# Platform for Workplace Model Based Learning – Learn How To (*H2o*)

**Project Manager**
**Name:** Andrea Polini
**E-mail:** andrea.polini@unicam.it
**Fax:** +39 0737 402563

Didactic material provided for the course in:
**Software Project Management**
**MSc in Computer Science**
**University of Camerino**

**DISCLAIMER**

# Project Abstract

### Context

In modern society compaies and public administration (PAs) are under pressure to constantly improve their service quality while coping with quickly changing context (changes in law and regulations, societal globalization, fast technology evolution) and decreasing budgets. Workers are challenged to understand and put in action latest procedures and rules within tight time constraints. *H2o* will build an innovative holistic e-learning platform for workplace learner that enables process-driven learning and fosters cooperation and knowledge-sharing.

### Project Innovations

*H2o* technical innovation is based on four pillars:

1. a new concept of model-based e-learning (both process and knowledge)
2. open and collaborative e-learning content management
3. automatic, learner-specific and collaborative content quality assessment
4. automatic model-driven simulation-based learning and testing

*H2o* considers learning and working strongly intertwined (learning while doing). The platform supports both an informative learning approach based on enriched business process (BP) models, and a procedural learning approach based on simulation and monitoring (learning by doing). Formal verification and natural language processing techniques will ensure quality of content and documentation. Specialized ontologies and KPIs will be defined to keep learners engaged, while automatically derived tests will challenge their acquired knowledge. *H2o* is inspired by open-source communities principles and cooperation spirit: contents are produced by the community, and meritocracy is naturally promoted, with leaders emerging because of their skill and expertise.

# Contents

**Appendix A – Glossary of Important Terms and Acronyms** **5**

**1 Concept and Project Objectives** **6**
1.1 Motivations . . . 6
1.2 Objectives and Challenges . . . 7
1.2.1 S&T objectives . . . 10
1.2.2 Performance Indicators . . . 11
1.3 Technical Approach . . . 12

**2 Managing *H2o*** **13**
2.1 Project characteristics . . . 13
2.2 Project document description . . . 13

**3 Cost forecasting** **15**
3.1 Estimation based on similarities . . . 15
3.1.1 Analogy . . . 15
3.1.2 Delphi technique . . . 15
3.2 Decomposition . . . 15
3.2.1 Product . . . 15
3.2.2 Wolverton suggestion . . . 16
3.2.3 The verification component . . . 17
3.3 Process based estimations . . . 17
3.3.1 Activities . . . 17

**4 Risk Management** **19**
4.1 Risk categories . . . 19
4.2 Checklist . . . 20
4.3 Brainstorming . . . 20
4.4 Risks to the schedule . . . 22

**5 Resource Allocation** **26**
5.1 Introducing resources in planning . . . 26
5.1.1 Resources . . . 26
5.2 Typical resources . . . 26
5.3 Resource requirements . . . 26

# List of Tables

1 Glossary of Important Terms . . . . . . . . . . 5
2 *H2o* Performance Indicators . . . . . . . . . . 11
3 Boehm's categories for project risks, and possible reduction techniques . . . . . . . . . . 22

# Appendix A – Glossary of Important Terms and Acronyms

Table 1: Glossary of Important Terms

| Term | Explanation |
|---|---|
| ADL | Architecture Description Language |
| API | Application Programming Interface |
| ATL | ATL Transformation Language |
| BMM | Business Motivation Model |
| BP | Business Process |
| BPDM | Business Process Definition Metamodel |
| BPM | Business Process Management |
| BPMN | Business Process Modeling Notation |
| CMMN | Case Management Modeling and Notation |
| DoDAF | USA Department of Defense Architecture Framework |
| DSL | Domain Specific Language |
| EA | Enterprise Architecture |
| EC | European Community |
| EMF | Eclipse Modeling Framework |
| EPBR | European Project Budget Reporting |
| ESB | Enterprise Service Bus |
| ETL | Epsilon Transformation Language |
| GPL | Gnu Public License |
| GWT | Google Web Toolkit |
| HTTP | Hypertext Transfer Protocol |
| IAM | Identity and Access Management |
| KPI | Key Performance Indicator |
| KMS | Knowledge Maturing Scorecoard |
| LBS | Learning By Simulation |
| LMS | Learning Management Systems |
| LOM | Learning Object Metadata |
| MDA | Model Driven Architecture |
| MDE | Model Driven Engineering |
| MODAF | UK Ministry of Defence Architecture Framework |
| NLP | Natural Language Processing |
| OBM | Object-Behavior Model |
| OMG | Object Management Group |
| OSS | Open Source Software |
| OWL | Ontology Web Language |
| PA | Public Administration |
| PAB | Project Advisory Board |
| PMB | Project Management Board |
| QuARS | Quality Analyzer for Requirements Specifications |
| RDF-S | Resource Description Framework Schema |
| RDF | Resource Description Framework |
| ROI | Return Of Investment |
| SBVR | Semantics for Business Vocabulary and Business Rules |
| SCA | Service Component Architecture |
| SOA | Service Oriented Architecture |
| SUAP | Sportello Unico Attività Produttive (Italian Office to register a new company) |
| SVM | Support Vector Machine |
| UML | Unified Modeling Language |
| UPDM | Unified Profile for DoDAF/MODAF |

# 1 Concept and Project Objectives

*H2o* will provide a social, collaborative and learning platform for workplace learner. Using the *H2o* platform, workers will be engaged in a holistic learning, collaborative and assessment experience, wholly centered around a graphical intuitive representation of the Business Processes (BPs) and other models describing the different perspective of an organization. More and more processes in place within an organization continuously evolve, as well as for their surrounding context. The platform complements the graphical representation with additional descriptive contents producing an enriched and machine-processable model. It maintains an up-to-date correlation between the information and the activities represented in the BP specification, as workers use the platform either for learning or for serving real requests.

## 1.1 Motivations

**Addressed scenario:** Complex organizations in modern society are characterized by the need to support extremely complex processes in order to provide their services to customers (citizens in case of PAs). Processes typically include several alternative paths, many of which are seldom activated. Complexity is raised by the fact that the provisioning of services is in most cases a collaborative activity shared among different, possibly many, offices. Market changes, or the introduction of new laws/regulations, or the frequent modifications to existing ones further result in the intertwined modification, creation, deletion of supporting services and processes. Finally provided services are in general quite interrelated, so that activities carried out to deliver a service can generally lead to the enactment of other processes not originally considered.

Because of the above features, complex organizations are never done with learning how to carry out their tasks. To support them in learning, managing and mastering the complexity of processes, *H2o* intends to coalesce into an integrated solution several converging powerful ideas and tools.

