SDP / PDS WORKING GROUP

STRUCTURING PROJECT DATA

PLANNING DIGITALIZATION

WHITE PAPER

S. BOUTIN G. TURRÉ

29/09/2023 – V 18

1 MOTIVATION

This work of the SMaP aimed initially at improving project data hygiene by providing a clean structuring of project data to allow relevant project data analysis including using artificial intelligence.

The working group identified shortcomings in the State-of-the-Art structuring of project data, and decided to redefine several classic concepts and to propose new ones. The work is based on doctrines underlying Project Management and Cost Engineering (as defined in PMI, IPMA, APM, AACE) but the improvement of concepts takes its roots in Systems Engineering.

The general objective behind this new approach is to:

- Help build a project planning much faster than before – 1 to 2 orders of magnitude.
- Keep formal traceability between Project planning and Project initial requirements
- Be able to compare different revisions and variants of a project and as a consequence improve capability to capitalize on project planning practices
- Help replanning of project when significant changes occur
- Provide a solid background for data analysis over reliable project information.

To support these objectives, we propose a "Manifesto" with the main changes of paradigm we think necessary to get that project information become formal and related to each other with clearly defined rules.

The “Commentary” chapter explains each of these concepts with more details.

A final chapter addresses a list of topics that are not described in detail in this paper but need to be considered for a complete overview of project engineering.

2 MANIFESTO

21- The project plan is issued from a product-tree and a set of standard activities and not from a work breakdown structure (WBS). (*)

22- The project product-tree is built accordingly with functional decomposition and systems engineering rules and not from a geometric breakdown. (**)

23- Each specific project activity is defined by instantiating a predefined standard activity on a project product, and not freely or by brainstorming to complete an ad hoc work breakdown structure(WBS).

24- Standard activities relate to combinations of life cycle processes and job functions and don't need to be a very complex and thorough description of company legacy processes provided by each job function. (***)

25- The logic of the project is formalized by an order on the project reviews and connecting product life cycles to project level reviews.

26- A planning is generated top-down from consolidated estimates at the level of each product of the project product-tree, and not bottom-up by gathering the task analysis of various work packages and trying to connect their interfaces. (****)

27- The work packages and the assignment of responsibilities are defined by grouping the activities already instantiated upstream and not as a pre-allocation of scope of work. (*****)

(*) In this way, the complexity to manage for the development of activities is two orders of magnitude lower than the manual development of the activities of a schedule.

(**) Both may partially match, but a subproduct shall always functionally contribute to the functions of its parent product.

(***) This allows systems life cycle processes to become the language for the description of project product life cycles and is also an opportunity of using Systems Engineering as a lever between project and engineering. Project does not need to understand thorough details of the engineering processes except for specific and isolated risks or costs relevant activities. SE life cycles are the answer.

(****) Automation and mastery of complexity allow the use of analytical estimates from analog or parametric hypotheses very early on, which would make them usable very early and would allow the first estimates to be challenging effortlessly.

(*****) The instantiated activities can pre-exist the creation of the WBS. They are naturally accessible by product of the project and by job function or supplier. Workpackage interfaces can thus be deduced from underlying activities interfaces as soon as they are created.

3 CONCEPTS REVISITED

31 Product and project product-tree

311- A product is an object whose performance and conformity are guaranteed by an organization on which systems life cycle processes do apply. So any product may be in one of the following maturity states: as needed, as specified, as architectured, as designed, as implemented/produced/built/integrated, as verified, as validated, as qualified, as transferred, as installed, as operated, as maintained, as retired.

312- The project product-tree is built according to the functional hierarchy as defined by systems engineering. A product functions is the result of interactions between all subproducts and their related functions.

313- The project product-tree is established top-down in early project steps and then progressively disclosed according to the needs of cost and planning estimates, and risk assessment.

314- The project product-tree includes all products on which the project team considers there will be work to be performed in order to reach the goal of the project.

315- The project product-tree includes products that may even not be delivered or may even not be under the responsibility of the project.

316- No organizational interface can exist without a product. Any transfer of items or documents between organizations relates to one or several products.

317- The project itself is considered as product and is the root of the project product-tree.

32 Standard activities and systems life cycle processes.

321- Standard activities are processes that are applied by an organization. Standard activities are continuously developed and improved by the business lines upon lessons learned.

322- Standard activities may not be formalized by an organization although they exist anyway explicitly or implicitly with a some maturity level. Maturity models like CMMi or SPICE describe best practices for standard set of processes that shall be formalized when reaching level around 3. Whatever Standard Activities are formalized or not or mature or not in an organization does not prevent from applying this white book.

323- For the purpose of project engineering, it is enough to consider standard activities are a subset of systems life cycle processes together with a competence (or skill or job function or resource kind) applied to it.

324- Systems life cycle processes are described in norm ISO 15288. We add one life cycle to this norm: Installation. The norm ISO 15288 is very focused on the design of innovative complex systems and software while many industrial domains rely mostly on integrating off the shelf products for which roughly only installation is required. We think this standard activity is not well enough captured by the Transition process.

325- Among system life cycle processes as defined in ISO 15288, project management activities are those applied on the project itself, the root of the project product. Supply processes to the customer are also applied at this level.

326- Among system life cycle processes as defined in ISO 15288, acquisition process may be applied on any product in the project product-tree.

327- Acquisition process means there is a transfer of responsibility to another organization. There shall be no confusion here with resource capacity subcontracting where the subcontracting organization keeps responsibility for the underlying product.

328- Standard activities bring their standard inputs and outputs as well as precedence constraints with other standard activities. This true as well for system life cycle processes for which ISO 15289 provides a set of standard inputs and outputs.

33 Project Activities

331- Any activity of a project is built through instantiation of a standard activity on a project product.

332- Conversely, there is no activity without an underlying product. Any activity has its root in a decomposition into products to which we apply one or several subsequent system life cycle process(es).

333- Instantiation applies also to inputs and outputs of standard activities resulting in inputs and outputs for Project Activities.

324- Project management activities are defined directly at the root of the project product-tree that represents the project viewed as a product.

325-The work breakdown structure (WBS) is an organization of project activities. It is intended to define responsibilities and accountability over sets of activities also named work packages. Although work breakdown structure and work packages could be defined at an early step, their exact content shall be defined after project activities allocation and so after project activities definition through instantiation.

34 Milestones and reviews

341- The general logic of the project is formalized by the order of the project reviews and their relationship to products activities.

342-. System life cycle processes come along with reviews allowing to decide the maturity level of a product.

343- The detail of the project logic is formalized by the order of the activities related to each and every product.

35 Information for lessons learned and benchmarks

351- Standard data is stable data that applies permanently to all projects of an organization.

352- It relates firstly to standard activities and then possibly also to standard activities as related to competences and possibly related to products. Possibly this can also relate to organizations performance when performing activities.

353- Systems engineering provides other opportunity for classification and analysis. For instance, functions realized by products form another classification criteria for comparing different projects or scope of projects.

354-. Among other things, lessons learned shall be captured in a baseline for estimating planning, costs, and assessing risks related to projects.

