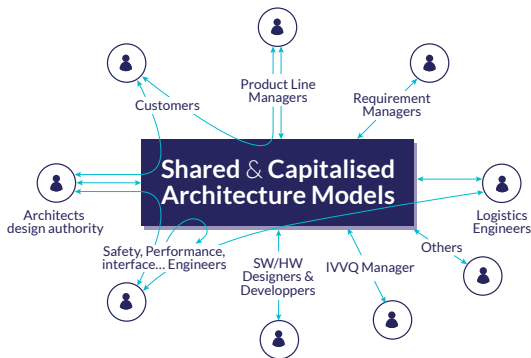


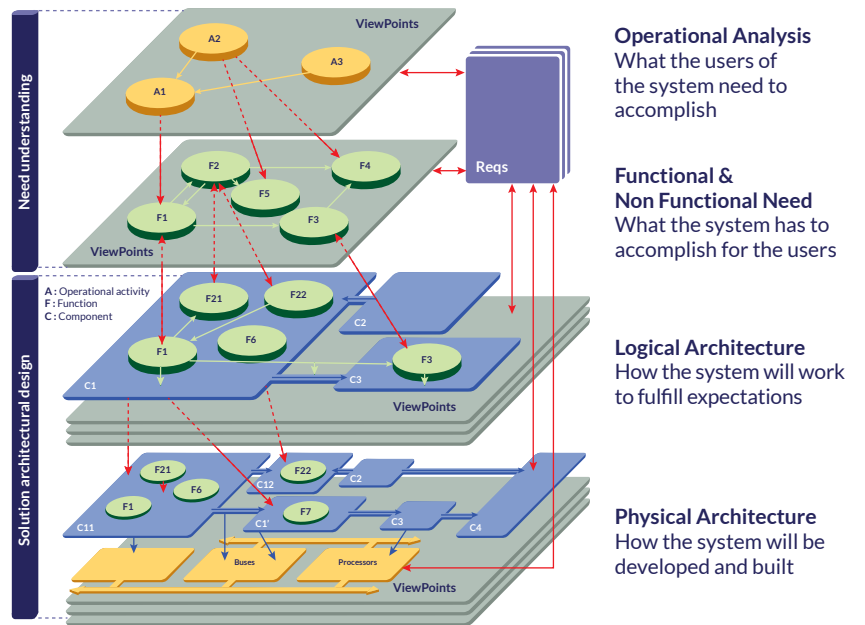
A TOOLED METHOD TO

Define, Analyse, Design & Validate System, Software, Hardware Architectures

Supporting Efficient Collaboration in Engineering



Validating/Justifying solution against Operational Need Easing Impact Analysis



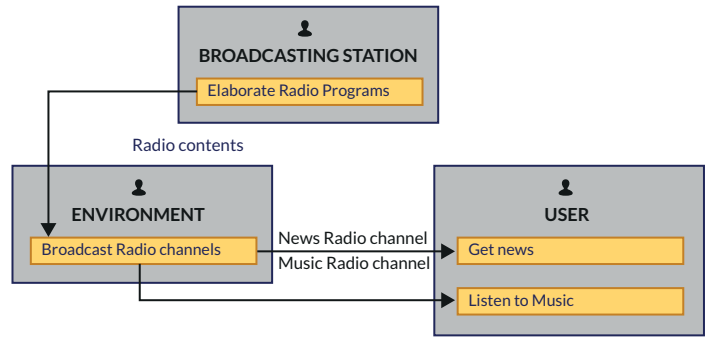
Compatible with most processes top-down bottom-up, iterative, legacy-based, mixed ...



Customer Operational Need Analysis

What the users of the system need to accomplish

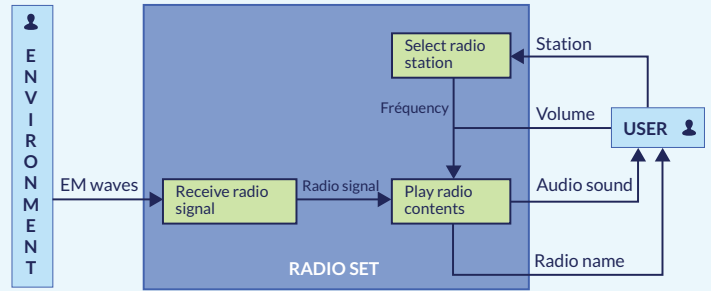
- ✓ Define operational capabilities
- ✓ Perform an operational need analysis