**Business Process Modeling potential:** Business Process (BP) refers to any structured collection of related activities or tasks that are carried out to accomplish the intended objectives of an organization. The *H2o* project considers BP graphical representations as an intuitive mean that can be exploited in innovative ways as the basis for teaching and training workers who need to face and operate complex BPs. Processes can be represented and organized using a multi-scale approach with different, even if related, levels of abstraction. The different scales should correspond to different learning objectives and phases, where more abstract representations can constitute the base for teaching/learning the overall objectives and organization of the process, while more detailed representations can support learning activities related to specific tasks foreseen by the process in which an organization and a worker is involved.

Nevertheless the sole BP model might not convey enough information to support on the one side the enactment of the represented process, and on the other side the training of the worker who is assigned to the tasks. *H2o* will include mechanisms for tracing and storing of additional information that will be used to augment the process graphical representation, and in particular to relate information with the specific tasks to which they refer to. In such a way the process graphical description can constitute the "main door" to obtain a direct access to the knowledge necessary to carry out a specific process activity.

**Open Source-like cooperation:** The project wants to bring the methodologies and the principles driving open source communities in the context of complex organizations. Within open source initiatives the software is collaboratively written by a community of people, all feeling in charge towards a shared objective. The paradigm tends to create meritocracy, whereby leaders emerge because of their skill. *H2o* supports the idea of promoting workers as "prosumers" of the learning materials. The result is a collaborative/learning platform in which knowledge is shared among the workers. In this sense *H2o* fosters the creation of a learning/teaching community in which BPs are modeled, documented and evolved with the help of, and by, the workers themselves. In the same way the *H2o* platform can help to identify and provide references to colleagues who are "the expert" in specific topics, and whom can be easily contacted for guidance and clarifications. The platform is conceived to foster cooperation and to reward particularly helpful colleagues.

Approaches to rewarding typically adopted by open-source communities will constitute the inspiration for the definition of suitable KPIs (Key Performance Indicators) in this context. The definition of precise KPIs both for content/model producers and for learners fosters user engagements, and will be part of the activities that will be carried on within WP5. The platform will permit to recognize expertises and competencies of content/model producers, and of learners on the base of their engagements in production and fruition activities, respectively. For instance, within an open source community the role of committer is reached after having supported the community with relevant patches. In a similar way a worker could become a recognized expert in a specific BP after he/she has been willing to provide support to colleagues and he/she has participated in the production of learning contents for that BP. It is worth noting that the rewarding mechanisms conceived and embedded in the platform can be much more effective if the organization formally recognizes participation and collaboration as meritorious activities.

**Holistic all-embracing approach:** The project develops an integrated approach to engage workers in learning activities at different times, following different paradigms, and by different means. *H2o* fosters both an informative learning paradigm, by which the workers learn accessing and studying the BP model and related material, and a performative learning approach, by which the workers operate within a simulated environment reproducing real requests through the enactment of a BP. The above are off-line strategies, in which the workers acquire knowledge before serving real requests. However the typical complexity of processes defeats the human capacity to acquire a full knowledge on any aspect of a BP just through informative and performative approaches. It is necessary that the workers can actually retrieve and process useful and context-dependent information while they are working on real cases. *H2o* complements the learning experience with on-line strategies in which the workers acquire knowledge while they are serving real requests, by seamlessly supporting a "training on the job" or "learn while doing" approach. In particular, while serving real requests the learner should have the possibility to easily and rapidly get in contact with an expert colleague, who could not work within the same office. In this way they will feel more comfortable in the actions/decision they put in place to process real requests. The availability of such processes and mechanisms creates the opportunity for establishing a lively and collaborative community, even through the introduction of specific rewarding mechanisms, whereby "senior" workers on the job will be encouraged to provide support to less experienced colleagues.

**Tacit knowledge highlighted:** An important characteristic of processes in place within complex organizations is the presence of internal procedures resulting from the establishment of "habits", and that in some cases may not be anymore required by law, regulations or organizational constraints. Because they are part of tacit knowledge, such procedures are particularly difficult to highlight and learn, when they are benign, and to eradicate, when they are useless bureaucracy; in both cases they make the work of new employees more difficult. The possibility to collaboratively model and discuss such processes, in order to document them, can help the training of new employees and can also facilitate the reorganization of offices and work units, and the removal of unnecessary procedures.

## 1.2 Objectives and Challenges

Through the development of the *H2o* learning platform, the development activity will address in a integrated way three major objectives: *(i)* the efficient production of the contents to be learned, which as described in the above section, will be complex, networked, and evolving; *(ii)* the effective support of a holistic learning experience by the workers; and *(iii)* the promotion of social networking and collaborative engagement for training in the workplace.

Each of the above objectives raises a number of scientific and technological challenges.

**Learning Contents Production:** Figure 1 sketches the content production process supported by the *H2o* platform. Navigating the figure top-down, in Step 1 a set of laws, rules and procedures are explicitly codified using a graphical BP notation. In addition to the process models themselves, there will be models representing the context of the process. This process context corresponds to different views as they are represented in the enterprise architecture, in particular the information objects (documents and data), organizational structure (departments, roles, persons), process results, IT systems, business rules, and the business motivation (policies, strategy, goals, objectives and influencers). All these models with their relations can provide a valuable source of information for the workers (to collaborate and to learn) to immediately grasp the BP objectives and context. This step is a collaborative effort that involves domain and technical experts. The first defined model refers to the specification of the interactions and information to be exchanged among the different company offices involved in a possible enactment of the process. *H2o* supports Step 1 with **collaborative and web-based editing**. The embedded editor will permit to export defined models in open standards format. Formal verification approaches are also embedded in the platform so to **highlight possible incorrect models** (e.g. existence of deadlock or live-lock in the models).

Once BP and related models have been specified, additional documentation is added in Step 2. This documentation aims at a clearer understanding of the overall process objectives and at clarifying the relation with the motivations that originated the process itself. *H2o* supports Step 2 through the **integration of a Wiki platform that is able to automatically reflect the structure of the specified BP.** This is done using advanced meta-modeling techniques and basing the derivation and structure of the documentation on a transformation of the specific meta-model related to the graphical notation and other defined context models. The availability of complex meta-models for representing the business process structure, its data, and its business rules, permits to exploit its use also to **assess the quality of the provided documentation with natural language processing techniques**. This results in the possibility of introducing mechanisms that suggest to the community what documents need to be improved. Such mechanisms are particularly appealing in a collaborative and open context.