4 COMMENTS

4.1 Products

4.1.1 Product, product hierarchy and life cycle.

A product is a **concrete object**, tangible or service, with performance and conformity **guaranteed by an organization** and having a **life cycle.**
A product can be composed of (sub)products.
A product has a certain **stable purpose** over time.
In the sense of systems engineering, a product, as we mean it in this document, is a **system**, which means that:

- The product ensures a set of **functions**.
- The components of the product (or sub-products) provide **sub-functions necessary to perform the product functions**
- The product has a **life cycle**, made up of successive system life cycle processes as defined e.g. in ISO 15288.

A simpler and intuitive definition of life cycle can be that a product is going through different phases: it is created, it is operated and maintained and at some stage, it is retired.
After all, in order to say something is a product, you need to ask you two questions: is there someone or some organization responsible for it? Does it have a life cycle?

According to ISO 9001, a product may be an equipment, material, a software, or a service. We extend this definition by saying a product may also be an group of persons performing a set of functions or a combination of all previous items.

We will provide several example of what is a product and what is not a product in the subsequent sections.

NB: The term product is overloaded and this is a major source of confusion. In ISO 9001, a product is the “result of a process” which is a considerably more permissive definition than the one we use. More or less everything is the output of some process depending on the viewpoint you take so everything can turn to be a product. We consider a more abstract definition of the word “product” associating products with a life cycle as described in next section.

We will come back to the link between process (standard activities) and products in the “Instantiating” section.

NB: There are two specific characteristics for a product existence in our definition: first it is the existence of a guarantee of quality issued by a legal entity or part of a legal entity. Second is the existence of a life cycle.

NB: The fact of asking for a guarantee shows that the product we are talking about is a “final” product, that is to say, pursuing a certain stable purpose over time. This remark will be clarified in a further paragraph explaining the difference between product and supply. The existence of a life cycle is another way to express that a product has a stable purpose over time.

NB. We will speak of **internal product** when a sub-organization delivers internally to a company a product which will not have the same level of guarantee as a purchase from a legal entity but which is based on an internal quality commitment to the company.

4.1.2 Project (root) product

By convention, the root of the project product-tree corresponds to the finality of the project and will be considered as a Product. We call it the project root product or project product-tree poot or simply project product.

From a practical point of view, the products which will be relevant for the end customer are often first level products under the root. But it's possible root itself be something more than an aggregate. Project often have complex objectives and the project outputs are seldomly a single item or document but rather a set of various items. We give some examples in section 4.1.4.

4.1.3 Product life cycle

System life cycle is a concept from systems engineering. It says that any system made by human is going through a set of maturity gates that correspond each to application of so called system life cycle processes. The system life cycle processes are described in the norm ISO 15288.

Any system/ product may be in one of the following maturity states: as needed, as specified, as architectured, as designed, as implemented/produced/built/integrated, as verified, as validated, as qualified, as transferred, as installed, as operated, as maintained, as retired.

These maturity states can be reached after applying one system life cycle processes among: operational and needs analysis, specification, architecture, design or definition, implementation, integration, verification, transition, validation, operation, maintenance, disposal. We may add qualification and Installation as two life cycle processes that may be needed although not described in ISO 15288:2015.

From the system life cycle processes, we define a list of standard outputs corresponding to each process in the table below:

Applied Life Cycle Process	**Life Cycle Process output**
Operation and Needs analysis	Product as needed
Specification	Product as specified
Architecture	Product as architectured
Design / Definition	Product as defined
Implementation/Integration	Product as built
Verified	Product as verified
Transition	Product as transfered
Validation	Product as validated
Operation	Product as operated
Maintenance	
Disposal	Product as retired
Qualification	Product as qualified
Installation	Product as Installed

In the table above, the "product as needed" may be called a need specification

The “product as specified” may include a technical specification, the related needs specification from previous step, a traceability matrix between needs and requirements from technical specification.
So the output we describe for each process is in fact a set of documents and items ensuring traceability with prior steps and consistent with the level of maturity of the product in its life cycle.

Although you could replace all outputs “product as …” by a set of documents as defined in ISO 15289 or ECSS, this simplified and abstracted set of outputs can be use at first for describing inputs and outputs of each process.

The outputs of system life cycle processes may be called process outputs, simply outputs, or also artefacts. An artefact usually means some intermediate result related to a maturity gate. These different outputs can be seen as an intermediate state reached by the product in its life cycle
It’s very important to note that the concept of product is much more general than the concept of artefact and that a product shall never be reduced to one of its related artefacts. For instance, the delivered building in the project is not the same as the building in the project product-tree. The concept of building in the project product-tree relates to all the life cycle of the building and not only to the particular solution issued at the end of the project.

It is very easy to make a confusion between a “product in the project product-tree” and a “product as transferred” and delivered to the project client. Transfer of some products to the Client is part of the goal of the project and the ultimate goal behind the life cycle. However, this kind of confusion between product and activities outputs shall be avoided for the sake of consistency of the whole approach we present.

4.1.4 Project product-tree.

The project product is the root of the **project product-tree**. Indeed, below, appears a tree structure of all other products in the project scope,

The project product-tree contains all the products on which the project thinks that there will be activities to carry out and only those products.

The products may be classified according to various criteria:

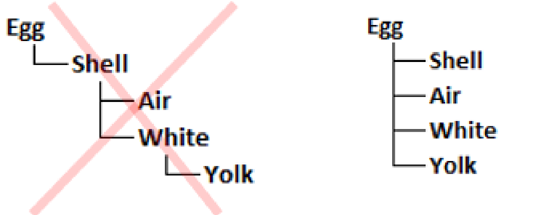
- By responsibility: products in the project product-tree may be under the responsibility of the project but you may find as well products under the responsibility of the Client, of a Stakeholder or of the Company hosting the product. For instance, the client may grant to the supplier some land or an access to the information system.
- By availability: some products are needed temporarily while others will permanently remain part of the project.
- By systems engineering classification between: system of interest and enabling systems. An enabling system supports a system-of-interest during its life cycle stages but does not necessarily contribute directly to its function during operation For instance, if the objective of the project is to produce a ship, the plant and tools needed to built the ship may be part of the project product-tree.
- By removed versus added products: removed earth for a house construction will be considered a product. A product is not necessarily something new or an existing item you must modify, it also includes removed entities.
- By kind: equipment, materials, software, service.

Products are organized in a tree according to systems engineering principles and for the sake of managing complexity.

We consider two types of hierarchies:

- **Functional hierarchy.** If a product contains sub-products, it means the sub-products contribute to the functions of the product, and so on if several hierarchical levels are necessary.

  - Be careful not to make a confusion between functional hierarchy and geometric encapsulation. Example of the egg: the shell contains the white and air, the white contains the yolk; but white does not contribute to shell functions! So, functionally the egg product/system directly contains 4 sub-products: Shell, Air, White, Yolk. Of course, these four sub-systems have many interfaces. So in the egg example, system layout is different from geometric layout.



3D or Geometric tree versus Functional tree

  - Another classical confusing case is the one of the Electronic Control Unit or more generally any computing unit. Software and hardware are two sub-products of the computer. Here again it's tempting to say that the hardware "contains" the software. Just because software is a memory configuration geometrically inside the hardware doesn't mean it's not a product on the same level as the hardware. From a functional point of view. Hardware can run multiple software, and software can run on multiple different platforms. In order to understand the behavior of an Electronic Control Unit, you have to think of the hardware and the software (in the meaning of a memory configuration) interacting. On the other hand, it is not the hardware development team that specifies needs to the software and conversely. This shows their independence and their own integrity. Here we will talk about the interface between the two. Moreover, in general, software and hardware are often delivered and guaranteed by different organizations.