System/SW/HW Need Analysis

What the system has to accomplish for the Users

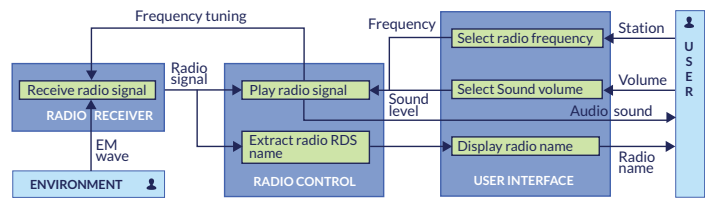
- ✓ Perform a capability trade-off analysis
- ✓ Perform a functional and non-functional analysis
- ✓ Formalise and consolidate requirements



Logical Architecture Design

How the system will work so as to fulfil expectations

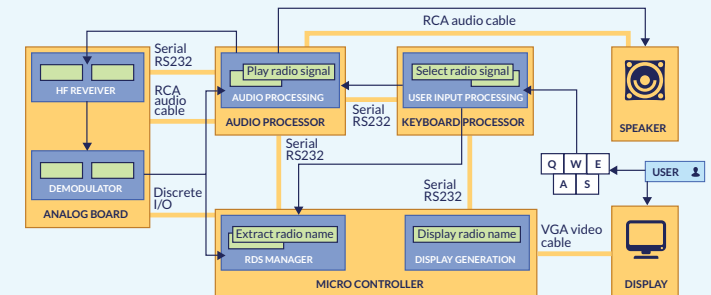
- ✓ Define architecture drivers and viewpoints
- ✓ Build candidate architectural breakdowns in components
- ✓ Select best compromise architecture



Physical Architecture Design

How the system will be developed & built

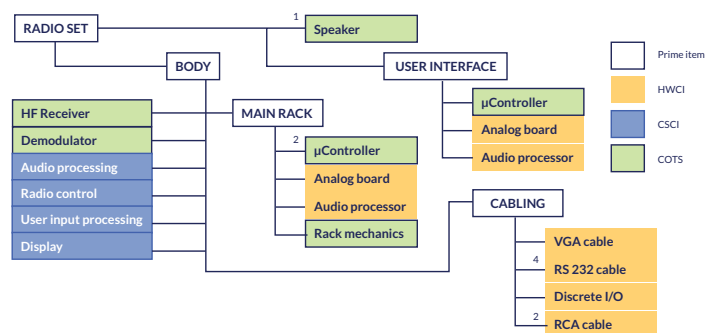
- ✓ Define architectural patterns
- ✓ Consider reuse of existing assets design a physical
- ✓ Design a physical reference architecture
- ✓ Validate and check it



Development Contracts

What is expected from each designer/sub-contractor

- ✓ Define a components IVVQ strategy
- ✓ Define & enforce a PBS and component integration contract



- Operational capabilities
- Actors, operational entities
- Actor activities
- Interactions between activities & actors
- Information used in activities & interactions
- Operational processes chaining activities
- Scenarios for dynamic behaviour

- Actors and system, capabilities
- Functions of system & actors
- Dataflow exchanges between functions
- Functional chains traversing dataflow
- Information used in functions & exchanges, data model
- Scenarios for dynamic behaviour
- Modes & states

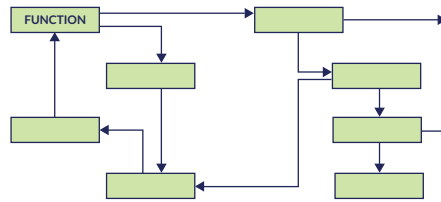
SAME CONCEPTS, PLUS :

- Components
- Component ports and interfaces
- Exchanges between components
- Function allocation to components
- Component interface justification by functional exchanges allocation

SAME CONCEPTS, PLUS :

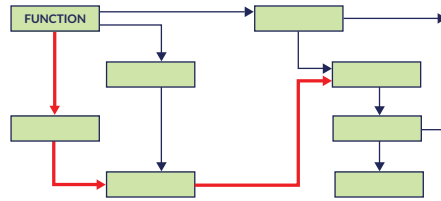
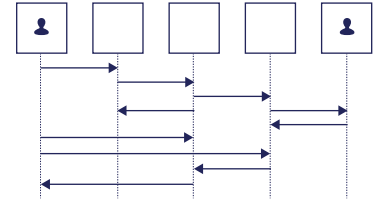
- Behavioural components refining logical ones, and implementing functional behaviour
- Implementation components supplying resources for behavioural components
- Physical links between implementation components

- Configuration items tree
- Parts numbers, quantities
- Development contract (expected behaviour, interfaces, scenarios, resource consumption, non-functional properties...)