From the global view derived in Steps 1 and 2, Step 3 shows that each office involved in a possible enactment of the process will have then to derive its internal process specification. The derived specification should permit **the correct collaboration with other involved offices in order to reach the global objectives**. The two steps defined for deriving the global specification and its description are similarly carried on within each single branch possibly involved in the process. Nevertheless in this case the BP specification needs to conform with the global specification in order to permit a correct interaction among the involved organization offices. The *H2o* platform takes advantage from the availability of a global specification to permit the **formal verification of possible mismatches between the global and local views**. In case an error is highlighted the problem is reported to the modeler. The platform permits also to navigate between the global and local specifications and related documentations.

**Holistic Learning Experience:** *H2o* supports a vision of the worker as a "prosumer" of contents and models. While the process described above mainly reports the point of view of the producer, the developed contents will be also consumed by workers to learn how to support a specific BP and related services. In such respect, local and global models and contents will be used at different times and with different learning objectives. The global view, in which all the stakeholders are reported, can be used to understand the general needs and objectives that the process intends to support. On the other side, the local view provides information on how to organize the work within a specific organization, also in order to permit the correct interaction with the other organizations.

Models and contents available within the *H2o* platform permit to support different approaches to learning. In particular an **informative paradigm to learning** is supported by the availability of models and related contents that can be accessed and studied by a worker. The learning is made particularly effective by the possibility of having different views on the process, and by the possibility of navigating the contents on the base of the activities foreseen by the process. Thanks to the availability of precisely defined notations for data description and process flow description, based on a precise meta-model, the *H2o* platform supports also a **performative paradigm to learning**. In particular the project will investigate the use of model descriptions to automatically derive process execution simulations through which the worker can directly assess their acquired knowledge. **Self-assessment** refers to the possibility of checking that the worker has acquired enough knowledge on the activities he/she will need to support. BPs of big companies and PAs are typically complex and include many alternative paths, in many cases seldom activated. Therefore, if on the one side worker can quickly become relatively expert with respect to the main scenarios, on the other side the scenarios related to the "exceptional" behavior will be hard to learn. The identification and the coverage of BP specification in order to assess the outcome of the learning process in some respect relate to the issues which testers face in checking the correctness of a software, through a finite set of experiments. *H2o* intends to include in the platform mechanisms to support **simulation and questionnaire-based assessment**. The development of these mechanisms will be inspired from testing techniques and meta-modeling manipulation techniques to select interesting traces/experiments.

Finally in order to permit process simulation, the *H2o* platform automatically derives an executable model of the process specification using model transformation techniques from the specified procedural and data models. These mechanisms will rely on a **sophisticated execution engine that will simulate the environment and the stakeholders participating in a given BP**. Furthermore, the execution engine needs to support the BP notation language and to be equipped with suitable monitoring tools. The monitoring platform is considered as an upper layer that will be able to follow the behavior of the initiated

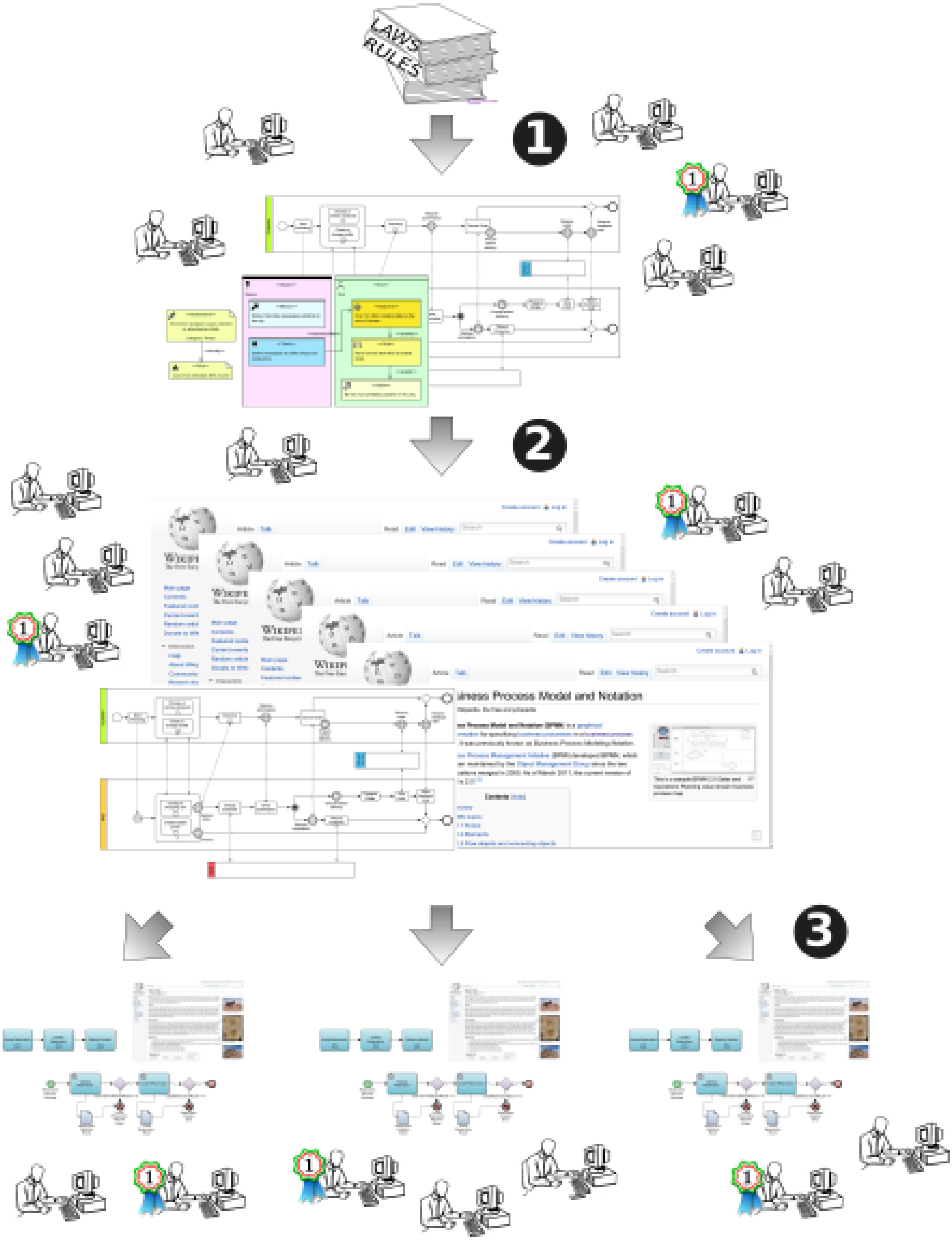


Figure 1: Models and Contents Production Process

BPs. Moreover, it will be able to support many BPs as they are enacted. The objective of such mechanisms is to operate run-time monitoring activities that will result in reports and graphs, which will be further used for analysis.

