

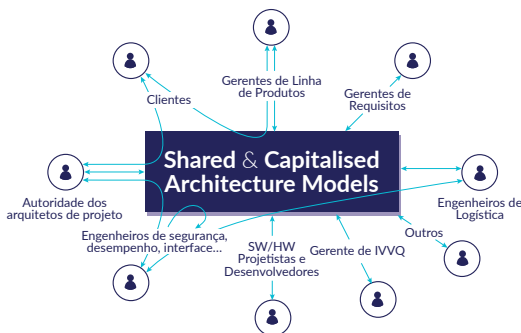


eclipse.org/capella

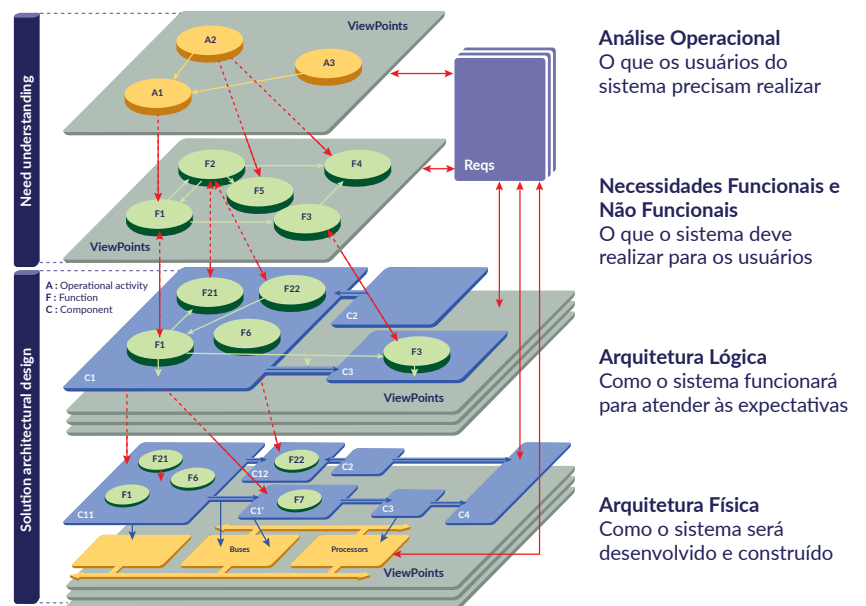
UM MÉTODO DE FERRAMENTAS PARA

# Definir, Analisar, Projetar e Validar arquiteturas de sistema, de hardware e de software

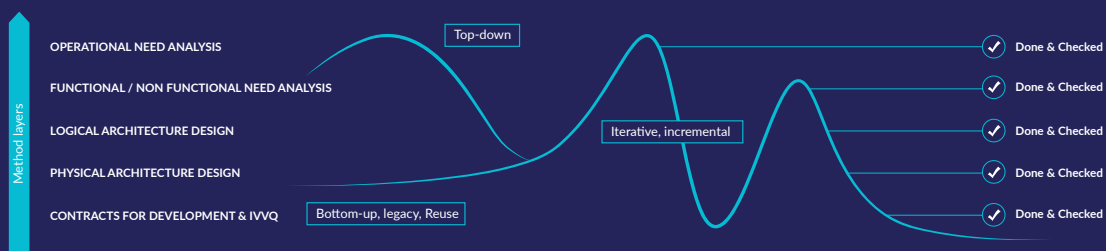
Apoiando a  
**Colaboração Eficiente**  
em Engenharia



Validando/Justificando  
solução em relação à **Necessidade Operacional**  
facilitando **Análises de Impacto**



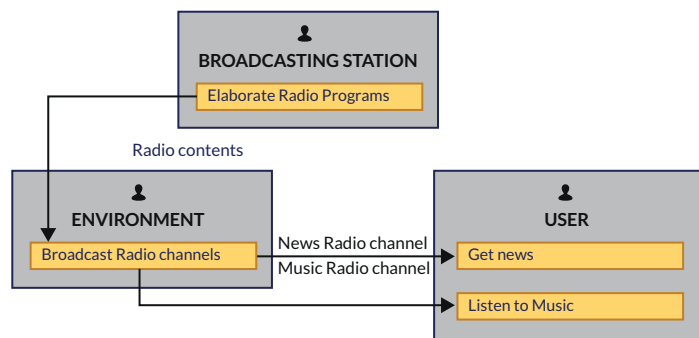
Compatível com a maioria dos processos  
top-down, bottom-up, iterativo, legacy-based, misto...



