

Nome: Raylander Marques Melo Matrícula: 494563

Web site:

Introdução :

Há pessoas que comparam documentação arquitetural com uma torre de marfim, ou seja, apenas consomem tempo e outros recursos preciosos para no final não serem utilizados.

Documentação arquitetural exerce papel importante e é útil desde que feita na medida certa e focada para quem a irá utilizar.

Documentação Arquitetural em Processos Ágeis:

Resistir a documentação arquitetural por achar perda de tempo.

Resistir a práticas arquiteturais, inclusive documentação na medida certa, tem pouco haver com agilidade, mas com teimosia, falta de conhecimento profundo ou imaturidade.

O que o documento arquitetural não é:

Raramente, uma decisão arquitetural, estará associada a uma *feature* especificamente

Especificações exaustivas de requisitos funcionais ou não funcionais, embora importantes para a elaboração da arquitetura, também não são documentos arquiteturais.(Pergunta)

Documentação Arquitetural e a comunicação:

Mais comunicação não é necessariamente uma coisa boa

Documentação arquitetural bem-feita exerce papel importante na comunicação.

Sem uma documentação bem feita corre risco de ocorrer um mal entendimento do sistema e acabar ocasionando problemas na implantação e afins.

Documentação da arquitetura de software e a complexidade:

A complexidade dos projetos de software também pode ser mitigada pela adoção de práticas eficientes de arquitetura de software.

Atividades suportadas pela documentação arquitetural:

Educação, seja para novos membros do time, analistas externos ou até mesmo um novo arquiteto.

Alinhamento com *stakeholders*

Base para a continuidade da construção e análise do software.

Precisa ser mais barato e menos arriscado adicionar features ou fazer ajustes quando há boa documentação arquitetural disponível.

O que geralmente precisa ser documentado:

Deve registrar, primeiro, aspectos que vão permanecer verdadeiros por mais tempo.

Também precisa dar ênfase na estratégia que irá autorizar mudanças

A documentação também deve expor a estrutura atual e suas propriedades.

Considerações sobre “Architecture Haiku”

A provocação é colocar um *overview* da arquitetura em uma única folha de papel A4. Assim, com menos espaços para indecisões ou prolongamentos.

Considerações sobre ADRs:

Se deve documentar decisões arquiteturais em documentos pequenos, podendo ser até em forma de texto, de forma que fique acessível a todos os integrantes da equipe de desenvolvimento.

Considerações sobre o modelo 4+1:

Sua proposta é descrever a arquitetura de sistemas intensivos de software com base em múltiplas visões

O maior problema é que embora seja descrito como um modelo arquitetônico, se afasta consideravelmente dos aspectos centrais dos objetivos de negócio, restrições e atributos de qualidade.

Considerações sobre notações gráficas informais:

O maior problema é que embora seja descrito como um modelo arquitetônico, se afasta consideravelmente dos aspectos centrais dos objetivos de negócio, restrições e atributos de qualidade.

Considerações sobre a utilização de ferramentas de arquitetura corporativa:

Ferramentas destinadas ao registro de arquitetura corporativa, se usadas apenas para o registro de artefatos de arquitetura de software acabam sendo, geralmente, soluções “pesadas”.

Entretanto, não se pode ignorar que o uso adequado abre possibilidades interessantes como, por exemplo, a realização de consultas no repositório arquitetural.

Grandes jornadas são mais fáceis com um mapa...:

Se deve ter um cronologia do que se deve seguir ou o que se já passou para que isso seja possível ser visto ao final do projet.

Vídeo zup:

Hype das linguagens:

Promover novas tecnologias tentando comparar linguagens e linguagens, ou seja, a questão de derrubar outra linguagem para alavancar uma nova.

O hype se retroalimenta deixando alguém como vilão.

Microserviços:

Foi criado para escalar equipes, ou seja, nem sempre usar microserviços nem sempre é uma solução para o sistema.

Crítica falando que não existe apenas monolítico e microserviços.

Arquitetura Java:

Alterando memória disponível no servidor de forma que consiga se utilizar o máximo possível.

Revisando os códigos para melhorar a escalabilidade. Melhorar a transição do sistema com banco.

Não abrir muitas conexões com o banco nem fechar. Usar conexões já abertas e predefinidas para isso, ou seja, configurar o pool conexões de forma certa.

Diminuir a quantidade de idas no banco, mandando o máximo de informações a cada ida.

Usar consultas planejadas, ou seja, faça as suas consultas de forma que elas tragam o máximo de informações possíveis a cada requisição.

Problema de picos de acesso, usar cache.

Usar de threads para executar coisas pesadas em paralelo.

Para grande quantidade de dados se usa uma fila para guardar quem vai executar primeiro, ou seja, ao invés de executar paralelamente ele executa tudo no tempo do servidor.

Escalabilidade vertical, melhorar a máquina, ou seja, muito custoso melhorias.

Cache distribuído

três valores - cache, processamento assíncrono e balanceamento de carga.