# Problemas com Monolito e SOA

* Muitos problemas em ambos os paradigmas
* O SOA praticamente não existe mais devido aos seus graves problemas
* Existem muitas razões para a necessidade do surgimente de novas arquiteturas

## Plataforma de Tecnologia Única - Monolito

* Monolito utiliza em todos os seus componentes a mesma tecnologia de desenvolvimento
* Muitas vezes é interessante ter a disposição diversas tecnologias que resolvem problemas diferentes ou atacam o problema de maneira mais eficiente
* Upgrade é um problema, já que qualquer atualização impacta a aplicação por completo, forçando a testar o sistemas por completo, tornando o processo muito mais demorado

## Deploy inflexível - Monolito

* Não é possível fazer deploy de apenas uma parte da aplicação
* Mesmo que apenas uma parte da aplicação seja atualizada, todos os componentes precisam ser compilados e deployados, forçando também a execução de testes rigorosos

## Recursos Computacionais Ineficientes - Monolito

* Recursos são compartilhados por todos os componentes da aplicação
* Não é possível aumentar os recursos apenas para um componente específico

## Grande e Complexo - Monolito

* Base de código extensa e complexa, com muitas dependências, difícil de manter, difícil de testar
* Pequenas alterações podem impactar a aplicação inteira

## Complicado e Caro ESB – SOA (Service Oriented Architecture)

* ESB significa Enterprise Service Bus, é um dos principais componentes do SOA, é responsável por fazer a comunicação do cliente para os serviços
* ESB são muito complexos e muito caros para se manter
* ESB tenta resolver tudo, isso é ruim pq não faz nada bem

Linha do tempo

Descrição gerada automaticamente com confiança média

## Falta de Ferramentas - SOA

* Parar o SOA ser eficiente, deve ter ciclos de desenvolvimento curto, já que o monolitos tem como um dos seus principais problemas os longos ciclos
* A ideia era permitir testes e deploys mais rápidos, porém sem ferramentas de automatização, ainda tomavam muito tempo
* A necessidade de realizar testes entre serviços tornou os testes ainda mais complexos e mais demorados do que em monolitos
* Ou seja, não houve um ganho de tempo