## Análise das Necessidades Operacionais do cliente

o que os usuários do sistema precisam realizar

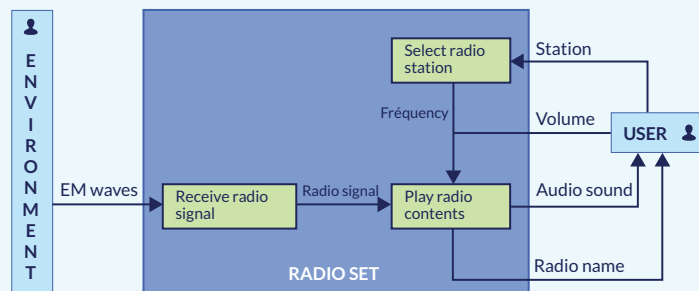
- ✓ Definir capacidades operacionais
- ✓ Realizar uma análise operacional de necessidades



## Análise de Necessidades do Sistema/ HW/SW

o que o sistema precisa realizar para os usuários

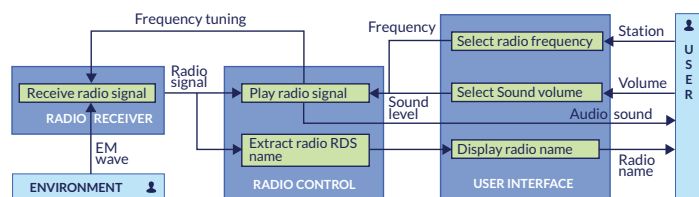
- ✓ Realizar a análise de capacidades
- ✓ Realizar análise funcional e não funcional
- ✓ Formalizar e consolidar os requisitos



## Arquitetura Lógica

Como o sistema funcionará para atender às expectativas

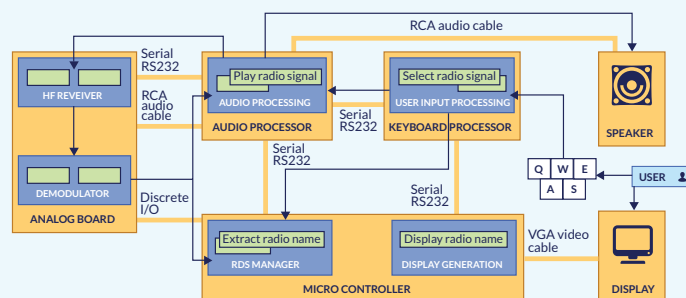
- ✓ Definir drivers de arquitetura e viewpoints
- ✓ Detalhar possíveis arquiteturas em componentes
- ✓ Selecionar a melhor arquitetura



## Arquitetura Física

Como o sistema será desenvolvido e construído

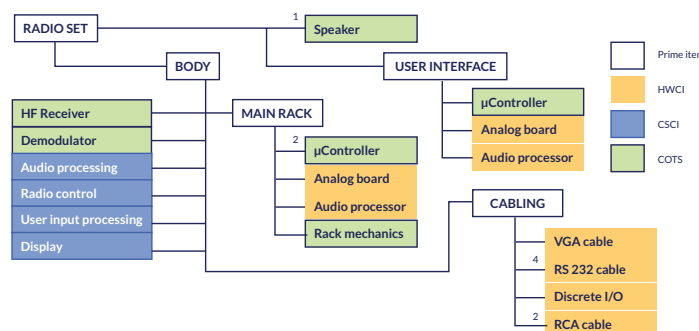
- ✓ Definir padrões de arquitetura
- ✓ Considerar a reutilização de elementos existentes
- ✓ Projetar uma arquitetura física de referência
- ✓ Validar e Verificar



