

23 - Habilidades de Liderança e Negociação

2025-12-24 10:10 am

Negociação e Facilitação

- Necessário entender o clima político da empresa.
- Toda decisão do arquiteto será questionada, tanto tecnicamente pelos desenvolvedores e outros arquitetos quanto stakeholders por restrições de negócio.
- Por exemplo, arquiteto tem que negociar transformar um database único em vários federados por domínio para diminuir riscos de disponibilidade.

Negociando com Stakeholders

- Cenário: Como convencer que uma nível de disponibilidade extremamente alto não é necessário, contrário à opinião do vice-presidente
 - Dificil navegação.
 - "Quero downtime zero" não significa nada de fato mas significa que a disponibilidade é um critério crítico.
 - Arquiteto deve esclarecer o significado das decisões técnicas com dados e informações.
 - Mencionar custo e tempo deve ser só o último passo da negociação, para não melar discussão já de início.
 - As outras justificativas devem vir primeiro.
 - "Dividir para conquistar"
 - Qualifique as necessidades do projeto por pedaço do sistema, reduz o escopo da negociação.

Negociando com Outros Arquitetos

- Cenário: Dois arquitetos discordam se comunicação assíncrona (defendido pelo arquiteto-chefe) ou REST (defendido por outro arquiteto)
 - Apelar para hierarquia é uma má opção, cria ambiente ruim.
 - **Demonstração vale mais que discussão**
 - Ideal é rodar um experimento teste no ambiente de produção.
 - Discussão acalorada e pessoal deve ser sempre evitada.

Negociação com Desenvolvedores

- Arquiteto não deve sair mandando a esmo nos desenvolvedores.
- Desenvolvedores não podem sentir que arquiteto está numa torre de marfim.
- Arquiteto deve sempre justificar e argumentar, não apenas ditar.
- Ideal é que por argumentação e discussão o próprio desenvolvedor chegue à conclusão do arquiteto.
 - Se houver discordância, desenvolvedor deve ter a chance de justificar e defender sua escolha, seja para descartá-la ou reforçá-la.

Arquiteto de Software Como Líder

- Metade do que é ser um bom arquiteto é saber lidar com pessoas e ter habilidades de liderança.

Os Quatro Cs da Arquitetura

- Há problemas que são complexos porque são complexos (*complexidade essencial*) e outros que são complexos porque foram feitos complexos desnecessariamente (*complexidade acidental*)
 - Desenvolvedores são atraídos pela complexidade como mariposas pela luz.
- Complexidade acidental deve ser evitada a todo custo.
- Evita-se isso com os 4 C's de um bom líder e bom time:
 - **Comunicação**
 - **Colaboração**
 - **Clareza**
 - **Concisão**

Seja Pragmático, Mas ainda Visionário

- Planejamento para o futuro mas sem largar a praticidade no presente.
- Pensamento estratégico.
- Orçamento, tempo, limitações técnicas e habilidades do time são restrições a serem consideradas.
- Não basta bolar um plano imenso complexo e mirabolante na teoria.
- Equilíbrio entre os dois não é fácil.

Liderando Times por Exemplo

- Liderança não vem do título.
- Todo problema é um problema que envolve pessoas.
- Conhecimento técnico é só uma parte.
- Tentar achar soluções colaborativamente.
- Linguagem importa, evitar imperativos.
- Pronunciar nomes das pessoas é importante.

- Como e quando apertar as mãos também é um detalhe importante.
- Às vezes é bom transformar demandas em pedidos de favor.
- Arquiteto deve se tornar a pessoa de referência do time.
- Convite para almoços informais para discutir questões no trabalho pode ajudar no relacionamento.

Integrando com o Time de Desenvolvimento

- Arquiteto tem muitas reuniões.
 - Nem toda reunião é necessária
 - Se for só para manter arquiteto informado, anotações bastam.
 - Ou participar só das partes relevantes.
- Ajudar tech lead com responsabilidades e reuniões pode criar senso de time.
- Número de reuniões convocadas pelo arquiteto devem ser o mínimo possível.
 - O que vale mais, a reunião ou o trabalho que os desenvolvedores estariam fazendo?
 - Ser rígido ao cronograma e assunto da reunião.