

19 - Decisões de Arquitetura

2025-12-23 06:49 am

Principal expectativa do arquiteto é tomar decisões de arquitetura

- Pode ou não envolver decisões de tecnologia.

Anti-padrões de Decisões de Arquitetura

Três principais antipadrões:

1. *Covering your assets*
 - Evita decisões por medo de errar
 - Uma forma de minimizar é esperar até o último momento responsável para decidir, esperar o máximo de informações possível.
 - Importante colaborar com o time de desenvolvimento continuamente
 - Colaboram pra dizer o que é ou não possível ou conveniente.
2. *Groundhog Day*
 - Ninguém sabe por que uma decisão foi tomada, então ela é discutida repetidamente.
 - Decisões têm que ter justificção técnica e de negócio
 - Apenas técnica não basta
 - Custo, tempo até mercado, satisfação do usuário e posicionamento estratégico costumam ser boas justificativas.
3. *Email-Driven Architecture*
 - Ninguém lembra ou sabe que houve uma decisão que está perdida entre e-mails.
 - Não é bom incluir decisão em um e-mail, costuma ter a decisão mas sem a justificativa.
 - Melhor apenas linkar o documento contendo a decisão bem explicada.
 - Apenas incluir as pessoas relevantes na comunicação.

Arquiteturalmente Significante

- Decisões de tecnologia podem ser decisões de arquitetura.
- Decisões relevantes são as que afetam:
 - *estrutura*
 - impacta padrões ou estilo de arquitetura (e.g. alteração do *bounded context*)
 - *características não-funcionais*
 - capacidades do sistema (e.g. desempenho)
 - *dependências*
 - Pontos de acoplamento entre componente/serviços
 - *interfaces*
 - Acesso e orquestração dos serviços (e.g. definição de contratos)
 - *técnicas de construção*
 - Alterações nas ferramentas e plataformas acessórias.

Registro de Decisões de Arquitetura

- *Architecture Decision Record (ADR)*

- Pequeno arquivo de texto de uma ou duas páginas especificando uma decisão.
- Cinco seções principais (pode ser diferente):

1. Título

- Frase Descritiva

2. Status

- Proposto, aceito ou substituído.
- Quando é substituído, presume-se que havia sido aceito antes.
 - Refere-se qual ADR fez a substituição e também qual aquela ADR substituiu, se for o caso.
- Pedir comentários também pode ser um Review
- Pode ou não requerer aprovação externa, a depender do impacto e custo.

3. Contexto

- Qual circunstância provocou a decisão?
- Pode incluir alternativas possíveis.

4. Decisão

- Decisão junto com a justificativa

5. Consequências

- Qual o impacto esperado da decisão.
 - Bom e mau.

6. Compliance

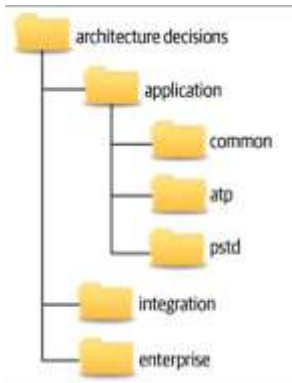
- Opcional
- Como a decisão será medida e gerida.
 - Medição manual subjetiva ou com uma função de score.

7. Anotações

- metadados como autor, datas, modificações, etc.
- Útil mesmo dentro do Git.

Armazenando ADRs

- Cada decisão tem que ter o próprio arquivo separado.
- Uma alternativa é colocar no Github junto com o código.
 - Desvantagens, especialmente em organizações maiores:
 - Nem todo mundo da organização pode ter acesso
 - Não é bom quando existe um contexto fora da aplicação em si (e.g. integrações).
- Melhor ainda é colocar em uma wiki ou um file system compartilhado.



- Dividido em:
 - Aplicação
 - Integração
 - Enterprise
 - Mudanças globais da organização

ADRs como Documentação

- Não há padrões amplamente definidos
- Funciona como documentação da arquitetura.

Usando ADRs para Padrões

- Define e justifica padrões utilizados.
- Permite refletir sobre padrões, se deve ou não ser utilizado.

Exemplo

ADR 76. Asynchronous Pub/Sub Messaging Between Bidding Services

STATUS

Accepted

CONTEXT

The Bid Capture Service, upon receiving a bid from an online bidder or from a live bidder via the auctioneer, must forward that bid onto the Bid Streamer Service and the Bidder Tracker Service. This could be done using asynchronous point-to-point (p2p) messaging, asynchronous publish-and-subscribe (pub/sub) messaging, or REST via the Online Auction API Layer.

DECISION

We will use asynchronous pub/sub messaging between the Bid Capture Service, Bid Streamer Service, and the Bidder Tracker Service.

The Bid Capture Service does not need any information back from the Bid Streamer Service or Bidder Tracker Service.

The Bid Streamer Service must receive bids in the exact order they were accepted by the Bid Capture Service. Using messaging and queues automatically guarantees the bid order for the stream.

Using async pub/sub messaging will increase the performance of the bidding process and allow for extensibility of bidding information.

CONSEQUENCES

We will require clustering and high availability of the message queues.

Internal bid events will be bypassing security checks done in the API layer.

UPDATE: Upon review at the April 14th, 2020 ARB meeting, the ARB decided that this was an acceptable trade-off and no additional security checks would be needed for bid events between these services.

COMPLIANCE

We will use periodic manual code and design reviews to ensure that asynchronous pub/sub messaging is being used between the Bid Capture Service, Bid Streamer Service, and the Bidder Tracker Service.

NOTES

Author: Subashini Nadella

Approved By: ARB Meeting Members, 14 APRIL 2020

Last Updated: 15 APRIL 2020 by Subashini Nadella