

Aluno: Thomás Ramos Oliveira

Documenting Architecture Decisions

Michael Nygard

Em projetos ágeis, a forma de documentar a arquitetura precisa ser diferente da abordagem tradicional, uma vez que nem todas as decisões arquitetônicas são tomadas de uma só vez, nem todas ocorrem no início do projeto. A filosofia ágil não é contrária à documentação, mas sim à documentação que não agrega valor real à equipe. Documentos volumosos ou desatualizados tendem a se tornar inúteis, pois dificilmente são lidos ou mantidos em dia. Por isso, Nygard defende o uso de registros pequenos e modulares, que possam ser facilmente atualizados e consumidos por todos os envolvidos. Um dos principais desafios em qualquer projeto de software é o acompanhamento das motivações por trás das decisões tomadas. Desenvolvedores que ingressam em projetos já em andamento frequentemente se deparam com escolhas anteriores sem entender o raciocínio que as motivou, o que gera duas possibilidades: aceitar a decisão sem questionar ou alterá-la sem compreender suas consequências. Aceitar decisões cegamente pode ser aceitável se o contexto original ainda for válido, mas pode se tornar problemático caso o cenário tenha mudado, exigindo uma revisão cuidadosa da decisão. Por outro lado, alterar decisões sem compreender completamente sua motivação pode resultar em impactos negativos inesperados, afetando requisitos não funcionais ou outros aspectos críticos do projeto. Portanto, a proposta de Nygard é evitar ambos os extremos, documentando de maneira clara e objetiva a razão de cada decisão arquitetônica.

Para lidar com esse desafio, o autor propõe a criação de registros denominados Architecture Decision Records, ou ADRs, que documentam decisões arquitetônicas significativas, ou seja, aquelas que impactam a estrutura do sistema, suas características não funcionais, dependências, interfaces e técnicas de construção. Cada ADR consiste em um arquivo de texto curto, de preferência utilizando uma linguagem leve de formatação como Markdown ou Textile, e armazenado no repositório do projeto, geralmente em uma pasta específica como `doc/arch/adr-NNN.md`. As ADRs são numeradas sequencialmente, e decisões que eventualmente forem revertidas permanecem registradas como “superseded”, mantendo assim o histórico do projeto. O formato de um ADR é simples, estruturado em cinco partes principais: título, contexto, decisão, status e consequências. O título deve ser uma frase curta e objetiva que descreva a decisão, como por exemplo “ADR 1: Deployment on Ruby on Rails 3.0.10”. A seção de contexto apresenta as forças que influenciam a decisão, que podem ser tecnológicas, políticas, sociais ou específicas do projeto, sempre de forma

neutra, sem juízo de valor. A decisão propriamente dita descreve a resposta às forças identificadas, escrita em frases completas, com voz ativa, utilizando construções como “Nós decidimos...”. O status indica se a decisão está proposta, aceita, depreciada ou substituída por outra. Por fim, a seção de consequências descreve todos os impactos resultantes da decisão, sejam eles positivos, negativos ou neutros, oferecendo à equipe uma visão clara de como a escolha afetará o desenvolvimento futuro do projeto. Cada ADR deve ser conciso, de uma a duas páginas, escrito como se fosse uma conversa com futuros desenvolvedores, enfatizando a clareza, a organização dos parágrafos e evitando a fragmentação de ideias que ocorre frequentemente em listas ou bullets.

O principal benefício do uso de ADRs é que cada registro captura uma decisão arquitetônica significativa e suas consequências, criando um contexto que servirá de base para decisões subsequentes. Dessa forma, mesmo quando há mudanças na equipe de desenvolvimento ou no quadro de stakeholders, o raciocínio por trás das escolhas anteriores permanece visível e compreensível. Isso evita confusões, mal-entendidos ou alterações impulsivas que poderiam prejudicar a qualidade e a consistência do projeto. Além disso, a documentação clara das decisões facilita a gestão de mudanças, pois qualquer modificação necessária no futuro pode ser avaliada à luz do contexto original, reduzindo riscos e garantindo que os objetivos arquitetônicos sejam respeitados. Experiências práticas com ADRs mostraram resultados positivos. Em projetos que adotaram essa abordagem desde agosto de 2011, várias equipes, compostas por seis a dez desenvolvedores que se revezaram ao longo do tempo, relataram que a documentação forneceu um nível de contexto altamente útil. Desenvolvedores novos puderam rapidamente entender decisões passadas sem a necessidade de longas reuniões explicativas, e clientes que planejavam reestruturações futuras encontraram nas ADRs uma forma de registrar intenções de longo prazo sem comprometer o andamento atual do sistema. Embora possa haver a objeção de que manter esses registros no controle de versão com o código fonte torne-os menos acessíveis a gerentes de projeto ou stakeholders externos, a prática mostrou que, em repositórios como os do GitHub, é possível compartilhar links diretos para as versões mais recentes, garantindo que a documentação continue acessível e visualmente amigável, como se fosse uma página de wiki.

Em resumo, os ADRs representam uma ferramenta valiosa para projetos ágeis, permitindo registrar, de forma clara e concisa, decisões arquitetônicas significativas e suas consequências, oferecendo contexto contínuo para desenvolvedores presentes e futuros, clientes e demais stakeholders. Eles garantem que as decisões não se percam ao longo do tempo, que os motivos de cada escolha permaneçam explícitos e que a equipe possa evoluir o sistema com segurança, sem medo de alterar ou manter

decisões sem compreender seu impacto. A prática de documentar arquitetura dessa maneira promove maior transparência, continuidade e previsibilidade no desenvolvimento, assegurando que cada decisão estratégica esteja fundamentada e compreendida por todos os envolvidos. Além disso, ao manter essas informações organizadas em arquivos curtos e acessíveis, evita-se o problema comum de documentos volumosos e desatualizados que raramente são consultados. A adoção de ADRs não apenas facilita a integração de novos membros da equipe, mas também auxilia stakeholders externos a compreenderem a lógica por trás das escolhas técnicas, fortalecendo a colaboração e reduzindo conflitos ao longo do projeto. Com base na experiência relatada, ADRs permitem que as intenções de longo prazo sejam comunicadas claramente, preparando o terreno para futuras alterações arquitetônicas sem comprometer a estabilidade atual do sistema, tornando-os uma prática recomendada para equipes que buscam combinar agilidade com documentação eficaz.