## Contratos de Desenvolvimento

o que se espera de cada projetista/subcontratado

- ✓ Definir uma estratégia de componentes IVVQ (Integrar,
- ✓ Definir e impor um contrato de integração de componentes e PBS



- Capacidades Operacionais
- Atores, entidades operacionais
- Atividades de atores
- Interações entre atividades e atores
- Informações usadas em atividades e interações
- Atividades de encadeamento de processos operacionais
- Cenários para comportamentos dinâmicos

- Atores e sistema, capacidades
- Funções do sistema e atores
- Fluxo de dados de trocas entre funções
- Fluxo de dados de cadeias funcionais
- Informações usadas em funções e trocas, modelo de dados
- Cenários para comportamentos dinâmicos
- Modos e Estados

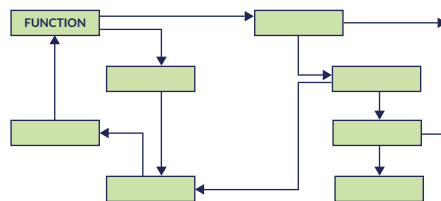
## MESMOS CONCEITOS DAS OUTRAS ETAPAS, ALÉM DE:

- Componentes
- Portas de componentes e interfaces
- Trocas entre componentes
- Alocação de funções em componentes
- Justificativa da interface de componente através da alocação de trocas funcionais

## SAME CONCEPTS, PLUS :

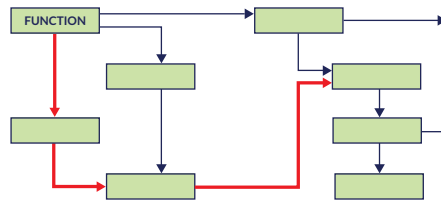
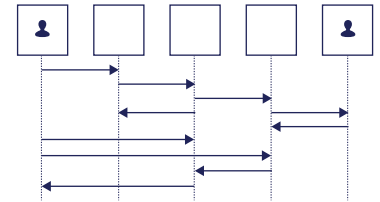
- Componentes comportamentais refinando componentes lógicos, implementando comportamento funcional
- Componentes de implementação fornecendo recursos para componentes comportamentais
- Links físicos entre os componentes de implementação

- Árvore de configuração de itens
- Números de peças, quantidades
- Contrato de desenvolvimento (comportamento esperado, interfaces, cenários, consumo de recursos, propriedades não funcionais...)

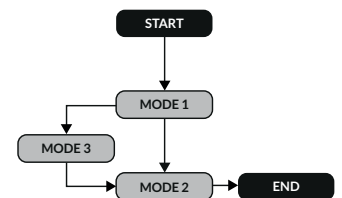
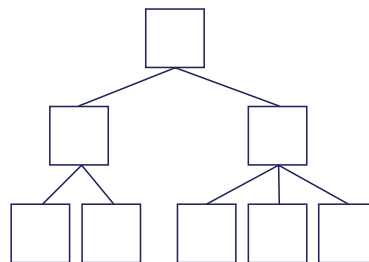


**Fluxo de dados:** funções, interações e trocas de atividades operacionais

**Cenários:** atores, sistema, interações e trocas de componentes



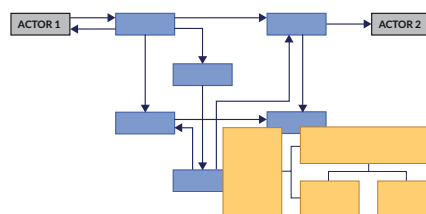
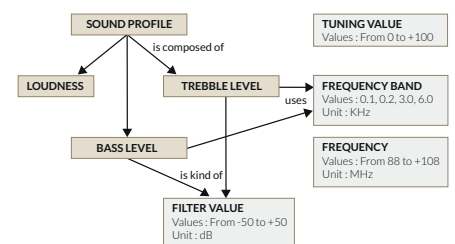
**Cadeias funcionais,** processos operacionais por meio de funções e atividades operacionais



