

ALUNO: João Gabriel Santos Andrade Almeida

No desenvolvimento ágil, a arquitetura de software exige uma abordagem distinta dos métodos tradicionais, pois as decisões não são definidas antecipadamente nem tomadas de uma vez. Em vez disso, reconhece-se que o entendimento e o conhecimento do sistema se aprimoram com a experiência, exigindo uma arquitetura adaptável. A documentação tem seu valor, desde que seja relevante para a equipe e de utilidade prática. Documentos longos raramente são atualizados e perdem o propósito. Documentos concisos e diretos têm maior probabilidade de permanecer precisos e servir como referência útil para os desenvolvedores. A verdade é que documentos técnicos extensos raramente são lidos. Muitos desenvolvedores já se depararam com projetos em que a documentação de especificação era maior que o código, dificultando sua leitura ou alteração. Informações concisas são mais fáceis de assimilar e manter, auxiliando na compreensão do sistema.

Um dos maiores desafios em qualquer projeto é entender o porquê de certas decisões. Um novo membro da equipe pode se sentir confuso ou frustrado ao encontrar decisões passadas sem uma explicação clara. Sem o contexto ou as motivações por trás de cada escolha, a pessoa tem duas opções ruins: aceitar a decisão sem questionar ou alterá-la sem entender as implicações. Concordar com uma decisão sem entender sua lógica pode ser arriscado se o cenário tiver mudado. Modificar a decisão sem compreender o raciocínio original pode prejudicar a arquitetura ou o valor do projeto, principalmente se a decisão estiver relacionada a um requisito não funcional, como segurança ou desempenho. O projeto corre o risco de se tornar inflexível e resistente a mudanças ou de perder coerência e estabilidade.

Para mitigar esse problema, sugere-se a utilização dos Registros de Decisões de Arquitetura, ou ADRs. Cada ADR é um documento conciso que descreve uma decisão arquitetural importante que afeta a estrutura, os componentes, os padrões ou as características não funcionais do sistema. O objetivo é registrar a motivação e o raciocínio por trás de cada escolha técnica, de forma simples, clara e rastreável. Assim, todos, incluindo os futuros desenvolvedores, podem entender por que as decisões foram tomadas, quais eram as circunstâncias e quais as consequências esperadas.

Os Registros de Decisão Arquitetural (ADRs) devem ser guardados no repositório do projeto, geralmente em uma pasta nomeada como doc/arch, sendo identificados por números sequenciais. Cada um desses registros é um arquivo de texto conciso, elaborado em uma linguagem leve como Markdown, com extensão de uma ou duas

páginas. Mesmo quando uma decisão é revogada, o documento original é preservado e sinalizado como "substituído", indicando sua revogação, mas mantendo seu valor histórico. Essa abordagem promove a transparência e a rastreabilidade, permitindo uma consulta facilitada do histórico das decisões arquiteturais.

Um ADR é estruturado em algumas seções simples. O título resume a decisão de forma sucinta e objetiva, utilizando uma frase nominal, como "ADR 1: Implantação no Ruby on Rails 3.0.10". O contexto detalha as influências presentes no momento da decisão — aspectos técnicos, políticos, sociais ou específicos do projeto —, sempre de maneira imparcial e descritiva, sem emitir juízos de valor. A seguir, a decisão é apresentada em frases completas e diretas, frequentemente iniciando com "Decidimos..." ou "Nós vamos...", descrevendo de forma clara o caminho selecionado. O status aponta o estágio da decisão, podendo ser proposta, aceita, obsoleta ou substituída. Por fim, as consequências explicam os resultados esperados dessa decisão, abrangendo tanto os efeitos positivos quanto os negativos, uma vez que toda escolha técnica envolve concessões que precisam ser compreendidas.

Um ADR de qualidade deve ser redigido como uma conversa com um futuro desenvolvedor, alguém que lerá o documento meses ou anos depois e precisará compreender a lógica por trás da decisão. Portanto, a escrita deve ser clara, completa e bem organizada, com frases completas e parágrafos coerentes. Listas podem ser utilizadas apenas para facilitar a leitura, mas nunca como substituição de uma explicação detalhada. O objetivo é que o leitor consiga entender a decisão sem necessitar de informações adicionais.

A principal vantagem dessa prática é a criação de um registro claro e acessível das decisões que moldam o projeto, fomentando a transparência e o aprendizado contínuo. Cada ADR detalha uma decisão que influencia as seguintes, originando uma espécie de linguagem de padrões em que decisões maiores definem o escopo para as menores. Isso estabelece uma estrutura coerente e rastreável, na qual a lógica arquitetural pode ser acompanhada ao longo do tempo. Outra vantagem significativa é que o uso de ADRs impede que o conhecimento fique restrito a alguns poucos desenvolvedores experientes, integrando-o ao próprio código do projeto. Assim, mesmo em caso de mudanças na equipe, o entendimento sobre as decisões permanece acessível a todos.

O artigo também compartilha vivências concretas da aplicação de ADRs em projetos. A aceitação por parte dos programadores e dos clientes tem sido bem

favorável desde a implementação. Mesmo em prazos pequenos, como algumas semanas ou meses, as equipes notaram melhorias relevantes na compreensão e na coordenação. Novos integrantes da equipe conseguiram assimilar ligeiramente o cenário e as metas futuras, diminuindo o período de ambientação. Em projetos onde os contratantes preveem uma remodelação futura da plataforma, os ADRs foram valiosos para documentar as perspectivas a longo prazo, impedindo que escolhas momentâneas acabem comprometendo alterações arquitetônicas futuras.