARTIGO “BIG BALL OF MUD”
VINÍCIUS GOMES RODRIGUES
ENGENHARIA DE SOFTWARE - PUC MINAS
BELO HORIZONTE - 2025

O artigo "Big Ball of Mud" discute uma das arquiteturas de software mais comuns na prática, mas frequentemente negligenciada na literatura: sistemas caóticos, mal estruturados e desenvolvidos de forma ad hoc, denominados "Big Ball of Mud" (BBoM). Os autores argumentam que, embora essa abordagem seja muitas vezes criticada, sua prevalência indica que ela resolve problemas reais do desenvolvimento de software, como prazos curtos, mudanças frequentes e restrições de recursos.

Um BBoM é marcado por características como “código espaguete”, em que possui funções longas, acoplamento excessivo e falta de modularização; evolução desorganizada, provocada por mudanças reativas sem planejamento, documentação ausente, a qual acarreta na dependência de desenvolvedores originais além da dificuldade de manutenção a qual possui efeitos colaterais imprevisíveis. Além disso, é comum encontrar variáveis globais abusivas e lógica duplicada, o que agrava a complexidade.

Foote e Yoder ressaltam que o BBoM não é fruto de incompetência, mas de pressões reais: entregas rápidas, requisitos voláteis, escassez de recursos ou experiência, e a transformação de protótipos em sistemas definitivos. Embora eficaz a curto prazo como em MVPs ou sistemas efêmeros, o BBoM gera dívida técnica, fragilidade e custos crescentes de manutenção a longo prazo.

Como alternativas, os autores propõem isolar partes críticas, criando interfaces limpas ao redor do caos, refatorar incrementalmente (melhorias graduais) ou, em casos extremos, reescrever o sistema. A chave é enxergar o BBoM não como um fracasso, mas como um sintoma de condições adversas, exigindo estratégias pragmáticas para mitigar seus riscos.

Aplicação em Cenário Real:

“Você é um desenvolvedor (ou arquiteto) em uma empresa de e-commerce. O sistema principal foi construído há 5 anos, começou como um MVP e evoluiu sem planejamento. Agora: O código é difícil de modificar, novos desenvolvedores

demoram meses para contribuir, cada nova funcionalidade gera bugs inesperados e o time vive apagando incêndios. Diagnóstico: É um clássico Big Ball of Mud (BBoM).”

Nesse cenário, em vez de culpar equipes anteriores, a abordagem seria:

1. Entender as causas históricas (prazos curtos, equipe enxuta);
2. Identificar pontos críticos (código mais modificado, áreas com mais bugs);
3. Isolar e refatorar módulos estratégicos (como pagamentos ou catálogo);
4. Padronizar práticas (nomenclatura, documentação atualizada semanalmente);
5. Criar uma camada de abstração para novos desenvolvimentos, reduzindo a dependência do código legado.

A lição central é que o BBoM é quase inevitável em contextos de pressão, mas seu impacto pode ser gerenciado com intervenções conscientes, equilibrando entregas imediatas e sustentabilidade a longo prazo.