In addition to learning activities to be carried on before serving within concrete enactments of a process, the *H2o* platform supports access to models and contents while a worker is serving real requests. In such a case the platform offers the possibility to directly access models and contents specifically related to the activity the worker is performing. This will reduce the burden of selecting relevant information among the big bunch of information related to a specific process, and will certainly make the learning experience easier and more effective, according to a **"training on the job" approach**. In order to permit a context-dependent provisioning of contents to workers the **platform provides an open API** to be used for interaction with external software systems possibly deployed within a company office.

**Improving Learners Engagement:** In *H2o* engaging learners and keeping them receptive are considered as relevant priorities. To reach these objectives it is important to take into account the previous knowledge and the experience of the learners. Only by providing relevant and new contents it is possible to have effective learning experience. *H2o* puts in place specific mechanisms to support this view. In particular through the adoption and definition of **ontology-based mechanisms** *H2o* **will permit to reason about contents and learners**. Specific ontologies will be conceived for classifying and reasoning about learning contents and learners, thus permitting to effectively **relate learning contents with learning needs**.

While supporting individual and self-directed learning, *H2o* intends to foster a social community in which learning takes place also thanks to engagement and interactions with other peers. Workers who are the expert for an issue (represented with a "cockade" in Figure 1), can thus be **identified and contacted by colleagues on the basis of their specific expertise**.

The platform will enable any worker to provide, at any time, feedbacks and suggest improvements on specified BP models. Nevertheless the models update should be in the responsibility of appointed employees. Uncontrolled modifications of such diagrams must be avoided since they can lead to unpredictable effects on the attached contents. On the other hand the wiki-based content can be edited directly by the workers in order to enrich the learning material and to provide support to colleagues. **Sharing and cooperation will be strongly fostered by the platform**, introducing mechanisms inspired by the open source and open model communities. Such mechanisms should permit to selectively evolve contents and models on the base both of subjective judgments, by the workers accessing to models and contents, and of objective measures, for instance considering monitored learning data such as the results workers get in simulations, and after having studied specific models and/or contents.

Engagement and social behavior will be fostered by the introduction and support of an **ontology model for KPI**. Such mechanisms are introduced in the platform to reflect learners as individuals, learners as participants and contents. Such characterization is codified in the **KPI model: it will permit to clarify objectives of learning, participation and social activities**, so improving engagements of learners. *H2o* will devote effort in the definition of a KPI ontology, by specifically devoting one workpackage to define specific mechanisms to be introduced in the platform.

### 1.2.1 S&T objectives

In summary the **scientific and technological (S&T) objectives** of *H2o* are:

- S&TO-1 – BP context and modeling: in this respect the project intends to identify and provide suitable notations/languages, possibly derived from standard BP notations, that are easily understandable to non experts in programming languages. Models will represent procedural knowledge and its context. The resulting modeling notations should permit to link the models with a set of related learning contents.
- S&TO-2 – BP multi-level views: modeled business processes (and related meta-models) should be visible and managed at different level of abstractions. High level views should be available to workers who can explore and learn the details of a business process possibly using a simplified notation. On the other hand detailed descriptions should be available for monitoring and simulation purposes. Both Local and Global views on process will be supported.
- S&TO-3 – Models and contents analysis and verification: defined business processes should be checked for adherence to qualitative properties such as deadlock and live-lock, and realizability. Particularly challenging seems the possibility to adopt natural language processing techniques to assess the textual documentation associated to a given process. In such a way the platform can automatically suggest the pages in the wiki (documentation) that need to be improved. These techniques will take advantage from the availability of complex meta-models permitting to relate models and contents.

- S&TO-4 – BP simulation and monitoring: specific mechanisms and engines for process flow simulation will be investigated. In addition, monitoring mechanisms will generate activity reports that will enable analysis and corrective actions. Such mechanisms will be useful for training and assessment purposes.
- S&TO-5 – Ontology Modeling for Learning in PA: focused ontologies to support e-Learning in complex organizations will be developed and released to the community.
- S&TO-6 – KPI for Learning in PA: specific KPIs and related ontologies will be defined and released. Such mechanisms will have to pose particular attention to fostering learner engagement and cooperation.

*H2o* will have to address the above listed objectives. The project will have also to experiment for learning and training purposes recent approaches in use in the software communities involved in model-driven, business process modeling, simulation and monitoring, ontology. The project specifically aims at deriving an integrated platform for learning and tutoring of workers. The research will be carried out in focused workpackages, each one resulting in the production of software elements that will be integrated to derive the final platform.

### 1.2.2 Performance Indicators

Table 2 outlines a list of indicators of the performance of *H2o* with reference to the S&T objectives.

Table 2: *H2o* Performance Indicators

| **Objective** | **Indicator type and indicative example** |
|---|---|
| S&TO-1 – BP context and modeling | **Availability:** Notations and languages for BP that result easier to be understood by non experts in BP. The resulting notation can be considered by extending standard notations (e.g. BPMN) with ad-hoc domain-specific notations.<br>**Dissemination:** Number of papers (including technical reports) published on defining, instantiating, or documenting the notations/languages above. Number of releases over open source dissemination platforms of products, models, and artifacts.<br>**Standardizations:** Submission of the notation above to the Object Management Group or other appropriate standards bodies for consensus building as potential new standard or extending existing ones. |
| S&TO-2 – BP multi-level views | **Availability:** Transformation engines for converting, referring, tracing, and dealing with the co-evolution of both model elements belonging to different views/scale, and their related learning contents.<br>**Dissemination:** Number of papers (including technical reports) published on these research aspects. Number of releases over open source dissemination platforms of products, models, and artifacts. |
| S&TO-3 – Models and contents analysis and verification | **Availability:** Automatic facilities implementing formal verification techniques for analysis, consistency checking and pragmatic ambiguity detection of BP model specifications and contents.<br>**Dissemination:** Number of papers (including technical reports) published on these research aspects. Number of releases over open source dissemination platforms of products, models, and artifacts. |
| S&TO-4 – BP simulation and monitoring | **Availability:** Configurable and multi source monitoring frameworks able to highlight criticalities in the simulated execution of BP and the associated learning contents and to provide feedback for the ranking activities.<br>**Dissemination:** Number of papers (including technical reports) published on these research aspects. Number of releases over open source dissemination platforms of products, models, and artifacts. |

| | |
|---|---|
| S&TO-5 – Ontology Modeling for Learning in PA | **Availability:** An ontology based on standards, to support BP-oriented learning. **Dissemination:** Number of papers (including technical reports) published on these research aspects. Release of the ontology to the open source community. |
| S&TO-6 – KPI for Learning in PA | **Availability:** An ontology-based KPI reference model for evaluating learning content and learner progress and fostering learners collaboration. **Dissemination:** Number of papers (including technical reports) published on these research aspects. Release of the KPI model to the open source community. |

