

Aluno: Thomás Ramos Oliveira

The Big Ball of Mud

O artigo The Big Ball of Mud ou a grande bola de lama, de Brian Foote e Joseph Yoder, aborda uma realidade desconfortável no desenvolvimento de software: a continuidade de sistemas com arquiteturas casuais e desestruturadas. Os autores argumentam que arquiteturas elevadas e elegantes sejam as melhores práticas, a arquitetura de fato, a mais comum no mercado, é a "Grande Bola de Lama"

O artigo busca trazer não apenas uma crítica mas porque essa abordagem, aparentemente indesejável, é tão popular e eficaz. Um sistema "Big Ball of Mud" é descrito no artigo como uma selva de código de espaguete cuja estrutura mostra sinais claros de crescimento não regulado e reparos inconvenientes. Além disso a informação é compartilhada de forma promiscua entre diferentes partes do sistema, muitas vezes sendo duplicada, e qualquer arquitetura original bem definida pode ter sido erodido a ponto de se tornar irreconhecível.

Foote e Yoder identificam um conjunto de forças que impulsionam organizações, mesmo as mais bem-intencionadas, a produzir tais sistemas. Pressões como tempo e custo são as mais proeminentes; a necessidade de entregar um produto rapidamente para atender a uma janela de mercado muitas vezes se sobrepõe às preocupações arquiteturais de longo prazo. A arquitetura é frequentemente vista como um luxo ou um risco que consome recursos preciosos. Outros fatores incluem a falta de experiência da equipe com o domínio do problema, o que impede a criação de uma arquitetura sofisticada desde o início, a complexidade inerente ao problema que o software tenta resolver e as mudanças constantes nos requisitos, que podem minar decisões arquiteturais fundamentais.

O artigo argumenta que a popularidade duradoura desse modelo significa que ele deve estar "fazendo algo certo". Em vez de condená-lo como um antipadrão, os autores o apresentam como um padrão de direito próprio, acompanhado de outras estratégias de enfrentamento que emergem dele:

- Throwaway Code (Código Descartável): Programas rápidos e sujos criados para um propósito específico, como um protótipo, que acabam se tornando permanentes porque "funcionam". Essas pequenas Gambiarras frequentemente evoluem para sistemas maiores e mais complexos.

- Piecemeal Growth (Crescimento Gradual): O processo de adicionar funcionalidades

de forma incremental, sem um plano mestre. Embora permita que o sistema evolua, esse crescimento descontrolado pode corroer a estrutura ao longo do tempo, levando a um "urban sprawl" (expansão urbana desordenada) de código.

-Keep It Working (Mantenha Funcionando): Um imperativo para sistemas críticos que não podem parar. Essa necessidade prioriza reparos imediatos e pequenos em vez de grandes revisões arquiteturais, garantindo a viabilidade do sistema em detrimento de sua pureza estrutural.

-Sweeping It Under the Rug (Varrendo para Debaixo do Tapete): Uma estratégia pragmática para lidar com o caos. Quando uma parte do sistema está bagunçada demais para ser consertada, ela pode ser isolada ou escondida atrás de uma interface limpa (uma Fachada). Isso limita a desordem e permite que o restante do sistema evolua de forma mais controlada.

-Reconstruction (Reconstrução): A solução final quando um sistema se torna economicamente inviável para manter ou evoluir. Envolve descartar a implementação antiga e recomeçar, mas com um benefício crucial: a experiência e o conhecimento do domínio adquiridos com o sistema anterior.

Por fim, "The Big Ball of Mud" não é uma condenação, mas um reconhecimento de que a arquitetura de software no mundo real é um processo confuso e evolutivo. Os autores sugerem abraçar um ciclo de aprendizado ao invés de criticar esses modelos. A abordagem "faça funcionar, faça direito, faça rápido" de Kent Beck é citada como um caminho: primeiro, focar na funcionalidade para entender o problema e, só então, refatorar em direção a uma arquitetura mais limpa, à medida que o conhecimento do domínio se aprofunda.

Caso Real de mercado:

A MP Fitness é um e-commerce de suplementos e acessórios para academia que começou com um website simples, apenas listando produtos e permitindo compras online. Com o tempo, novas oportunidades foram surgindo e o site recebeu funcionalidades como programa de pontos, blog e promoções sazonais. Tudo era feito com foco em entregar rápido, sem um planejamento arquitetural profundo. Por isso, o sistema acabou ficando no estilo descrito no artigo Big Ball of Mud: código espalhado, difícil de manter e cheio de gambiarras, mas ainda funcional para o negócio.

Quando surgiu a ideia de lançar a linha vegana, na minha opinião, a melhor solução para o prazo apertado seria aplicar o Piecemeal Growth, mas já planejando uma refatoração futura. Eu adicionaria os campos e regras necessárias de forma

incremental, como filtros e promoções específicas, para garantir que a funcionalidade fosse lançada no tempo certo, mas documentando e isolando o que precisaria ser melhorado depois.

Com o site ficando mais complexo, eu aplicaria o Keep It Working, priorizando pequenos ajustes e correções pontuais para não comprometer a operação do website. A ideia seria manter a estabilidade do sistema sem arriscar que uma grande mudança derrubasse funcionalidades críticas.

Quando percebesse que certas partes do código estavam se tornando muito confusas, eu aplicaria o Sweeping It Under the Rug junto com uma Façade. Criaria um serviço intermediário de catálogo que concentrasse todas as regras de produtos, permitindo que o restante do site consumisse essas informações de forma mais limpa, sem lidar diretamente com a bagunça interna. Isso facilitaria a manutenção e prepararia o terreno para melhorias maiores.

Quando chegasse o momento de criar um website exclusivo para a marca vegana integrado ao estoque e ao sistema de pedidos, na minha opinião a melhor solução seria partir para a Reconstruction. Eu reescreveria o sistema do zero, aproveitando o conhecimento adquirido com os anos, e usaria uma arquitetura baseada em microsserviços para permitir que as duas marcas tivessem vitrines separadas mas compartilhassem a mesma base de dados e serviços, garantindo escalabilidade e organização.

No fim, eu vejo que aplicar os conceitos do artigo Big Ball of Mud não significa trabalhar de forma desleixada, mas sim entender que no mundo real muitas vezes precisamos “fazer funcionar rápido” e depois evoluir o sistema de forma planejada. O importante é saber quando cada padrão deve ser usado para equilibrar velocidade de entrega e sustentabilidade da arquitetura.