Resenha do artigo Big Ball of Mud

Aluno: João Vitor Romero Sales
Disciplina: Projeto de Software
Professor: João Paulo Aramuni

## Big Ball of Mud: Uma Jornada pela Arquitetura de Software no Mundo Real

O artigo investiga um padrão arquitetônico de software prevalente, mas frequentemente ignorado, que possui seis padrões:

BIG BALL OF MUD
THROWAWAY CODE
PIECEMEAL GROWTH
KEEP IT WORKING
SWEEPING IT UNDER THE RUG
RECONSTRUCTION

Em contraste com os padrões elegantes e bem definidos encontrados em livros didáticos, como DDD, MVC, e outros, a realidade do desenvolvimento de software frequentemente nos coloca frente a sistemas confusos e desorganizados, uma espécie de favelas de código. Em vez de ignorar essa realidade desconfortável, o artigo se propõe a entendê-la. Afinal, se esta arquitetura (BIG BALL OF MUD) é tão comum, ela deve estar "fazendo algo certo" e os autores exploram as forças que impulsionam sua criação e discutem maneiras de navegar nessa área pantanosa.

### Por que o BIG BALL OF MUD?

Diversos fatores contribuem para a proliferação desse padrão. Prazos apertados frequentemente forçam os desenvolvedores a priorizar funcionalidade em detrimento da arquitetura, resultando em código "quick and dirty" (THROWAWAY CODE) que, apesar de temporário em intenção, torna-se permanente.

Experiência limitada no domínio, rotatividade de membros da equipe e a pressão constante para manter os sistemas funcionando (KEEP IT WORKING) também favorecem o crescimento fragmentado (PIECEMEAL GROWTH), corroendo a estrutura original do sistema.

O artigo argumenta que, embora o BIG BALL OF MUD possa ser um estágio natural no início do desenvolvimento, ignorá-lo pode levar um sistema a se tornar insustentável e difícil de manter a longo prazo.

## Lidando com o BIG BALL OF MUD

Os autores oferecem algumas estratégias para lidar com esse padrão: Esconder a complexidade por trás de fachadas (SWEEPING IT UNDER THE RUG) pode ser uma solução temporária, tornando o sistema mais apresentável sem resolver seus problemas estruturais.

A refatoração, por sua vez, representa um passo mais significativo na direção certa arquiteturalmente. Ao melhorar gradualmente o código e remover duplicações, os desenvolvedores podem começar a recuperar a integridade arquitetônica.

Em casos extremos, a única solução pode ser a reconstrução completa do sistema. No entanto, mesmo nessa situação drástica, as lições aprendidas com o sistema original, especialmente em relação ao domínio e aos requisitos, são valiosas.

## Olhando para o futuro

Os autores argumentam que a chave para evitar a BIG BALL OF MUD não reside em buscar uma arquitetura perfeita desde o início, mas sim em reconhecer a natureza evolutiva do software. Compreender as pressões, restrições e a dinâmica do desenvolvimento de software, como a necessidade de entregar valor rapidamente e responder a mudanças nos requisitos, é crucial para criar sistemas mais resilientes.

Ao invés de condenar este tipo de arquitetura, o artigo encoraja a comunidade de desenvolvimento de software a enfrentá-lo com seriedade, aprendendo com seus erros e buscando continuamente maneiras de melhorar a arquitetura dos sistemas.

## Crítica

Ao se construir sistemas, desde os mais simples aos mais complexos, por mais que exista a vontade e a empolgação inicial de iniciar o processo de desenvolvimento, deve-se imprescindivelmente primeiro pensar na arquitetura do projeto. Esta fase, caso seja necessário, deve ser até mesmo extendida brevemente, uma vez que bem feita, pensando em todos os requisitos levantados e nas melhores decisões, tendo em vista o propósito de utilização do software e o contexto do time de desenvolvimento, acredito que seja possível sim, evitar a arquitetura BIG BALL OF MUD.

Boas práticas como refatoração ou até mesmo reconstrução de partes do sistema, devem também ser empregadas em todo o processo de desenvolvimento, por mais que gerem atrasos na demanda atual do sistema, geram inúmeros benefícios e economias a médio e longo prazo.

Assiduamente, ao surgir a demanda de novos componentes no software, deve-se realizar este processo de tomada de decisão arquitetural antes de iniciar o de desenvolvimento. E este deve ser de fácil implantação, uma vez que desde o início do processo de construção do software, boas práticas foram aplicadas.