### 1.3 Technical Approach

The technical approach the *H2o* project will follow in order to achieve the S&T objectives introduced in Section 1.2.1 can be summarized as follows (with tools and adopted notations):

- Comprehensive domain modeling of multi-scale business processes representing the internal procedures/regulations within PAs:
  BPMN, BMM, EMF/Ecore [2] (meta)-modeling notations; ADOxx (meta)-modeling notation
- Identification and formal representation of the meta-information that needs to be captured in order to trace both tasks (e.g. elements), and related learning contents within a business process over multiple scale view with different abstraction levels:
  EMF/Ecore [2] (meta)-modeling notations; ATL [3], and ETL [4] model-to-model transformation languages; Gra2MoL [5] grammar-to-model transformation language
- Identification and formal structure of ontologies that organize the knowledge over the business processes and their related learning contents:
  ArchiMEO – Enterprise Architecture Ontology; ATHENE
- Design and development of a collaborative e-learning platform where several approaches (e.g. for simulation, contents/learning reputation, quality evaluation) can be plugged in:
  XWiki; OBM[1]
- Integration of infrastructure for business process (i.e. BPMN) graphical editing and management: MagicDraw; Cameo Business Modeler; Petals BPM
- Integration of open ontology-based modeling and reasoning engine:
  Protege; Pellet
- Design and develop a simulation engine will support the execution business processes. On the one hand the simulator will includes means for automatic synthesis of the behavior of the involved parties. On the other hand it will include social means for gathering real learners:
  Petals Enterprise Service Bus
- Design and develop an event-based monitoring infrastructure on top the business process simulation engine. These infrastructure will also leverage the definition and implementation of mechanisms enabling reputation of both contents and learners: Petals Enterprise Service Bus; Drools Fusion; RuleML
- Development of infrastructures for formal verification and linguistic quality evaluation for business processes and their related learning contents:
  Support Vector Machines; K-fold Cross Validation; Rule-based Text Search; QuARS

[1] see at www.obm.org

# 2 Managing *H2o*

In the previous section a general description of the objectives of the project has been reported. The section could be considered as a quite refined overview describing an innovative software system for e-learning in teh workplace. In this, and following sections the objective is to apply the project management approaches presented in the SPM course. Choices and strategies will be discussed and detailed.

## 2.1 Project characteristics

In managing a software project specific characteristics have to be considered in order to derive a project plan. The first thing we have to consider refers to the motivations that lead to the definition of the project.

1. You act as the Contractor
2. In house software (build or buy): software that you need to run your business
3. Participation to research and innovation proposals

The context has an impact on the information that you need to collect in order to prepare a project document description.

**Which is the situation that best fit to our context?** In different context teh work to be done can be slightly different and differences will be reflected in the document to be delivered to the customer.

1. The first case refers to a situation in which the project is developed as result of a specific request made by a customer that contacted your organization. The customer has a specific need and wants to verify that it can be satisfied by the development of a software systems. In this case you will have to clarify the nature and complexity of the project, and then make a first estimation of costs. The customer will not receive details on how you derived such estimation. On the other hand specific investigations have to be carried on in order to derive reliable values. The customer is mainly interested in knowing that the system can be built at a "resonable price". In general you will not the one carrying on a market analysis.
2. The second case refers to the development of a sosftware system in order to carry on your business or to derive a product to be sold. In this case the market analysis assume a fundamental role since youo have to decide what kind of competitors are around and why your solution could be better or it could be better to buy it. The cost assessment is equally important to decide if it is worthy to proceed with the plan.
3. the third case refers to a situation in which you participate to a public competition for the provisioning of a system solving a specific problem. In such a case the market analysis is relevant in order to clarify the added value of your product but does not serve to justify the feasibility of the project. Indeed this has been somehow already established by the customer. This is the case of software systems required by public administrations or by R&D project financed by EU commission (or similar). In such a case often the document will have to include a detailed overview of costs. On the base of real costs the value of the project is derived.

## 2.2 Project document description

A document describing a project proposal can include all, or a subset, the following sections, that all together serve to establish the feasibility of a project, and how operatively the system will be developed:

- *Introduction and background*
- *The proposed project*
- *The market*
- *Organizational and operational infrastructure*
- *The benefits*
- *Outline of the implementation plan*
- *Costs assessment*

- *The financial case*
- *Risk analysis*
- *Management plan*

The sections will include different information depending on the target readers as illustrated in Section 2.1.

# 3 Cost forecasting

Different approaches can be taken to face the estimation problem:

- *Delay estimation* until late in the project
- Base estimates on *similar projects* that have already been completed
- Use relatively *simple decomposition techniques* to generate project cost and effort estimates
- Use one or more *empirical models* for software cost and effort estimation

**Which strategy would you consider more suitable in our context?**

## 3.1 Estimation based on similarities

In such a case the estimation is based on the cost recorded in carrying on a *project that appear to be rather similar*

Strategies are then based on *"Ask the expert"* approach:

### 3.1.1 Analogy

1. select several experts
2. ask them to independently provide an estimation for the software
   - *optimistic, pessimistic and most likely*
3. $S = \frac{s_{opt}+4s_{ml}+s_{pes}}{6}$

### 3.1.2 Delphi technique

1. select several experts
2. ask them to independently provide an estimation for the software
3. organize a meeting in which they can present their results
4. ask them if they want to revise, if yes go to 2
5. compute the estimation $S$ as the average

**Let's make a try!**

In general assessment strategies based on experience cannot lead to realiable results for really big projects if applied following a "one-shot" approach. In particular this is the situation that we are considering in our context. In such a situation it would be more reasonable to perform a first decomposition of the system so to identify smaller components that when integrated provide the needed functionality. Experience can then be fruitully deployed to derive values for the various components and then to derive a value for the integration.

## 3.2 Decomposition

To reach a more realiable estimation of the project costs the problem can be decomposed according to:

- *Product*
- *Process*

costs are then associated to *"components"-"activities"* while the overall estimation is a the result of recomposition of the various factors

### 3.2.1 Product

To proceed with a product based decomposition a *rough architecture and main components are identified*

**How would you organize the components for the system we need to build?**

In Figure 2 the architecture defined by the Learn PAd prject is reported. The decomposition is inspired on a repository architecture where models are stored on a common repository and accessed by the various components. At the same time for each component specific connectors have to be developed.