- **Organizational hierarchy**. When many products are present at the same hierarchical level, they can be grouped arbitrarily into subsets. Those organizational groupings that are essential for mastery of information will not be supported at instantiation. It is an ease of reading and organization for the user.

4.1.5 Examples of project product-trees

For manufactured products, a project often aims to deliver an operational production line. For example, a vehicle project will aim to deliver a production line at 1,000 units/day of a new vehicle that meets a set of requirements. But it may also include a pack for maintenance, Country audits allowing the vehicle to run in said countries, Sales and marketing kit, may be a set of vehicles built during the tuning of the product line, a user guide for the product.

A satellite project could aim to deliver a telecommunications service that meets a set of performance requirements and is piloted by a control center located at a certain location. Here the constellation of satellites which could be necessary to carry out this service does not even appear in the definition of the product of the project.

A project can also be a feasibility study for a new product or service. In this case, the product of the project is this new product or service at the feasibility stage which will take the form of a dossier.

During a recipe modification project in a pastry shop, the product of the project will be the production system for the new cakes created, i.e. a chain of new supplies, new recipes and all the necessary utensils. to their realization that it would be necessary to modify or buy.

Below you will find a set of early project product-tree for various kind of projects. We draw attention on some subtleties when necessary.

New product manufacturing project

- Production line
- New product definition
- Tools and documentation for after sales and maintenance network
- Conformity and authorization for selling product in target countries
- Sales and marketing kit for manufactured project
- Set of products for supporting first sales
- Supplier chain logistics service
- New product logistics service
- New product user's guide

The real output of the project will be an operating production line delivery e.g. 10 000 parts a day. Some products in the product-tree may be documents (classified as software) like the product user's guide. Other may be services, e.g. for logistics here.

New car fanufacturing project

- Production line
- New car definition
- Tools and documentation for after sales and maintenance network
- Conformity and authorization for selling new car in target countries
- Sales and marketing kit for manufactured project
- Set of cars issued from production line for supporting first sales
- Supplier chain logistics service
- New product logistics service
- New car user's guide

That may be strange at first but the goal of a car project is not to issue a car but to issue a production system yielding a flow of new cars. So in this context, we might not call a car issued from the production line a "product" from the project point of view. It is just an item that is most probably out of scope of the project, except may be for a limited set of cars produced during the ramp up of the production line and for the purpose of verifying quality

New Satellite communication service

- Satellite constellation
    - Satellites product line
    - Supporting module defiition
    - Payload definition
    - Satellites supply chain logistics
    - Satellites logistics to launchers
    - Launcher services
    - Ground system
- Ground system for communication management
- Operation guide

In this example, the New Satellite communication service is at the same time the root of the project and the product that will be delivered eventually to the end customer when it will be in a sufficiently mature state. This is an example where the objective of the project is to deliver a service (which is a particular kind of products).

- **Movie project**
  - **Post production system**
    - **Movie raw record**
      - **Casting**
      - **Decoration**
      - **Costume**
      - **Makup**
      - **Special Effects**
      - **Equipments (camera, lightning…)**
    - **Post production studio service**
    - **Music**
  - **Location services**
  - **Transport services**
  - **Distribution system**

The example of the Movie project is directly based on the credits you can read at the end of a movie. When establishing this list, you need to think of the products with a life cycle that you do have in the movie project. Think of every line here and their life cycles. The Movie project has a design phase the output of which is the scenario and the script. The scenario and script become inputs to all subproducts during the design phase. Probably location and transport services are widespread in the Movie raw record tree. It can be useful to gather all such services together but they could be as well attached to every elements of the product-tree that requires these services.

Movie raw record is a parent product of casting, decoration, costume… the reason is that all these sub products do not have another finality than to support the movie recording. For the same reason, the Movie post production system is the parent of Movie raw record and music for instance.

- **Company transformation**
  - **Former organizations**
  - **Former information system components**
  - **Former processes**
  - **New organization V1**
  - **Add on to information system V1**
  - **New process V1**
  - **Pilot project new process V1**
  - **New organization V2**
  - **Add on to information system V2**
  - **New process V2**
  - **Pilot project new process 2**
  - **Brainstorming Workshops**
  - **Audits**
  - **Training program**
  - **Governance system**

In the company transformation project example, a process is a product in the software category. A transformation is nothing else than the substitution of former processes by new processes together with the changes in operating them (by means of an organization and an information system). Remember the project product-tree should contain all products on which the project thinks there will be some activities to perform.

Removed products are part of the project product-tree. The reason is that there will be activities to perform on these projects, e.g. retirement but it may also be reuse for some parts or supply to another organization.

Here we added two pilot projects and two versions of the new system. Projects are cases where newly organization, information system and processes act together on a representative case in order to verify and validate intermediate transformation objectives. Having revisions of a system explicit in the product-tree really make sense as soon as this will be transferred to the end client at some stage.

My new car

- **Car Acquisition process**
- **Funding**
- **Communication services**
- **Transports**
- **Registration**
- **Insurance service**

My new car project is an example of a very small project that a lot of the readers have certainly experienced. We refer mostly here to personal experience of the writers.

This project looks very much like an acquisition process in a company except a person usually does not benefit from an off the shelf process and means to perform it.

So here, we imagine that the project owner (who whishes to acquire a new car) has to define his own acquisition process and run it. The acquisition process may consist in several sub activities – gather functional needs, compare offers from various car makers and find a market price, define the criteria for the choice including, total cost of ownership of the intended use including foreseen cost of maintenance, key functional requirements and performance requirements like available space, autonomy, vehicle minimal seat height, number of years of guarantee, kind of available fundings, new or vehicle already in use, earliest date of availability…

My new car project is an example of a very small project that a lot of the readers have certainly experienced. We refer mostly here to personal experience of the writers.

This project looks very much like an acquisition process in a company except a person usually does not benefit from an off the shelf process and means to perform it.

So here, we imagine that the project owner (who whishes to acquire a new car) has to define his own acquisition process and run it. The acquisition process may consist in several sub activities – gather functional needs, compare offers from various car makers and find a market price, define the criteria for the choice including, total cost of ownership of the intended use including foreseen cost of maintenance, key functional requirements and performance requirements like available space, autonomy, vehicle minimal seat height, number of years of guarantee, kind of available fundings, new or vehicle already in use, earliest date of availability…

