



From the White Paper "The Role of Systems Architect" by the Cercle CESAM - Article 10:
Manage Functional and Physical Interfaces, Internal and External*

|

Preamble

Along with the definition of black box and white box architectures, the architect must identify the interfaces, external and internal (with external systems, between system functions and between its components).

Just as he must ensure that each function and each component is held accountable by a role, he must also put under control all the interfaces identified.

THE ESSENTIAL

The complexity of a system is also linked to the number of its interfaces which can condition the number of stakeholders to be involved. Interface management (human, physical, software, etc.) is a key activity in risk control on a project. In the black box part, the definition of the interfaces is a major element in the formalization of the scope of study and the integration of the system into its environment.

THE MAIN PITFALLS

Among the main pitfalls:

- Identify the external interfaces without asking the question of the need to be covered and go too quickly into technical detail (e.g. wanting to finely define the types of data to be transmitted). This is often amplified by a desire for reuse that is poorly framed in relation to the new context/need
- Not monitoring the maturity of interfaces, i.e. not managing an interface because it is still immature (this puts people on hold and generates potential delays) and/or

not anticipating the fact that the definition of certain interfaces will evolve (which generates a redesign perceived as useless)

- Update the interfaces without informing the stakeholders affected by this modification
- Failing to identify responsibility for an interface
- Poor management of the configuration of the two organs on either side during the integration of the interfaces
- Interfaces well defined on specific aspects (ex: static mechanical functioning) which a priori allow integration but which are finally poorly defined on other aspects (ex: dynamic functioning)
- Diverting interfaces (in the context of reuse) and risking deviation from the nominal design of the interface (with associated stakes, which would be more expensive than defining the correct interface)
- A poor definition of external interfaces leading to an ambiguity on the perimeter (ex: the interface to authenticate the user: the client considers that he is inside and the software designer considers that he is outside its perimeter)

BEST PRACTICES

Some good practices to consider:

- Use architecture to identify interfaces
- Ensure that each interface is well supported (in responsibility)
- Set up an interface maturity management process
- In the context of reuse, the question of the need to be answered with each interface
- Put polarizers (Poka-Yoke) on physically identical but functionally different interfaces (ex: polarizer on empty wall outlets and oxygen in hospital rooms)
- Define all the characteristics of the interface (the nominal and the robustness at the output of the nominal)
- Use MBSE to define all interfaces as comprehensively as possible
- Standardize the way to define interfaces
- Rely on the mastery of interfaces to optimize interactions with stakeholders (e.g. the management of several interfaces with the same stakeholder can increase the communalization between these interfaces)

TESTIMONIALS

We have compiled here several verbatim statements from project managers or system architects from different companies, which echo this phase:

- “ *We have set up a process for managing the maturity of interfaces, in particular for monitoring mechanical interfaces. Each interface document has a defined and displayed maturity level, which makes it possible to share the risks associated with each interface.*
- “ *On the software systems, we proceed with an early integration rather than a gradual increase in maturity of the interface. (Be careful, however, that this practice is not duplicated by the pitfall mentioned above on questioning the need to be covered).*
- “ *For the hardware: we are rather on the implementation of a virtual integration stage (integration of digital business models - 3D, hydraulics, electrical, etc.) to anticipate integration problems*
- “ *Implementation of a tooling management of interfaces using the CESAM framework in Xatis (Safran)*

***The Cercle CESAM** is a working group whose objective is to develop and share a pragmatic international system architecture standard and to apply it to each major industrial field for the commercial benefit of CESAMES's community members.

Next article 11 is expected Friday July the 7th, 2023

All articles of “The Role of Systems Architect” Serie can be found and downloaded on this CESAMES's webpage:

<https://cesames.cn/fridayshares/>