

Resenha Crítica do artigo *Documenting Architecture Decisions*

Introdução

O artigo *Documenting Architecture Decisions*, publicado em 2011 por Michael Nygard no blog da Cognitect, propõe um método pragmático e objetivo para a documentação de decisões arquiteturais em projetos de software. Inserido no contexto das metodologias ágeis, o texto busca responder a uma questão recorrente: como registrar decisões técnicas relevantes sem recorrer a extensos documentos, que geralmente se tornam obsoletos e ineficazes. Nesse sentido, o autor introduz o conceito de *Architecture Decision Records* (ADRs), registros curtos e padronizados destinados a preservar não apenas o conteúdo de uma decisão, mas, sobretudo, as motivações, o contexto e as consequências associadas a ela.

A importância dessa proposta reside no fato de que, em projetos de longo prazo ou em ambientes com alta rotatividade de equipes, a ausência de documentação adequada leva a riscos consideráveis. Decisões podem ser aceitas cegamente ou revertidas de forma precipitada, sem avaliação das implicações, comprometendo a sustentabilidade do sistema. Dessa forma, o artigo de Nygard contribui para a consolidação de práticas ágeis de documentação, oferecendo um modelo capaz de equilibrar leveza e valor informacional.

Desenvolvimento

O ponto de partida do autor é a crítica aos documentos arquiteturais extensos e centralizados, considerados inviáveis em projetos ágeis. Nygard observa que tais artefatos dificilmente são atualizados, além de apresentarem pouca utilidade prática para desenvolvedores e demais stakeholders. Em contrapartida, a proposta dos ADRs baseia-se na produção de registros pequenos, versionados junto ao código, que preservam a memória das escolhas arquiteturais e podem ser mantidos de forma incremental.

Cada ADR adota uma estrutura padronizada composta por cinco elementos: **título**, **contexto**, **decisão**, **status** e **consequências**. Essa organização, inspirada no estilo das *design patterns* de Christopher Alexander, permite que cada decisão seja compreendida em termos de forças em tensão e da resposta encontrada pela equipe. O aspecto central não é apenas registrar a decisão em si, mas documentar as razões subjacentes e seus possíveis desdobramentos positivos, negativos ou neutros.

Um diferencial importante é a preservação do histórico: quando uma decisão é revista ou substituída, o ADR correspondente não é apagado, mas marcado como *superseded*. Essa

prática evita a perda de informações valiosas sobre o percurso do projeto, além de fornecer subsídios para avaliar mudanças de contexto que motivaram a revisão.

A experiência relatada pelo autor reforça a relevância da proposta. Equipes que adotaram ADRs em projetos reais relataram benefícios como maior clareza no entendimento do sistema, facilidade no planejamento de reestruturações futuras e maior alinhamento entre desenvolvedores e clientes. Embora exista a preocupação de que o uso de repositórios de código possa restringir o acesso de stakeholders não técnicos, Nygard observa que plataformas como o GitHub mitigam esse problema ao disponibilizar os documentos de forma amigável.

Aplicação prática no mercado

A aplicação prática do conceito de ADRs pode ser observada em projetos de **migração arquitetural em larga escala**, como a transformação de sistemas monolíticos em microsserviços. Nesses contextos, decisões fundamentais — escolha de mecanismos de comunicação assíncrona, padronização de interfaces, estratégias de monitoramento ou políticas de resiliência — precisam ser registradas de forma clara e acessível.

Por exemplo, ao decidir entre Kafka ou RabbitMQ como barramento de mensagens, a equipe pode registrar no ADR os fatores técnicos e organizacionais que influenciaram a escolha, as vantagens esperadas e os riscos associados. Caso no futuro a decisão seja revisada, o ADR original continuará servindo como referência histórica, enquanto um novo documento registrará o contexto atualizado.

Além disso, startups que priorizam a entrega rápida de produtos mínimos viáveis podem se beneficiar da prática, utilizando ADRs para registrar decisões provisórias. Isso garante que, no momento em que a arquitetura precisar evoluir, as razões por trás das escolhas originais estarão disponíveis, evitando que tais decisões sejam mantidas ou descartadas de forma arbitrária.

Conclusão

O artigo de Michael Nygard apresenta uma contribuição significativa para a prática da engenharia de software ao propor um método de documentação compatível com os princípios ágeis. Os ADRs se destacam por sua simplicidade, clareza e potencial de manutenção, proporcionando uma forma de registrar não apenas as decisões arquiteturais, mas também o raciocínio e as circunstâncias que as motivaram.

Ao criticar a ineficácia dos documentos extensos e defender registros modulares, o autor propõe um modelo que fortalece a rastreabilidade, a transparência e a sustentabilidade dos projetos. Em contextos de alta complexidade, os ADRs funcionam como instrumentos de preservação do conhecimento organizacional, reduzindo riscos de estagnação ou de mudanças precipitadas.

Em síntese, *Documenting Architecture Decisions* reafirma a necessidade de práticas de documentação que equilibrem agilidade e longevidade, e oferece uma ferramenta aplicável tanto em projetos corporativos de grande porte quanto em iniciativas ágeis de menor escala. Seu impacto ultrapassa a dimensão técnica, estendendo-se à governança de projetos e ao alinhamento entre equipes técnicas e de negócios.