My new house

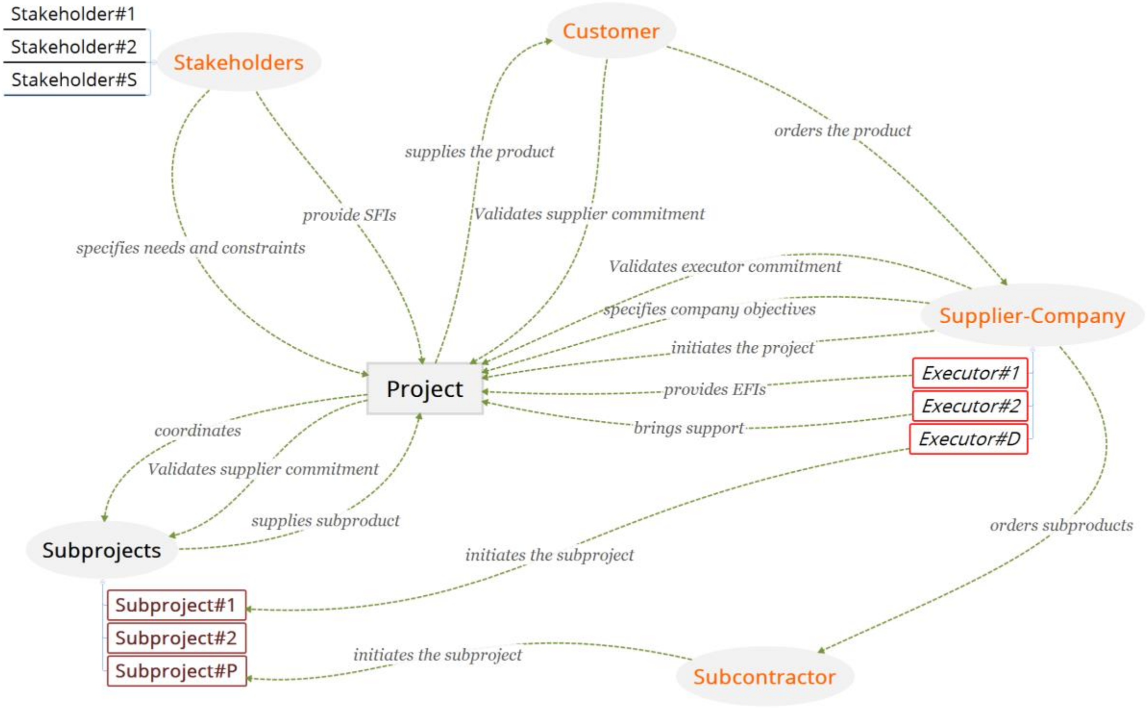
- **Land**
- **Construction permit**
- **Earth removed**
- **House**
- **Potable water**
- **Used water**
- **Electricity**
- **Gas**
- **Fences**
- **Garden**

For an house project, the construction permit is a very interesting case of a product that is not under the responsibility of the project. It is delivered or not by a public service on which the project has not control.

These examples demonstrate the generality of the concept of project product-tree and how it can be adapted to any kind of project.

4.1.6 External analysis and CFI, EFI, SFI

External analysis is a step where you consider all entities around project organization and identify everything the project will receive or send to these entities.

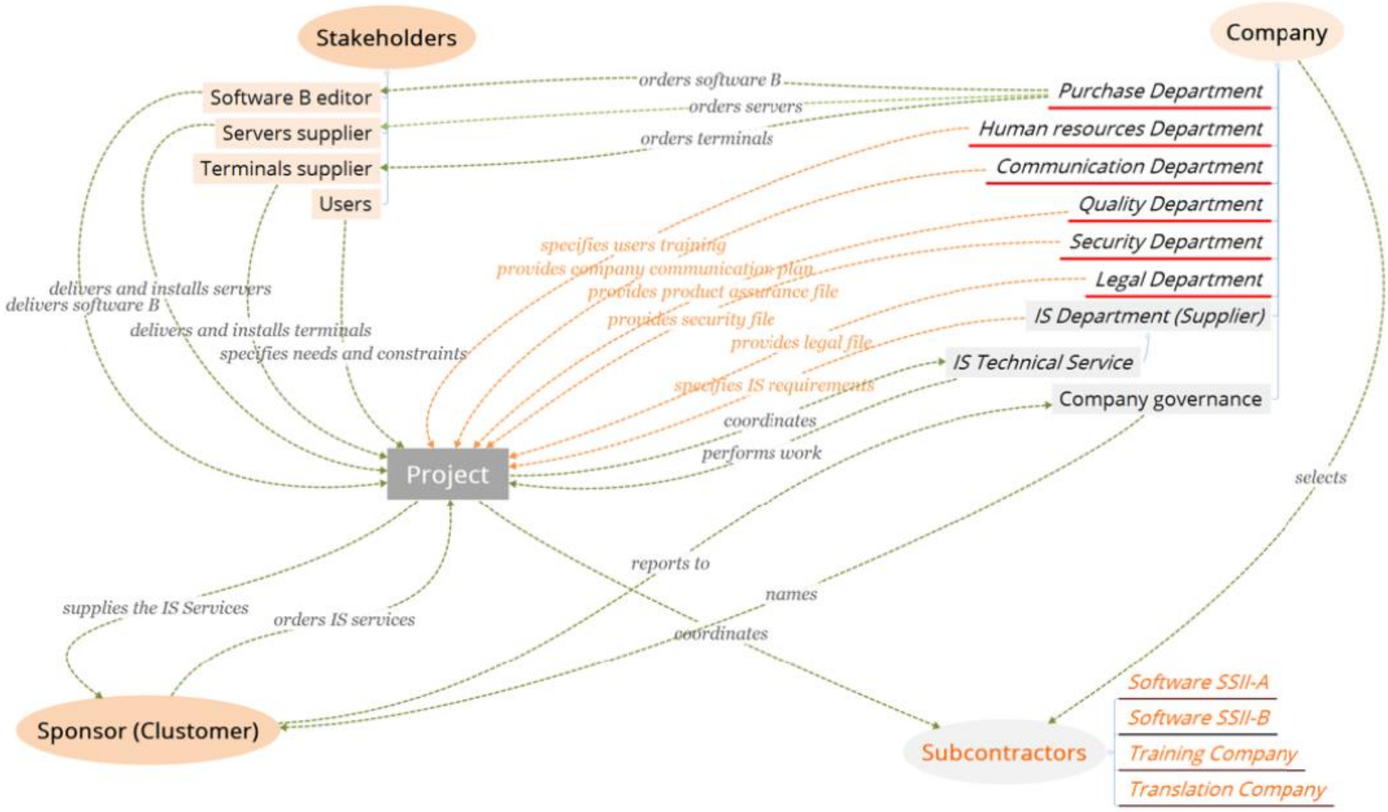


SFI stands for Stakeholder Furnished Items, a SFI is an item delivered to the Project by a Stakeholder, an organization with which the Project is not in a Client-Supplier relationship.

EFI are Enterprise Furnished Items, Items that belong to the organization and delivered to the Project.

Although not displayed in the Figure for the sake of keeping it as simple as possible, **CFI** as Customer Furnished Items correspond to items delivered to the Project by the Customer.

Here are some examples of external analysis for an information system and a house projects:



There is no CFI.

