

Entre a Arquitetura e a Realidade: Uma Análise sobre "The Big Ball of Mud"

A computação é um ramo da matemática, criada inicialmente para realizar cálculos complexos de forma ágil e eficiente, evitando o máximo de erros. Com a evolução da área e da complexidade dos sistemas, problemas recorrentes foram sendo detectados e padrões de soluções foram criados para lidar com cenários específicos, visando à escalabilidade, à organização e à aceleração no desenvolvimento de software.

Nesse sentido, destacam-se padrões como o “Singleton”, em que, no ambiente de desenvolvimento, uma classe suporta apenas um tipo de instância; o “Abstract Factory”, que utiliza interfaces para projetar comportamentos genéricos e injeta esses comportamentos em classes “fábrica”, que possuem métodos para fabricar produtos de um tipo particular; e o “MVC”, padrão que herda os conceitos de estruturação da arquitetura de camadas e separa as responsabilidades do software em três camadas principais: *Model*, que é responsável pelo modelo das entidades; *View*, que é encarregada das telas/visualização do sistema; e *Controller*, que faz a ponte entre as entidades e o que é mostrado ao usuário.

Apesar de existirem inúmeros modelos de desenvolvimento que auxiliam cotidianamente na construção de aplicações, os profissionais ainda têm que lidar com duas das maiores dificuldades: a pressão e o curto prazo para construir um produto. Em um ecossistema de mercado onde novas tecnologias são lançadas diariamente, novos modelos são construídos, sempre há um novo concorrente e tudo é demandado "para ontem", os engenheiros e arquitetos se deparam com decisões difíceis que devem ser tomadas para dar continuidade ao projeto. São nessas decisões que entra o padrão de arquitetura de software mais comum na prática: “The Big Ball of Mud” (A Grande Bola de Lama).

No artigo “The Big Ball of Mud”, os autores a descrevem como um sistema que não possui uma arquitetura aparente, sendo caracterizada por ser uma estrutura "desordenada, extensa, desleixada, feita de fita adesiva e arame farpado, uma selva de códigos espaguete". Apesar disso, o objetivo do texto não é condenar ou invalidar essa abordagem, mas destacar sua popularidade duradoura, visto que, de certa perspectiva, essa estruturação resolve problemas reais sob a pressão do mundo real. Dessa forma, a essência da obra é entender como e por que ela surge.

Em vista disso, Foote e Yoder apresentam algumas razões pelas quais a bola de lama surge e persiste. O “Throwaway Code” (Código Descartável) é um código funcional que foi escrito de forma rápida e suja para atender a um propósito específico e temporário, mas que, por atender a esse propósito e pela pressão de clientes ou prazos, acaba sendo mantido e não evoluído. O “Piecemeal Growth” (Crescimento Fracionado) ocorre quando o software evolui de forma incremental, adicionando funcionalidades de maneira localizada e oportunista, sem levar em conta a arquitetura global da aplicação; essa abordagem causa uma “erosão arquitetônica”, e cada nova

adição, feita para resolver um problema, compromete a estrutura geral do sistema, levando-o a se tornar uma bola de lama. O "Keep It Working" (Mantenha Funcionando) é uma estratégia de manutenção que prioriza manter o software em funcionamento. Esse comportamento incentiva pequenas modificações e rejeita grandes revisões, vistas como arriscadas, e embora preserve a vitalidade da operação, pode impedir melhorias arquitetônicas. O "Sweeping It Under the Rug" (Varrendo para Debaixo do Tapete) é uma prática que negligencia a organização e tenta, ao menos, esconder a bagunça no código; isso mantém a fachada limpa, mas deteriora o núcleo do programa. Após todos esses pontos, o software provavelmente se encontrará em um status impossível de manutenção e evolução, causando a sua reconstrução total.

Dessa forma, podemos concluir que a Grande Bola de Lama é um padrão arquitetural não desejado nas implementações de sistemas, mas muito presente no ambiente real de produção. Além disso, ela é causada principalmente por questões que fogem do controle dos desenvolvedores, como tempo e custo, mas também tem sua justificativa na má elaboração do ciclo de desenvolvimento de software, em que o problema que o software resolve é mal compreendido, o que gera mudanças frequentes nos requisitos da aplicação. Tais mudanças podem ir diretamente contra premissas estabelecidas anteriormente para a arquitetura original, exigindo a perturbação da arquitetura para acomodar a mudança. Por fim, podemos entender que a "Grande Bola de Lama" não deve ser vista como um fracasso definitivo, mas como uma fase quase inevitável no ciclo de vida de um software. O verdadeiro desafio não é evitar a lama, mas sim criar uma cultura de desenvolvimento que valorize o aprendizado contínuo, consolidação e refatoração.

Referências:

FOOTE, Brian; YODER, Joseph. The Big Ball of Mud. In: *Fourth Conference on Pattern Languages of Programs (PLoP '97)*, Monticello, Illinois, 1997.

Autor da resenha: Rafael de Faria Neves Alves Franco