In the following we consider a revised and simplified version of the system as sketched in Figure 3

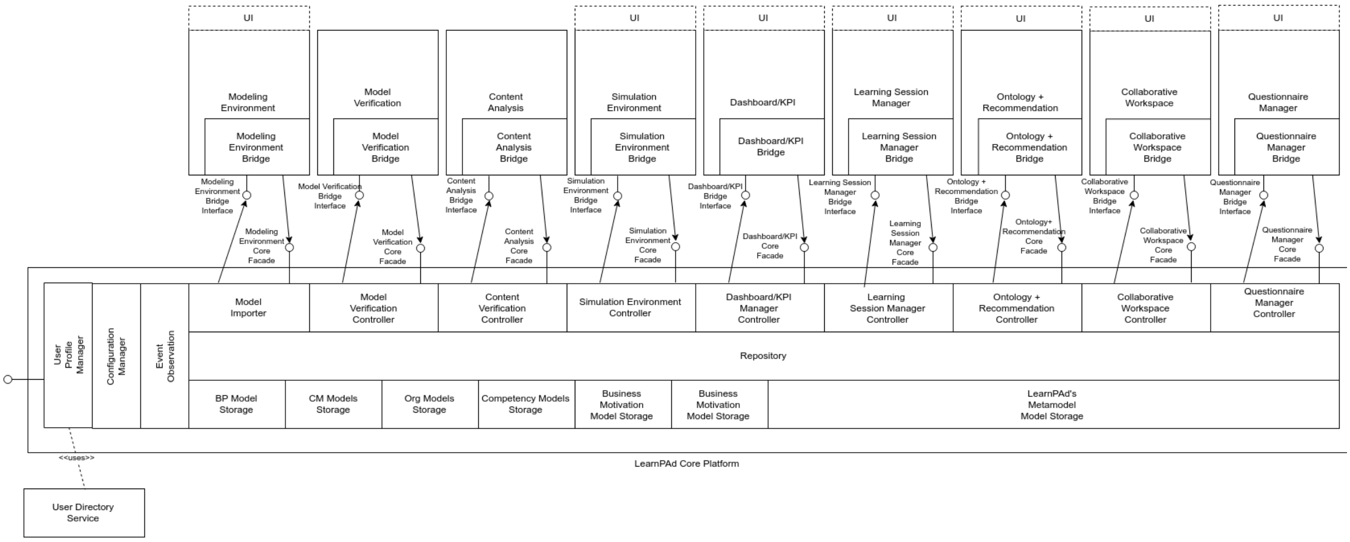


Figure 2: Repository based architecture

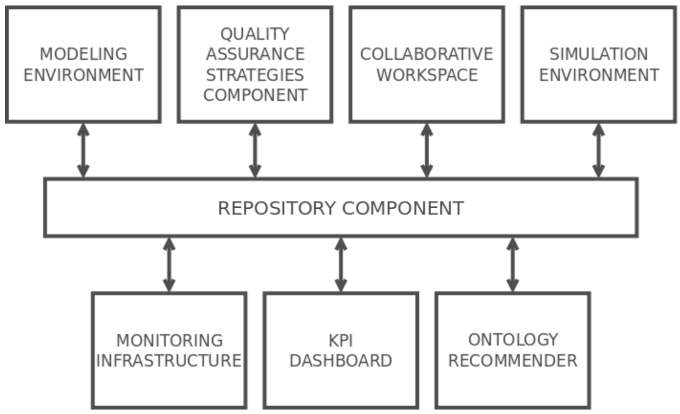


Figure 3: *H2o* architecture

#### 3.2.2 Wolverton suggestion

Wolverton suggest to codify previous experiences in a matrix where rows are *type of components* while columns are related to *difficulty* related to two different factors:

- *Novelty* (with two levels: O,N)
- *Complexity* (with three levels: E,M,H)

Cells contains the historical cost for the development of one *unit of development* for such kind of components

| | Difficulty | | | | | |
|---|---|---|---|---|---|---|
| *Type of Software* | OE | OM | OH | NE | NM | NH |
| Control | 21 | 27 | 30 | 33 | 40 | 49 |
| Input/Output | 17 | 24 | 27 | 28 | 35 | 43 |
| Pre/post processor | 16 | 23 | 26 | 28 | 34 | 42 |
| Algorithm | 15 | 20 | 22 | 25 | 30 | 35 |
| Data management | 24 | 31 | 35 | 37 | 46 | 57 |
| Time-critical | 30 | 35 | 37 | 45 | 55 | 65 |

Figure 4: Exemplificative table

### 3.2.3 The verification component

The platform should include a mechanism enabling the verification of models with respect to structural and temporal properties. In particular it is important that before publishing a model on the collaborative platform a quality check is performed. In case no issues are highlighted the model is published and ready to be used for learning purpose.

The first version of the system is intended to work with BPMN 2.0. In order to permit the formal verification o models it is important to define a precise semantics for BPMN constructs and then to use such semantics in order to carry on the assessment.

The verification components can be decomposed in the following sub components that will use Model based strategies to carry on the transformation:

- BPMN parser
- BPMN translator
- Model verifier

**Let's apply the wolverton strategy!**

## 3.3 Process based estimations

Simple approach that foresees the decomposition of the project in *smaller activities* and the provisioning of estimation for each activity. Data are organized in a matrix in which *rows include the various components* and the *columns the engineering activites*. Each cell will include the Person Months (PM). Such approach is particularly effective in big projects when combined with product decomposition (hybrid approach). In this case the work to carry on is identified in terms of deliverable and components and then activities are identified in order to permit the development of such artefacts.

### 3.3.1 Activities

The division of the project in smaller components enables the refinement of the activities needed to develop the system. At the same time it is necessary to consider effort for the integration of the system. When the project is rather big the activities are generally clustered in workpackages. A **workpackage (WP)** is structured then over a set of tasks that collectively permit to reach the WP objectives.
**The organization of activities, considering dependencies, deadlines etc leads to the definition of a development process for the system!**

For the development of the considered system we could immagine that the following workpackages are needed:

- WP0 – Staffing
  - T0.1 – Recruiting senior staff (2 week)
  - T0.2 – Recruiting junior staff and programmers (4 weeks)
  - T0.3 – Forming junior staff and programmers (4 weeks)
- WP1 – Requirements Elicitation and Analysis
  - T1.1 – Requirements Collection and analysis 1st it. (2 weeks)
  - T1.2 – Requirements Collection and analysis 2nd it. (2 weeks)
  - T1.3 – Requirements Collection and analysis 3rd it. (8 weeks)
  - T1.4 – Requirements maintenance (12 weeks)
- WP2 – Overall Architectural Design and System Integration
  - T2.1 – System decomposition and architectural specification (4 weeks)
  - T2.2 – Integration procedures and mechanisms definition (4 weeks)
  - T2.3 – Integration infrastructure and interfaces definition (4 weeks)
  - T2.4 – System Integration and testing 1st it (12 weeks)
  - T2.5 – System Integration and testing 2nd it (12 weeks)
  - T2.6 – Configuration procedures (4 weeks)
  - T2.7 – System configuration and deployment (4 weeks)
- WP3 – Quality Assurance Strategies