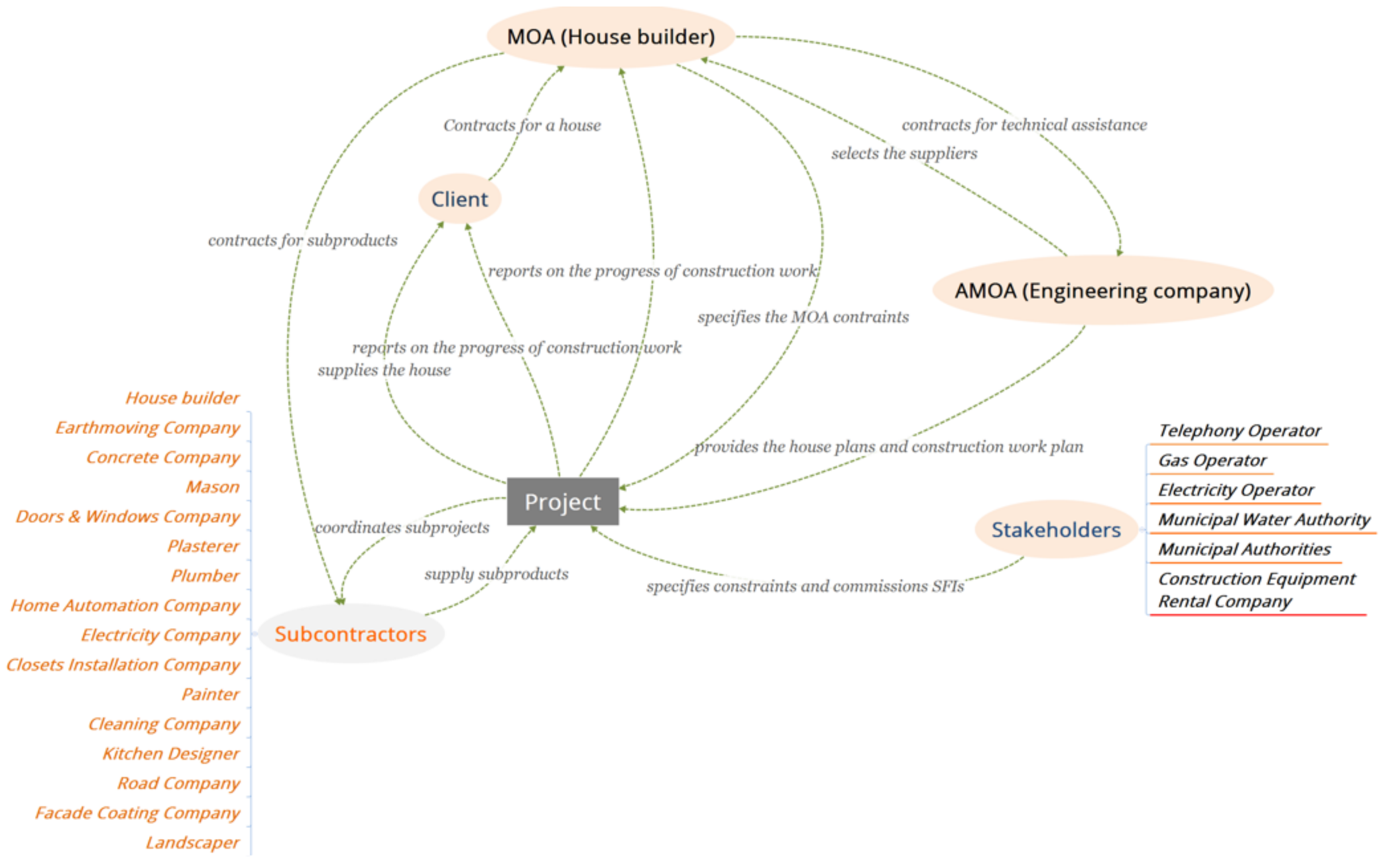
The *pool of users* which intervenes at the request of the project during the qualification works is a SFI.

EFIs are:

- the *product assurance file* provided by Quality Department.
- the *product legal file*[1] provided by Legal Department.
- operation and maintenance concepts and Company IT requirements provided by IS Department.
- the *services change file* provided by Communication Department.
- the *list of V1 – V2 users, provision of training premises* and *company needs for training* provided by Human Resources Department, who is responsible for training staff.
- the *company IT security file*[2] provided by Security Department, who ensures the protection of sensitive data.
- *access control to the company* provided by Security Department, who controls people's access to the company's means and premises.
- *Current IS services documentation* is made available by the IS Department.
- Software B licenses, servers and terminals acquired by the Purchase Department.
- *software A* and *IS application database* under warranty (releases V1-G and V2-G) provided by IS Department. They are respectively components of the IS application V1 or V2 which will be used for qualification of IS services.
- the deployment of IS services for users involved in the transition activities[3] is performed by IS Department.

Supplier of servers, terminals or Software B licenses are stakeholders. They must provide contact information so that the project can ask them questions about the use of their products, if needed.

Some EFIs are elements of the Project Product tree because the project must perform activities on these products. It is the case of servers, terminals and Software B.



[1] Some subproducts come from foreign countries (servers are produced in USA, terminals in China, software A is developed in Morocco, licenses of software B are purchased in USA). Suppliers must respect government constraints such as export control or customs regulations.

[2] The processing of sensitive data requires compliance with the legislation of each country where the services will be implemented.

[3] In particular, the application and terminals must be deployed to the users involved in the transition activities.

The contract between the client and the architect includes:

- land registry for the implantation of the house;
- building permits;
- soil analysis report prerequisite for building the foundation of the building;
- the availability of the land during all the construction time of the house.

There is no CFI and EFI.

SFIs are for the provision of infrastructure external to the land:

- Access path to the land, that is provided by the Town planning services.
- Electric meter, that is provided by the Electricity Operator. Its commissioning is conditioned by the supply of the Electricity conformity report that must be obtained by the Electricity Company
- Telephone box, that is provided by the Telephony Operator
- Water meter, Wastewater collector and Storm water collector that are provided by the Municipal Water Authority
- Gas meter, that is provided by the Gas Operator. Its commissioning is conditioned by the supply of the Gas conformity report that must be obtained by the Plumber Company
- Site Electric meter provided by the Electricity Operator and Site Water meter provided by the Municipal Water Authority for the duration of the work
- Site construction equipment, means leased to the Construction equipment rental Company for the duration of the work

When establishing the Project Product Tree, you should elaborate in parallel a Project external analysis in order to identify all interfaces with Client, Stakeholders, Company, Suppliers. On some cases, most of the Project Product Tree can be found during this analysis when the Project is merely an acquisition Project.

4.2 Activities

4.2.1 About activities

We use the word “Activity” in its usual meaning, as a set of actions performed with the help of some resources.
Activities are the basis for scheduling and analytical cost estimates. An activity can be complex and include sub-activities but in this document we will not differentiate between simple and complex activities.

NB:
- We will avoid talking about **process** and **task** to simplify.
- We will speak of **standard activity** instead of process, using the terminology of the ISO 19008, an Oil&Gas norm.
- Tasks have a meaning close to that of activity. The term activity seems to us the most widespread. Tasks are often containing a set of activities. Input and output would be usually associated to tasks and a task may contain an set of activities without relying directly activities to task inputs and outputs. In this paper, we will consider activities do have inputs and outputs and can contain sub activities so that activities covering all aspects of tasks.

A distinction will be made between a **specific activity** that is specific to a project and a **standard activity** that applies to many different projects and is not yet specialized. As we will see later, the specific activity derives from the standard activity through the process of **instantiation**.

Activities have inputs/outputs. When delivered to the project client or a stakeholder, an output will be referred to as a **deliverable**.
The activities inputs/outputs lead to precedence constraints: an activity cannot start before having received at least one input or end before having received all its inputs. Moreover, the purpose of an activity is to build its outputs. Any activity is always producing some output.

Reminder: It is very important not to confuse the words product and input/output.
We already provided an explanation of this by making a clear distinction between product and product life cycle. We will clarify this point later.

