

Resenha: Documenting Architecture Decisions

O artigo "Documenting Architecture Decisions" de Michael Nygard, publicado no blog da Cognitect em 15 de novembro de 2011, aborda a importância e a forma de documentar decisões arquitetônicas em projetos ágeis. Nygard argumenta que, embora métodos ágeis priorizem software funcional sobre documentação abrangente, a documentação de decisões arquitetônicas é crucial para a sustentabilidade e a compreensão de um projeto ao longo do tempo.

Contexto: A Necessidade de Documentação em Projetos Ágeis

Nygard inicia o artigo destacando que a arquitetura para projetos ágeis precisa ser descrita e definida de forma diferente. Nem todas as decisões são tomadas de uma vez, nem todas são finalizadas no início do projeto. Ele enfatiza que métodos ágeis não são contra a documentação, mas sim contra a documentação sem valor. Documentos que auxiliam a própria equipe podem ter valor, mas apenas se forem mantidos atualizados. Documentos grandes raramente são atualizados, enquanto documentos pequenos e modulares têm uma chance maior.

Um ponto crucial levantado é que ninguém lê documentos grandes. Nygard menciona que muitos desenvolvedores já trabalharam em projetos onde o documento de especificação era maior (em bytes) do que o código-fonte total. Tais documentos são grandes demais para serem abertos, lidos ou atualizados. Peças menores e "mastigáveis" são mais fáceis para todos os stakeholders consumirem.

Um dos aspectos mais difíceis de rastrear durante a vida de um projeto é a motivação por trás de certas decisões. Uma nova pessoa que entra no projeto pode ficar perplexa, confusa ou irritada com alguma decisão passada. Sem entender a lógica ou as consequências, essa pessoa tem apenas duas escolhas:

1. **Aceitar cegamente a decisão:** Isso pode ser aceitável se a decisão ainda for válida. No entanto, pode ser problemático se o contexto mudou e a decisão deveria ser revisitada. Se o projeto acumular muitas decisões aceitas sem compreensão, a equipe de desenvolvimento pode ficar com medo de mudar qualquer coisa, e o projeto pode colapsar sob seu próprio peso.
2. **Mudar cegamente a decisão:** Isso pode ser aceitável se a decisão precisar ser revertida. Por outro lado, mudar a decisão sem entender sua motivação ou consequências pode significar prejudicar o valor geral do projeto sem perceber (por exemplo, a decisão apoiava um requisito não funcional que ainda não foi testado).

Nygard conclui que é melhor evitar tanto a aceitação cega quanto a reversão cega.

Decisão: Registros de Decisões Arquitetônicas (ADRs)

Para resolver esses problemas, o artigo propõe manter uma coleção de registros para decisões "arquitetonicamente significativas". Estas são decisões que afetam a estrutura, características não funcionais, dependências, interfaces ou técnicas de construção do sistema.

Um Registro de Decisão de Arquitetura (ADR - Architecture Decision Record) é um arquivo de texto curto em um formato semelhante a um padrão Alexandrino. Embora as decisões em si não sejam necessariamente padrões, elas compartilham a característica de equilibrar forças. Cada registro descreve um conjunto de forças e uma única decisão em resposta a essas forças. A decisão é a peça central, então forças específicas podem aparecer em múltiplos ADRs.

Os ADRs serão mantidos no repositório do projeto sob `doc/arch/adr-NNN.md` e usarão uma linguagem de formatação de texto leve, como Markdown ou Textile. Eles serão numerados sequencialmente e monotonicamente, e os números não serão reutilizados. Se uma decisão for revertida, o ADR antigo será mantido, mas marcado como "superseded" (substituído), com uma referência ao seu substituto.

O formato proposto para um ADR tem poucas partes para facilitar a digestão:

- **Título:** Frases nominais curtas, como "ADR 1: Implantação no Ruby on Rails 3.0.10".

- **Contexto:** Descreve as forças em jogo (tecnológicas, políticas, sociais, locais do projeto), que provavelmente estão em tensão. A linguagem nesta seção é neutra em valor, descrevendo apenas fatos.
- **Decisão:** Descreve a resposta às forças, declarada em frases completas e voz ativa ("Nós iremos...").
- **Status:** Uma decisão pode ser "proposta", "aceita", "depreciada" ou "superseded" (substituída), com referência ao seu substituto, se for o caso.
- **Consequências:** Descreve o contexto resultante após a aplicação da decisão. Todas as consequências (positivas, negativas e neutras) devem ser listadas, pois afetam a equipe e o projeto no futuro.

O documento inteiro deve ter uma ou duas páginas. Cada ADR deve ser escrito como uma conversa com um futuro desenvolvedor, exigindo um bom estilo de escrita, com frases completas organizadas em parágrafos. Listas são aceitáveis apenas para estilo visual, não como desculpa para fragmentos de frases.

Status e Consequências

O artigo em si é formatado como um ADR, e Nygard relata que o formato foi aceito e tem sido usado em alguns projetos desde o início de agosto de 2011. O feedback inicial de clientes e desenvolvedores tem sido bastante positivo. Desenvolvedores que rotacionaram pelos projetos usando ADRs apreciaram o grau de contexto que receberam ao lê-los.

Os ADRs têm sido especialmente úteis para capturar intenções de longo prazo, como planos de re-arquitetura futura, garantindo que essas intenções não sejam inadvertidamente dificultadas por decisões atuais.

Uma objeção potencial é que manter os ADRs no controle de versão com o código os torna menos acessíveis para gerentes de projeto, stakeholders e outros que não vivem no controle de versão. No entanto, na prática, os projetos da Cognitect usam repositórios privados no GitHub, onde links para a versão mais recente podem ser facilmente compartilhados. Como o GitHub processa Markdown automaticamente, os ADRs são tão amigáveis quanto qualquer página wiki.

Conclusão

Michael Nygard conclui que os ADRs estão se mostrando uma ferramenta útil e que continuarão a ser usados. O artigo "Documenting Architecture Decisions" oferece uma abordagem prática e eficaz para garantir que as decisões arquitetônicas sejam registradas, compreendidas e acessíveis, promovendo a transparência e a sustentabilidade em projetos de software, especialmente em ambientes ágeis. Ao focar em documentos pequenos, modulares e com um formato claro, os ADRs ajudam a preservar o conhecimento e a motivação por trás das escolhas arquitetônicas, evitando a "aceitação cega" ou a "mudança cega" de decisões passadas.

Referências

[1] Nygard, M. (2011, 15 de novembro). *Documenting Architecture Decisions*. Cognitect Blog. Disponível em: <https://cognitect.com/blog/2011/11/15/documenting-architecture-decisions.html>