

Resenha – “Documenting Architecture Decisions”, de Michael Nygard (2011)

O artigo “Documenting Architecture Decisions”, escrito por Michael Nygard, fala de um jeito bem direto sobre algo que muita gente na área de desenvolvimento sente na pele: a dificuldade de entender *por que* certas decisões arquiteturais foram tomadas em um projeto. Nygard traz uma proposta simples, mas poderosa — criar registros curtos e objetivos chamados ADRs (Architecture Decision Records) para documentar decisões importantes de arquitetura de software.

Logo de cara, ele critica aquela velha mania de produzir documentos enormes e inúteis, que ninguém lê nem atualiza. Segundo o autor, a ideia não é ser contra a documentação, mas sim contra documentação sem valor. Ele defende algo mais leve, que realmente sirva à equipe — pequenas anotações que registrem as decisões arquiteturais e o raciocínio por trás delas. Em vez de um livro de 300 páginas que envelhece em duas semanas, ele sugere documentos curtos, escritos em Markdown, guardados junto com o código no repositório do projeto.

O coração da proposta são os ADRs — cada um registra uma única decisão importante, com seções simples: *Contexto*, *Decisão*, *Status* e *Consequências*. É quase uma conversa com o “eu” do futuro (ou com o próximo desenvolvedor que entrar no time). A ideia é explicar não só o que foi decidido, mas *por que* aquilo fez sentido naquele momento. Assim, evita-se que futuros desenvolvedores mudem algo sem entender o impacto, ou aceitem uma decisão antiga sem questionar se ela ainda faz sentido.

Nygard também destaca um ponto bem real: em projetos ágeis, as decisões não acontecem todas de uma vez, então é essencial ter um jeito prático de registrar e revisar essas escolhas ao longo do tempo. Ele mostra ainda que manter os ADRs no repositório é melhor do que jogá-los em um wiki esquecido — afinal, o código e as decisões caminham juntos.

No final, o autor compartilha a própria experiência com o método, dizendo que os resultados têm sido positivos. As equipes que adotaram ADRs conseguiram manter o contexto vivo e facilitar a entrada de novos desenvolvedores. Mesmo os clientes, segundo ele, gostaram da transparência e clareza que os registros trazem.

No fundo, o artigo é um convite para repensar a forma como documentamos software: menos burocracia, mais utilidade. É sobre criar uma memória viva das decisões do projeto, escrita de um jeito humano e acessível. E convenhamos — para quem já se perdeu tentando entender o que “eles estavam pensando” quando fizeram certa escolha no código, essa proposta soa quase como um alívio.