4.2.2 Standard activities

This ISO 19008 refers to standard activities that can be applied to all types of projects. They have inputs/outputs which are documents or items (physical entities). Activities inputs and outputs of standard activities are also standard. We use standard activity the same way as process in this paper.

Each company has its own processes that can be formalized or not. The Capability Maturity Model or the Spice model are Models to estimate the maturity of a company in executing properly its processes.
CMMi for instance provides a scale of maturity with roughly 5 levels.
Level 1 is usually called the heroes era. Success of a project relies and individuals bringing skills know-how, rigor and high involvement in their work.
Level 2 requires projects do have their own processes and apply them accurately.
Level 3 requires processes are established at company level and apply to all projects.

Level 4 requires ability to measure efficiency of applying processes.
Level 5 means ability to perform continuous improvement relying on tools developed at levels 3 and 4.

It is commonly agreed that a company product quality and efficiency depends directly on its ability to manage and run it's processes with accuracy and the CMMi scale gives an indication how this is the case.

So having an off the shelf set of processes to be applied is rather common but this depends on the industrial domain. Many processes are well described in norms and doctrines. This is the case for systems engineering processes and Project Management processes.

Systems engineering processes are well described in the INCOSE handbook and also in norm ISO 15288.
Inputs and outputs of life cycle processes are defined in ISO 15289.
In the satellite domain, ECSS is providing an accurate description of project management and systems engineering processes.
Project management and cost engineering activities are also explained by several doctrines (PMI, AACEI, IPMA, APM (Prince2)). In France, the MOP law standardizes activities for public tenders.

Other technical activities are often standardized as well in national or international norms. Civil engineering, command & control, hardware, mechanics, …

4.2.3 Standard activities: make or buy

There are only three **ways to obtain** products from the project product-tree:

- Make (renewal, implementation by the organization hosting the project)
  - A variation of **make** is **reuse**
- Acquire
- Buy (catalog purchase, purchase of an external implementation)

Standard activities therefore begin with these three ways.
Standard activities for "making" are typically design or production activities
Acquiring is about getting another entity making the product of interest.
The purchase is generally made by the company hosting the project. For purchasing activities, the standard activities are essentially the management of Purchasing requests to the company and the receipt of purchased supplies.

There are two other ways to point out which are not different in substance but from an acquisition.

- First, the implementation of products or services by a stakeholder. Here stakeholder means an actor with whom the project has a relationship but without a contractual client/supplier relationship. For example, the water company that comes to make the connections to rainwater and waste water network when you build a house. It is a public service. You have no power over the water company and vice versa. The law imposes a certain number of things on each entity.
- Second, the collaboration where two or more entities decide to co-produce a product. Here "Make" takes on another dimension in the relationship between the project and the co-contracting companies.

4.2.4 Specific activities and Instantiation. 3D picture.

Project-specific activities are those defined to describe a project.

Suppose your project aims at building a house.

- On one hand, a standard activity could be "build a wall".
- On the other hand a product may be a particular wall of the house.
- The instantiated activity will be "build this particular wall of the house".

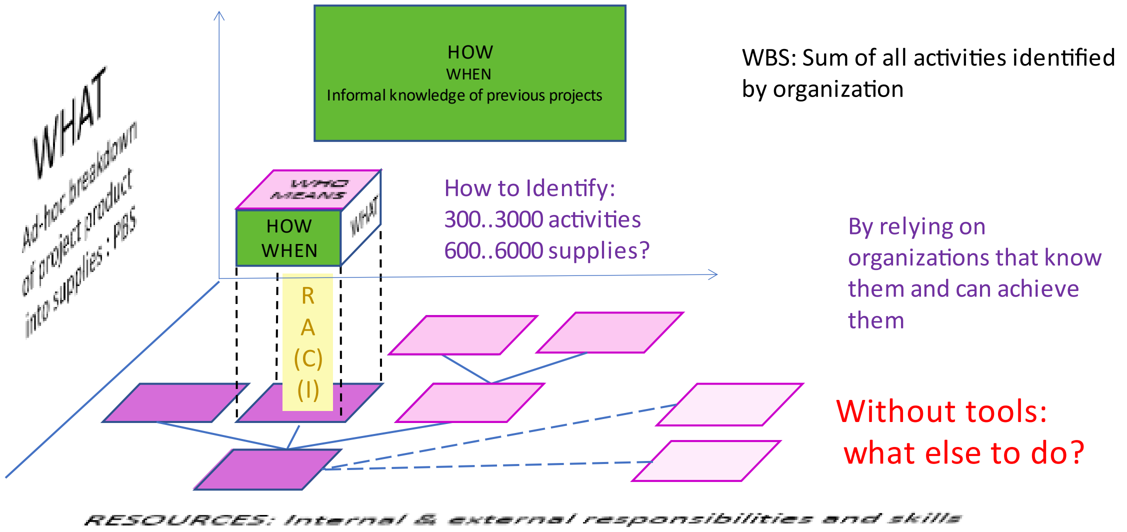
At first sight, this may seem obvious. The difficult part is that during the instantiation, all the inputs and outputs of a specific activity are consumed and produced by other specific activities. It is possible to set up an algorithm to specify the instantiation process of standard activities to products in order to get specific activities (or simply activities) in such a manner that inputs and outputs will he instantiated consistently as well.

So the point is not only to instantiate activities but to take into account all the interfaces and precedence constraints between activities at the same time.

The following **3D Figures** illustrate the specialization and t**ransition from a standard activity and from a product to a specific activity**.

They also present the 3 main areas that bring together **all the project data**: products, activities and resources.

Current approach to identifying project activities



In state of the Art, activities are rather identified bottom-up.

When projects are close to former projects, a reuse and adaptation of existing activities and subsequent planning are performed.

When projects are complex, it is often the case a set of organization is identified early to help doing the project and a Work Breakdown Structure is initiated to clarify the role of each organization. Each organization then define its activities bottom-up in each Workpackage (a set of activities under responsibility of the organization). Afterwards, there will be a work to understand interfaces between the workpackages.

When projects are quite new or the organization carrying the project has not much experience, defining the activities of the project is a challenge.

In all these cases, the definition of activity is bottom up in the first place which means it’s not ego less

Each organization, each team, each person contributing is doing this according to its own experience and understanding.

There is a high variability in the way two project teams may handle their project information both inside a given organization or across different organizations in the same domain.

This makes benchmarking and lessons learned difficult to establish.