  - T3.1 – Quality assurance strategies definition 1st it (4 weeks)
  - T3.2 – Quality assurance strategies definition 2nd it (8 weeks)
  - T3.3 – Formal Semantics for BPMN (12 weeks)
  - T3.4 – Architectural definition and data formats 1st it (4 weeks)
  - T3.5 – Architectural definition and data formats 2nd it (4 weeks)
  - T3.6 – Quality assurance component implementation 1st it (8 weeks)
  - T3.7 – Quality assurance component implementation 2nd it (16 weeks)
- WP4 – Modeling Environment
  - T4.1 – Modeling environment selection
  - T4.2 – Exporting mechanisms and data formats
  - ...
- WP5 – Collaborative workspace
  - T5.1 – Collaborative workspace funtionality identification
  - T5.2 – Architectural definition
  - T5.3 – Collaborative infrastructure implementation
  - ...
- WP6 – Simulation Environment
  - T6.1 – Simulation strategies for enterprise models
  - T6.2 – Architectural definition
  - T6.3 – Simulation infrastructure implementation
  - ...
- WP7 – Monitoring Infrastructure
  - T7.1 – Monitoring funtionality identification
  - T7.2 – Architectural definition
  - T7.3 – Monitoring infrastructure implementation
  - ...
- WP8 – KPI and Dashboard
  - T8.1 – KPI identification
  - T8.2 – Dashboard funtionality identification
  - T8.3 – Architectural definition
  - T8.4 – Dashboard infrastructure implementation
  - ...
- WP9 – Ontology Recommender
  - T9.1 – Reccomender funtionality identification
  - T9.2 – Architectural definition
  - T9.3 – Ontology and reccomender infrastructure implementation
  - ...
- WP10 – Project Management

The decomposition of the activities permits to apply the "divide et impera" approach to the estimation of the project costs. At first it will be necessary to organize the activities in gantt chart considering possible dependencies among them. The activities have been decomposed using ordering relations (1st, 2nd that in any case represent implicit dependencies). Clearly the organization of the activities is reflected on the process that is adopted for the development of the system. In the following we consider a subset of activities and dependencies.

The following list reports the identified dependencies (" x depends from y" is written as "x ← y") leading to the precedence network of Figure 5:

- T0.3 ← T0.2
- T1.1 ← T0.1

- T2.1 ← T0.1
- T1.4 ← T1.3 ← T1.2 ← T1.1
- T2.3 ← T2.1 ← T1.1
- T2.5 ← T2.4 ← T2.3 ← T2.2
- T3.2 ← T3.1 ← T0.3
- T3.3 ← T0.1
- T2.5 ← T3.6
- T3.7 ← T3.6 ← T3.4 ← T3.1
- T3.7 ← T3.3
- T2.4 ← T2.6
- T2.5 ← T3.6
- T2.7 ← T2.6
- T2.7 ← T2.5
- T2.7 ← T3.7
- T3.7 ← T1.4
- T3.7 ← T3.2

Using the precedence network we can apply forward and backward passes to derive suitable earliest and latest starting dates, as well as earlist and latest end dates. The activities for which the two set of dates coincide are on the critical path. Activties on the critical path will have a *float value* equal to zero. Figure 6 reports the net after the forward and backward passes.

# 4 Risk Management

Risk management is *one of the most important activity* in project planning. It has to do with

- *anticipating risks* that might affect the project schedule, budget, or quality of the the software being developed (identify *cause-effect relations*)
- *taking actions* to "handle" the risks
- *documenting* the risks in the plan

Effective risk management makes it *easier to cope with problems* and to ensure that these do not lead to unacceptable effects. Indeed every *plan is based on assumptions* and *risk management tries to plan for and control the situations where those assumptions become incorrect*

## 4.1 Risk categories

- *Project risks*: affect the project schedule or resources
  - *Staff turnover*: experienced staff will leave the project before it is finisched
  - *Management change*: There will be a change of organizational mgmt with different priorities
  - *Requirements change*: There will be a larger number of changes to the requirements than anticipated
- *Product risks*: affect the quality of the of the software being developed
  - *Specification delays*: Specification od essential interfaces are not available on schedule
  - *CASE tool underperformance*: CASE tools which support the project do not perform as anticiapted
- *Business risks*: affect the business perspectives for the organization developing or procuring the software
  - *Technology change*: The underlying technology on which the system is built is superseded by new technology
  - *Product competition*: A competitive product is marketed before the system is completed

