

Resenha sobre o artigo: Documenting Architecture Decisions

O artigo de Michael Nygard apresenta uma abordagem pragmática para documentar decisões arquiteturais em projetos ágeis através dos Architecture Decision Records (ADRs). O autor reconhece uma dor comum no desenvolvimento de software: a perda do contexto e da motivação por trás de decisões importantes ao longo do tempo. Quando novos desenvolvedores entram em um projeto, frequentemente se deparam com escolhas que parecem estranhas ou questionáveis, sem ter acesso ao raciocínio que as fundamenta. Essa falta de contexto leva a dois comportamentos prejudiciais - aceitar cegamente decisões que deveriam ser revisitadas ou alterar impulsivamente escolhas que ainda são válidas. Os ADRs surgem como solução para esse dilema, propondo registros curtos, numerados sequencialmente e versionados junto ao código, que capturam o contexto, a decisão tomada, seu status e suas consequências de forma clara e acessível.

A estrutura proposta para os ADRs é notavelmente simples e eficaz, composta por apenas cinco elementos: título descritivo, contexto das forças em jogo, a decisão em si (escrita em voz ativa), status da decisão e suas consequências. Essa simplicidade é intencional e alinha-se perfeitamente com os princípios ágeis de documentação útil e mantida. Os documentos devem ter apenas uma ou duas páginas, escritos como uma conversa com um desenvolvedor futuro, usando prosa completa em vez de fragmentos ou bullet points excessivos. Decisões obsoletas não são removidas, mas marcadas como supersededidas, preservando o histórico de raciocínio que pode ser valioso para compreender a evolução do sistema.

O conceito de ADR demonstra uma maturidade importante no entendimento de que documentação não é inimiga da agilidade - documentação sem valor é. A abordagem modular e enxuta dos ADRs contrasta fortemente com especificações monolíticas que ninguém lê ou atualiza. Além disso, o artigo destaca um aspecto crucial: as consequências de uma decisão frequentemente se tornam o contexto para decisões subsequentes, criando uma linguagem de padrões similar à proposta por Christopher Alexander. A escolha de manter os registros no controle de versão, embora possa parecer limitante para stakeholders não técnicos, na prática se mostra conveniente com ferramentas como GitHub que renderizam Markdown automaticamente.

A proposta de Nygard merece reconhecimento por sua praticidade e elegância. Em vez de criar mais um processo burocrático, os ADRs se integram naturalmente ao fluxo de trabalho de desenvolvimento e oferecem benefícios tangíveis: onboarding mais rápido de novos membros, decisões mais conscientes sobre quando mudar ou manter escolhas arquiteturais, e preservação do conhecimento institucional. O feedback positivo relatado pelo autor, mesmo em um período curto de adoção, sugere que a técnica preenche uma lacuna real. No contexto atual de microsserviços, sistemas distribuídos e equipes remotas, onde a transferência de conhecimento é ainda mais desafiadora, os ADRs se tornam ainda mais relevantes como ferramenta de governança técnica leve e eficaz.

Aluno: João Marcelo Carvalho Pereira Araújo

Professor: João Paulo Aramuni

Disciplina: Projeto de Software