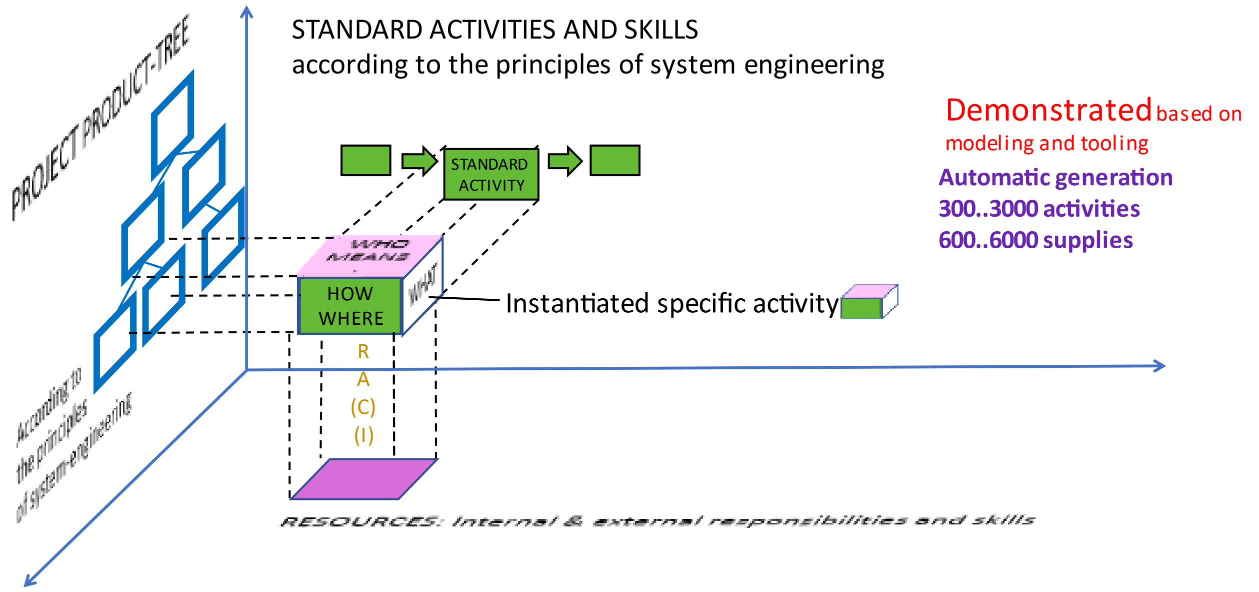
Eventually we must consider that there is no theory of planning development that integrates the possibility of comparing variants and versions of a planning.

The main objective of this white paper is to establish a reliable and efficient theory of project planning.

It is reliable because, as we will see, the activities are faithfully traceable from project objectives and activities and precedence conditions can be formally generated from a minimal set of project inputs.
It is also reliable because we can easily compare versions or variants of a project (version means same project definition but with different level of maturity of the project definition, variant means different project definitions upon a given project objective). As a consequence, it is also possible to perform reliable benchmarking between different projects.

It is efficient because most of the activities and planning can be generated automatically.

Our approach for identifying project activities



The idea is that because of the instantiation mechanism, it is only needed to associate standard activities to products in order to get exhaustively all instantiated activities.

For instance, suppose you start with a project product-tree with 10 products. For each product you associate a standard activity with say 10 sub activities and each sub activity has in average 9 outputs.

As a consequence, you put 20 information as input (10 products and the choice of 10 standard activities) and you get about 1000 information as output (100 instantiated activities and 900 outputs).

It's also possible to automatize the scheduling or at least propose a good approximation that can be tuned by the project team afterwards.

This is a top-down approach completely different from the state-of-the-art approach.

The instantiation mechanism ensures traceability and automation of the planning definition and a very important leverage.
As a consequence, lessons learned allow improving continuously standard activities, benchmarking is possible, project information quality is excellent and can be the basis for applying further analysis or AI technics.

At this stage of our paper, instantiation therefore requires a library of standard activities built on feedback from past projects. The library can then be continuously improved based on feedback from past and current projects.
More specifically, it is relevant to capitalize on how standard activities have been instantiated on past projects (on which products, with which resources and skills, etc.).

Now we explain why applying systems engineering rules for the project product-tree is important.

- To understand why it is very important to follow these rules, we must return to the example of the egg. In the case of the Egg structure, suppose you rely on the wrong geometric tree and we need to specify what is an egg. According to the geometric tree, we just need to specify the needs for the shell and the shell will specify the needs for the air and the white in the egg: this is simply nonsense. How the shell of an egg would know about the white and why it is needed? On the contrary with the correct functional structure, it is correct to say that the egg shall specify the needs for all sub components (shell, air, white, yolk). Same with our software system, the hardware itself cannot send a needs specification to the software. How a microprocessor would know about the software that it will execute? Again, this is nonsense.

In order for the approach to make sense and for instantiation to work, it is required that the project product-tree meets systems engineering rules.

At this stage, the whole approach relies on the existence of a library of standard activities, something that may seem unreachable for most of the companies.

Next chapter explains it’s possible to rely on an approximation of the standard activities based on systems life cycles as defined by systems engineering.

4.3 Approximating standard activities with system life cycles

4.3.1 No activity without a product

Any project activity is always performed on a product.

An architect designs a house, a building, a plant, something anyway.
A painter is using paint.
An electrical installer is installing a harness and electrical equipment.
A cook is cooking some food.
A driver is operating a truck.
Fundamental science is operating on theories that would be part of software. Of course, the more abstract is the activity, the less physical is the product.

Whatever activity you think of, the activity is operating on some object. However, when someone describes his activity, it is not always obvious to identify all the products (enabling systems, product of interest) he is using. A painter would tell you about the way or technics he applies the paint but you should not forget that, first of all, it is the paint performed which is the product.

4.3.2 No activity without a job function

Any activity requires some skills or capabilities. We call this a job function

4.3.3 The life-cycle & job function pair

Given these preliminary remarks, we remind from the product section that any product has a life cycle.
It is possible to approximate any standard activity with a pair consisting of a life cycle and and job function.

For instance, in the case of the painter, we can translate the paint activity as a triple: (the paint, Build, Painter).

In that case the standard activity would be (Build, Painter) and the product on which you apply the activity is "The Paint".

System lifecycle processes apply to any system. In particular they apply to any product. Conversely, any job function operates with a system life cycle (design, production, verification, validation…).

As a consequence, any organization can rely on standard systems life cycle libraries and a set of job functions to define its own standard activities and apply the approach we present.

4.4 Milestones and reviews

Reviews can be standard or specific. The purpose of a review is to verify that an intermediate objective has been achieved. For this, a certain number of supplies and deliverables are reviewed during a meeting to confirm that they have been completed and are accepted.

Reviews can be presentations to **decision makers**. In particular, a number of reviews are requested by the end customer. Conversely, reviews will be requested from subcontractors. Other reviews may be requested by the project to verify the physical progress of the project. We can see the reviews as a point of contact between the project and other job functions since it is at this moment that the project can accept the outputs of activities.

The reviews aim to have the activity managers explain to the decision maker (for example the project), the progress of their field, the problems encountered, the changes proposed. The result is the acceptance of supplies, with possible reservations, or corrective actions (unplanned), or a rejection. In general, during reviews requested by a customer from his supplier, the review is organized by the supplier and the customer is invited. More generally, the review is organized by the teams that produces the reviewed supplies.

Project reviews are specific reviews at project level.

The Project reviews and then product reviews **convey the logic of the project**. The availability dates of the products and the end of the activities that create them are derived from this. Of course, there are iterations. But at the very beginning of the project, when the specifications are only functional, there shall exist a general frame of the main project reviews, with dates and duration objectives.

Standard reviews are reviews of **standard outputs** produced by **standard activities**.

We will always consider reviews as a specific kind of activities.

4.5 Standard data for comparison

4.5.1 Why compare?

It is impossible to judge an activity performance without comparing to similar ones either from the past or from other projects. Many comparisons are relevant:

- between various revisions of the project plan during project planning
- between the forecast and the actual status.
- between variants of the same project even if the scope of the project changes
- between similar projects
- between projects that are similar on the scope of a product or an organization.