**Modos e Estados** de atores, sistema, componentes

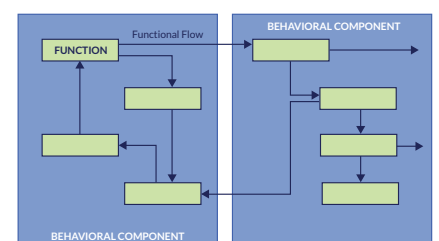
**Diagrama de detalhamento** de funções e componentes

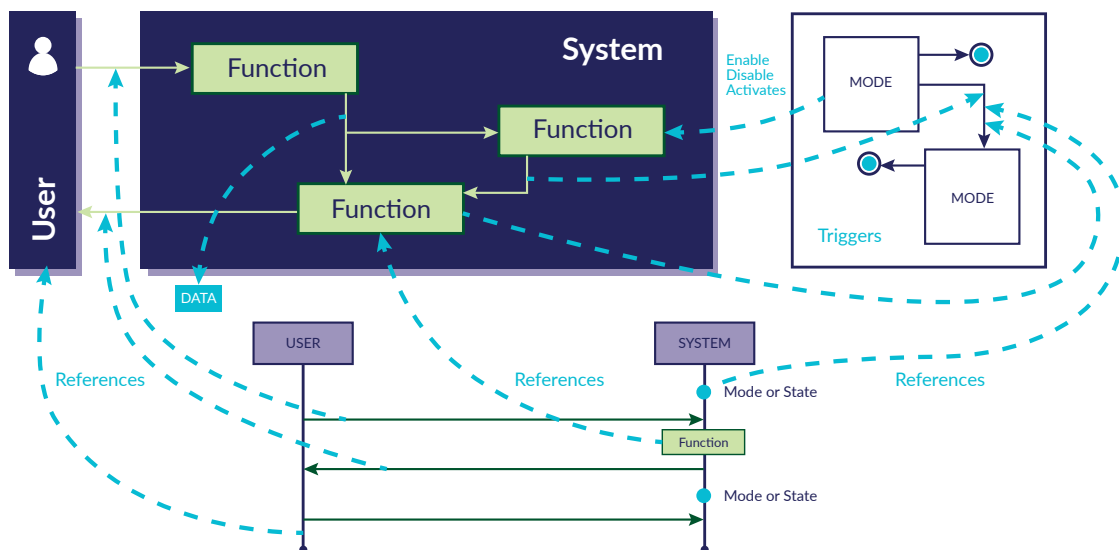
**Modelo de dados:** fluxo de dados e conteúdos sobre cenários, definição e justificativa de interfaces



**Rede de componentes:** todos os tipos de componentes

**Alocação** de atividades operacionais para atores, de funções para componentes, de componentes comportamentais para implementação de componentes, de fluxos de dados para interfaces, de elementos para itens de configuração





## Verificação e validação da solução em relação às ocorrências não funcionais e industriais

Etapas do método	Exemplo de dados específicos de desempenho	Exemplo de dados específicos de segurança
ANÁLISE DE NECESSIDADES OPERACIONAIS	Tempo máximo de reação à ameaça	Eventos indesejados
ANÁLISE DE NECESSIDADES FUNCIONAIS E NÃO FUNCIONAIS	Cadeia funcional (CF) para reagir à ameaça. Latência máxima permitida em CF	Cadeias funcionais críticas associadas aos eventos
PROJETO DE ARQUITETURA LÓGICA	Complexidade de processamento e troca. Alocação de cadeias funcionais	Caminhos de redundância protegendo cadeias funcionais
PROJETO DE ARQUITETURA FÍSICA	Consumo de recursos na CF. Latência computacional resultante	Modos de falha comuns. Propagação de falhas em CF
CONTRATOS DE DESENVOLVIMENTO E IVVQ	Recursos alocados para satisfazer a latência	Nível de confiabilidade necessário

- ✓ Custos e Cronograma
- ✓ Interfaces
- ✓ Desempenho

- ✓ Manutenibilidade
- ✓ Segurança/Proteção
- ✓ ...

- ✓ IVVQ
- ✓ Política do produto

