Universidade Catolica de Minas Gerais

Projeto de Software
2024

Prof. João Paulo Carneiro Aramuni
Aluno: Jhonatan Gutemberg Rosa Ferreira

Resenha do artigo:

Big Ball of Mud

Brian Foote, Joseph Yoder

Os padrões de arquitetura de software se complementam para promover boas práticas e, muitas vezes, "apagar incêndios"em situações críticas. As "grandes bolas de lama"surgem ao longo do projeto devido a fatores como falta de tempo para planejamento, prazos curtos de entrega ou desconhecimento de boas práticas arquiteturais e padrões de projeto.

Os padrões Big Ball of Mud, Throwaway Growth, Piecemeal Growth, Keep it Working, Sweeping it Under the Rug e Reconstruction não são necessariamente antipadrões, pois têm objetivos específicos, como entrega rápida e baixo custo. Porém, a qualidade do software é afetada a médio e longo prazo, resultando em manutenção complexa, alta taxa de bugs e dificuldade na adição de novas funcionalidades.

Durante o desenvolvimento de um software, seja ele novo ou uma continuação, a equipe muitas vezes se depara com a necessidade de resolver problemas urgentes. Nessas situações, é importante tomar decisões rápidas e eficazes sobre manter o sistema funcionando da forma como está ou realizar intervenções mais profundas. Algumas forças tem influência na decisão, sendo necessário analisar cuidadosamente a arquitetura do software, levando em conta várias premissas importantes.Por exemplo, o tempo , a experiência e habilidade dos membros da equipe, a complexidade do sistema, a capacidade de lidar com mudanças, e os custos associados a essas decisões.

Mesmo em grandes instituições e organizações as grandes bolas de lama estão presentes. Elas nascem por falta de tempo quando uma nova funcionalidade precisa ser implementada e o prazo para elaboração é curto. Também, a experiência da equipe relacionado a uma nova arquitetura pode influenciar na melhora ou na piora da abstração dos problemas em particular.

Quando um sistema não tem uma arquitetura bem definida, ele se torna complexo, e até mesmo pequenos problemas podem ser difíceis e arriscados de resolver. As novas funcionalidades ou melhorias podem quebrar o sistema devido à fragilidade de sua arquitetura. Além disso, o custo para realizar qualquer ação no projeto se eleva, e é nesse ponto que muitos investidores desistem. Para evitar essa situação, as "bolas de lama"acabam surgindo como uma tentativa de contornar o problema.

Em algumas situações, as equipes criam protótipos seguindo o padrão "Big Ball of Mud", visando entregar o sistema rapidamente, com a intenção de corrigir as pequenas "bolas de lama"ao longo do projeto. No entanto, isso geralmente não acontece, pois prevalece a mentalidade de "se está funcionando, não é necessário alterar".

O uso de códigos descartáveis é uma prática onde se desenvolve um protótipo com a intenção de ser descartado após o uso temporário. No entanto, isso muitas vezes não ocorre, assim como um pequeno galpão construído para produções menores raramente é demolido quando um galpão permanente é erguido. E geralmente essas soluções são usadas quando o prazo de entrega já está próximo.

O desenvolvimento incremental divide o processo em partes menores, buscando entregar funcionalidades em ciclos menores.Esse formato de evolução permite uma maior adaptabilidade do projeto ao cliente, pois permite mudanças ao longo da evolução do sistema.

Muitos sistemas exigem atenção constante, e uma interrupção pode causar perda de vida e prejuízos financeiros significativos. Nessas condições, o desenvolvimento de subsistemas pode minimizar riscos de interrupções desnecessárias. Embora a abordagem reativa não seja ideal globalmente, ela elimina erros evidentes e facilita a correção de problemas menos óbvios.

Ocultar uma grande bola de lama é uma abordagem que visa manter a fluidez do sistema, partindo do princípio de que, se não é possível eliminar os problemas, pelo menos é melhor ocultá-los. Código desorganizado e emaranhado é difícil de manter e evoluir, e tende a piorar ainda mais se não for controlado.

Por fim, a reconstrução, nem sempre é possível aproveitar o código de um software desenvolvido de forma desordenada. Novos requisitos podem inviabilizar a continuidade do código sendo mais barata financeiramente recomeçar do que continuar a evolução. Outras causas possíveis para a reconstrução podem ser citados troca de equipe, tecnologia ultrapassada e complexidade para alteração do código atual.

Cada padrão citados é usado de acordo com as características de cada equipe, projeto e experiências dos envolvidos. Não é possível deixar de salientar que esses padrões muitas vezes podem ser usados como estratégias para manter alguns cargos no time, porem isso não é uma justificativa para sua utilização.