However, most of the time, these comparisons are difficult to perform. We are therefore often happy enough to compare the end of the project realization to its initial plan, and we give up other kind of comparisons.

However, if one day we want to use artificial intelligence to control the costs and deadlines of projects, we will not have to be content to compare a project to itself, but all projects of the same type to themselves. , and this in time and space.

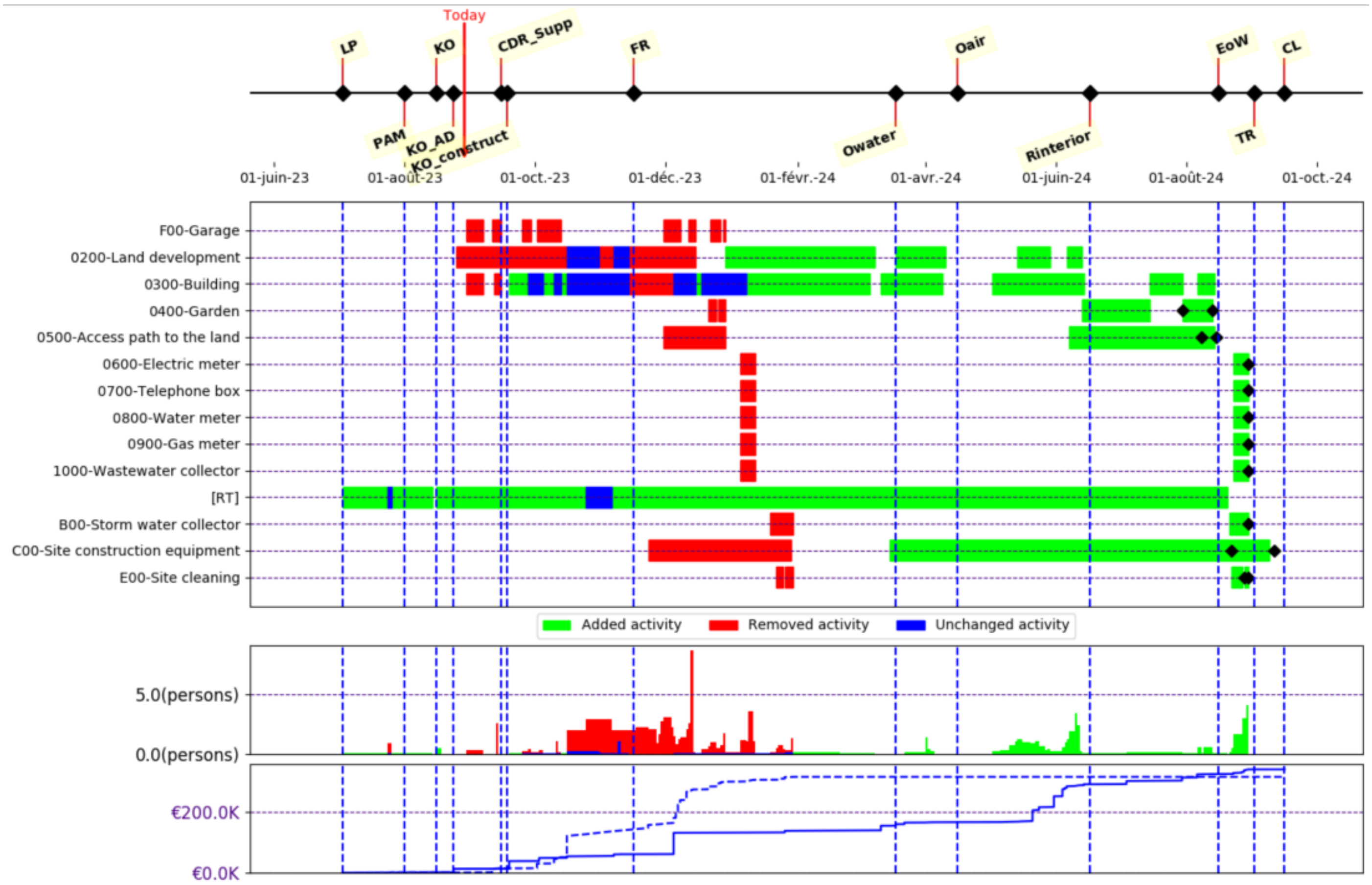
4.5.2 How to compare?

Standard data are stable data that applies permanently to all projects. Obviously, Standard activities are always part of the standard data. In the context of a single project, the project product-tree should be stable, at least at high level. When comparing similar projects, the projects product-trees can be aligned and project product-trees may form a stable pattern.

Once standard data is identified, it's possible to compare the items that are transverse and standard enough between different projects. The ability to rely on a top-down definition of activities gives the opportunity to extend the comparison field w.r.t. the poor bottom-up information structure of ad hoc work breakdown structures filled with ad hoc activities as designed by each organization and each person in each organization.

For instance, in the context of a project the project product-tree is progressively disclosed and the life cycles and job functions applied to the products are tuned progressively as well. In this context, it is possible to compare projects by filtering job functions involvement and cost per product. During progressive disclosure, applying systems engineering rules, the high level products will remain stable during the project. For the same reason, comparing project variants with limited changes in the project product-tree is valuable.

In the figure below, a comparison between two variants of a house project (kit house versus classical house) is displayed:



First life line reminds main project reviews. Then tree time diagrams aligned with reviews are displayed below. The first one is representing a list of products and at the periods on which activities are performed on the. The second diagram is a workload representation in time. The third diagram is an S curve representing purchase and acquisition expenses.
The green color represents activities occurring only in the reference project for the comparison (classical house here). The red color represents activities occurring only in the project to which we compare the reference (kit house here). The blue color represents activities in both projects.
In the S curve, plain line is the reference project and dotted line relates to the project to which we compare the reference.

We see here the advantage of the product-oriented decomposition to make a comparison.
Here we see that the right part of the diagram is full of green without blue and red which means the kit house project ends up much earlier.
The first product activities are completely red which means this product exists only in the Kit house variant.

It’s possible to visualize this representation on a part of the project.

5 DECISIONS

The working group resolved to:

- Open up internationally
- Develop prototype software to demonstrate the feasibility of Structuring Project Data and the benefits of the approach.

We offer to:
- continue the analysis of resource concepts, work packages, task flow chart;
- conduct tests on the software now available;
- work on the application of AI using this structure;
- assess the benefits of such an approach;
- facilitate its deployment.

The expected benefits are:

- Demonstrate benefits of the new approach

- Quality management of company projects based on the continuous improvement of standard activities.

- An ability to compare the performance of projects in real time or variants or versions of the same project

- A reduction in costs and delay thanks to this quality and software support thanks to a better structuring of information.

TABLE OF CONTENTS

P 2	1 Motivations
P 3	2 Manifesto
P 4	3 Concepts revisited
P 4	3.1 Product and project produc-tree
P 4	3.2 Standard activities and life cycle processes
P 5	3.3 Project activities
P 6	3.4 Milestones and reviews
P 6	3.5 Information for lessons learned and benchmarks
P 7	4 Comments
P 7	4.1 Produits
P 17	4.2 Activities
P 19	3D Pictures
P 22	4.3 Approximating standard activities with life cycles
P 23	4.4 Milestones and reviews
P 24	4.5 Standard data for comparison
P 26	5 Decisions
P 27	Table of contents