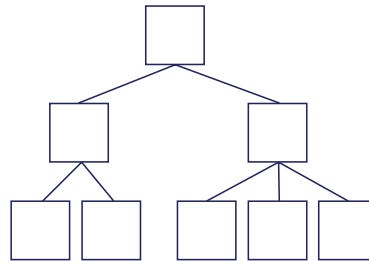


Dataflow: functions, op. activities interactions & exchanges

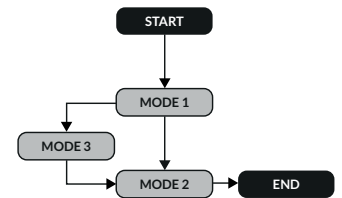
Scenarios: actors, system, components interactions & exchanges



Functional chains, operational processes through functions & op. activities

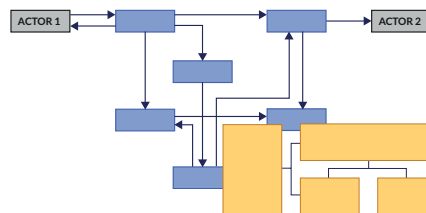
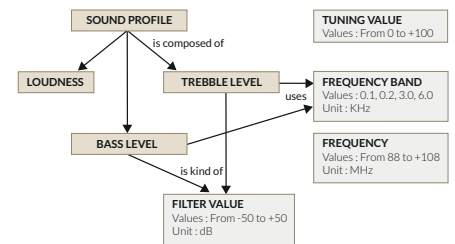


Breakdown of functions & components



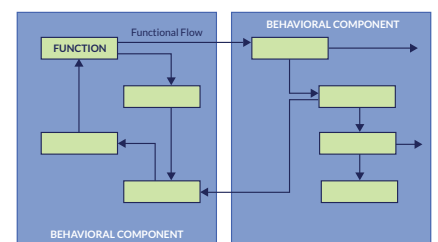
Modes & states of actors, system, components

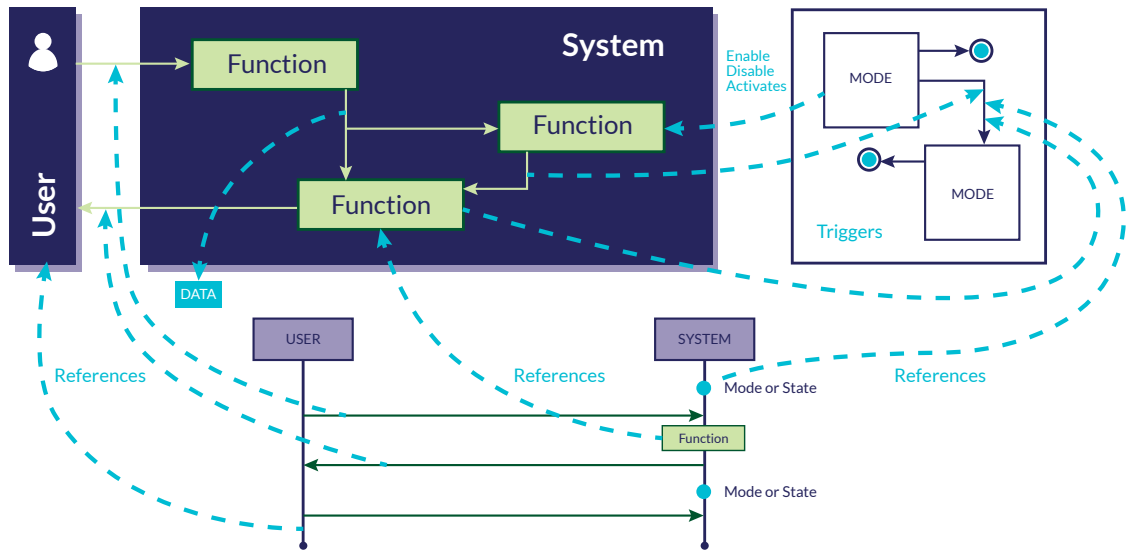
Data model: dataflow & scenario contents, definition & justification of interfaces



Component wiring: all kinds of components

Allocation of op. activities to actors, of functions to components, of behav. components to impl. components, of dataflows to interfaces, of elements to configuration items





Verifying & checking solution against Non-functional & Industrial Stakes

Method layers	Performance specific data sample	Safety specific data sample
OPERATIONAL NEED ANALYSIS	Max reaction time to threat	Feared events
FUNCTIONAL/NON FUNCTIONAL NEED ANALYSIS	Functional chain (FC) to react to threat. Maximum allowed latency on FC	Critical functional chains associated to events
LOGICAL ARCHITECTURE DESIGN	Processing & exchanges complexity Functional chains allocation	Redundancy paths securing functional chains
PHYSICAL ARCHITECTURE DESIGN	Resource consumption on FC Resulting computing latency	Common failure modes Fault propagation on FC
CONTRACTS FOR DEVELOPMENT & IVVQ	Allocated resources to satisfy latency	Needed reliability level

- ✓ Cost & Schedule
- ✓ Interfaces
- ✓ Performance

- ✓ Maintainability
- ✓ Safety/security
- ✓ ...

- ✓ IVVQ
- ✓ Product Policy

