

1 - Introdução

Monday, 3 March, 2025

8:14 Manhã

- O que é Arquitetura de Software?
 - Muitas funções
 - Nem sempre claro definir
 - Contém muitas ideias ultrapassadas
 - Contexto muda o que é viável
- Definição
 - Blueprint?
 - Roadmap?
- Quatro componentes da Arquitetura
 - a. Estrutura
 - Estilo geral (e.g. microsserviços)
 - b. Características
 - Critérios de sucesso (e.g. escalabilidade, testabilidade, etc)
 - c. Decisões
 - Regras do sistema (e.g. quais camadas fazem o quê)
 - Pode ter variâncias (exceções)
 - d. Princípios
 - Diretriz
 - não é uma regra (e.g. usar mensageria assíncrona)
- Oito Expectativas
 - a. Tomar decisões de arquitetura
 - Orientas, não especificar, quando der
 - b. Analisar continuamente a arquitetura
 - Verificar se decisões antigas seguem boas
 - c. Assegurar conformidade com decisões
 - Inspeccionar desenvolvimento
 - d. Exposição a experiências diversificadas
 - Amplitude > Profundidade
 - e. Ter conhecimento sobre o domínio do negócio
 - f. Ter habilidade interpessoal
 - Tão importante quanto técnica
 - g. Entender e lidar bem com questões políticas
 - Toda decisão será questionada
- Práticas de engenharia
 - Escala Elástica - gearar mais instâncias de recursos sob demanda
 - Arquitetura != Desenvolvimento (e.g. Agile) - por tradição
 - [XP - Extreme programming --> Continuous Delivery]
- Impossível antecipar incógnita desconhecidas
- "Arquitetura Evolutiva" - fácil de alterar e adaptar
- Avaliação objetiva das características da arquitetura
 - Métricas, testes, etc
- Devops
 - Associação óbvia com arquitetura
 - 1990 a 2000 - terceirizavam operação - arquitetura que limitava operadores
- Processo
 - Processo (como é criado) é independente da arquitetura (estrutura)

- Mas pode não ser completamente
 - E.g. considerar ciclo mais rápido de feedback
 - Migração de arquitetura -> Agile
- Dados
 - DB-Admin trabalham junto de arquiteto
 - Simbiose
- 1ª LEI - Tudo é uma concessão (trade-off)
- 2ª LEI - O porquê vale mais que o como