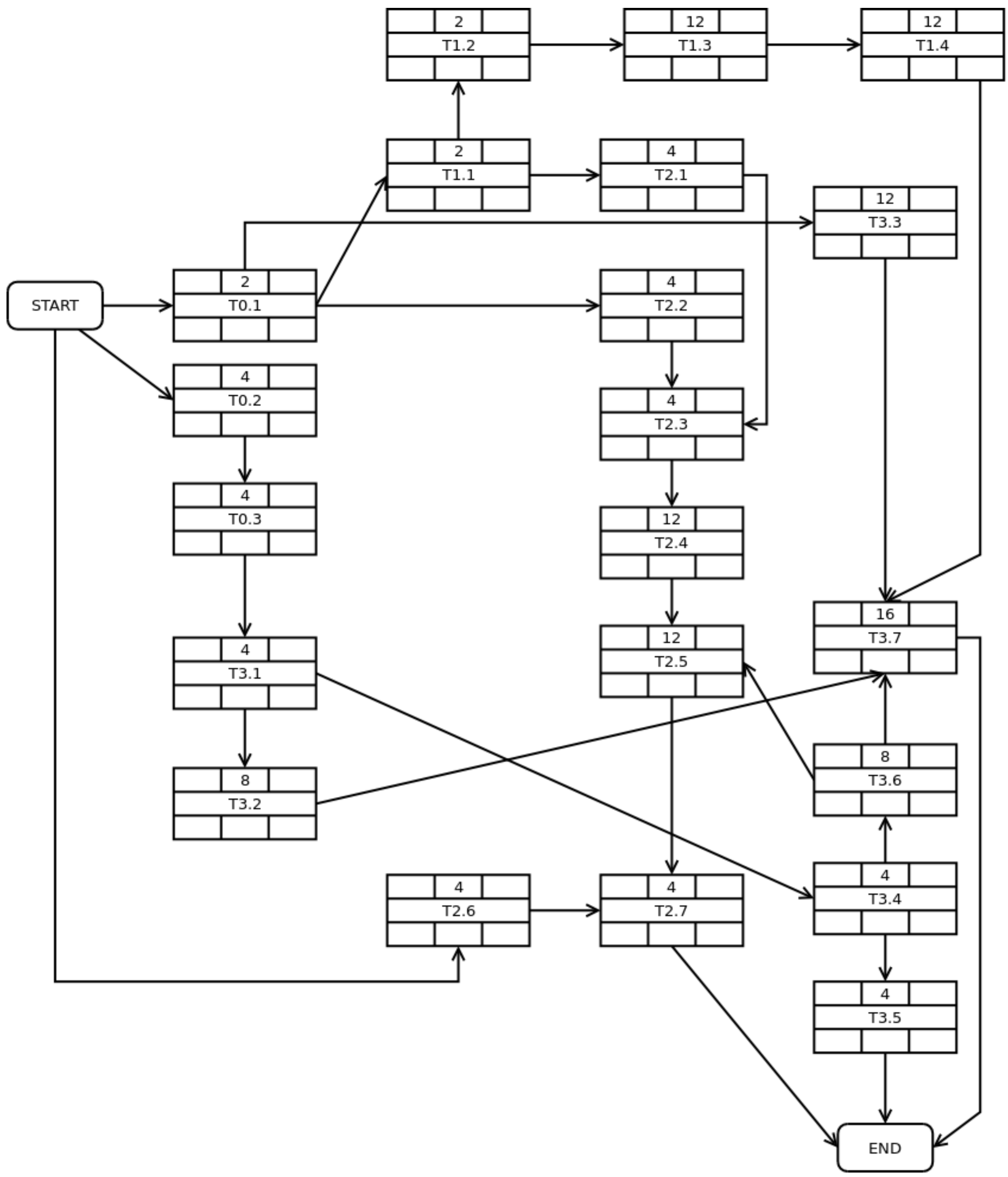


Figure 5: Precedence network

Clearly there is not a clear cut among the different categories
**Given the considered project which are the risks that you would consider more luckily?**
Two main techniques can be used to identify risks:

- checklists
- brainstorming

## 4.2 Checklist

In Table 3 is reported the list proposed by Barry Boehm for the list of 10 most recurring risks for software development.

## 4.3 Brainstorming

Questions to drive brainstorming

1. Have top software and customer managers formally committed to support the project?
2. Are end users enthusiastically committed to the project and the system/product to be built?

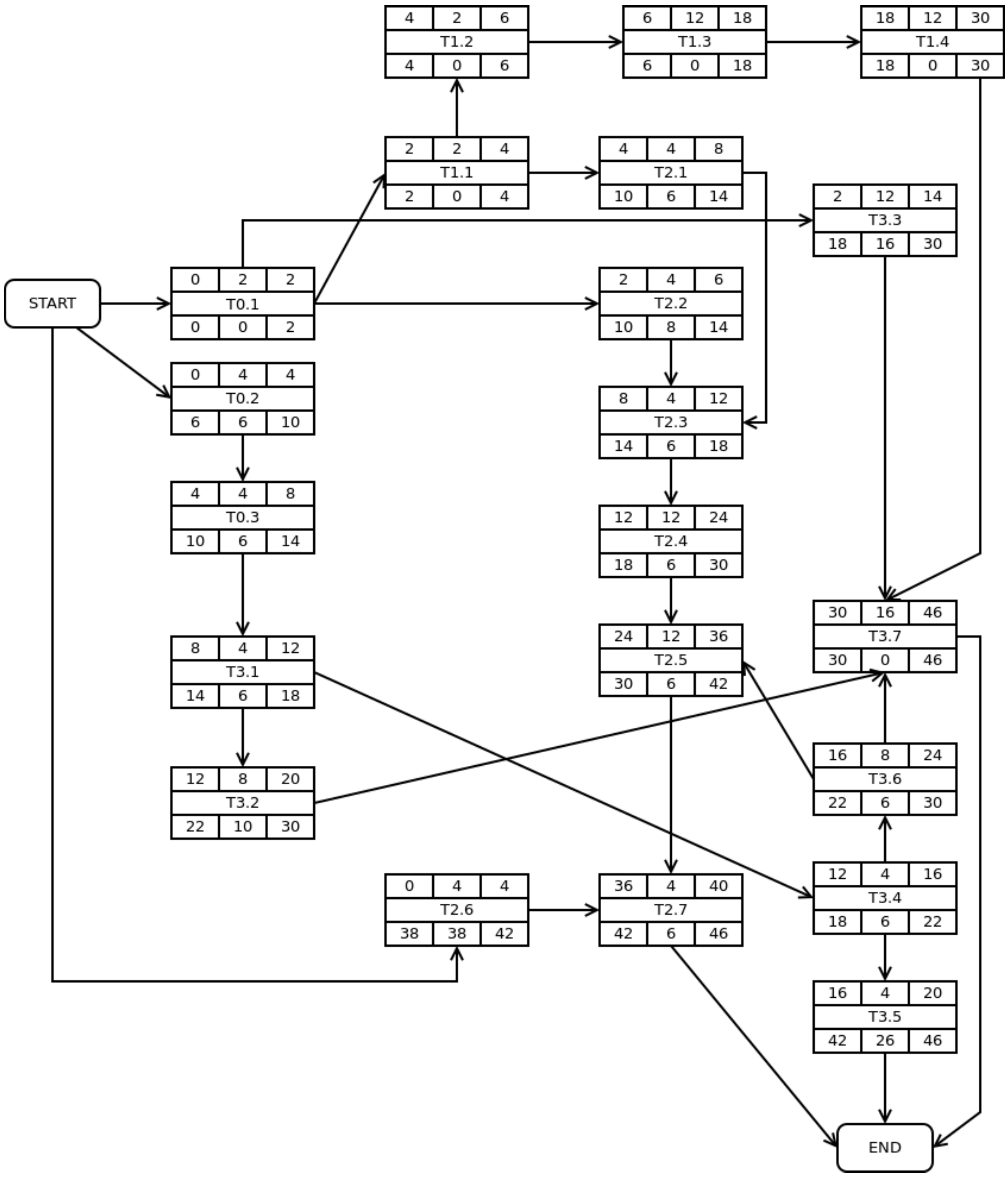


Figure 6: Precedence network after forward and backward passes

3. Are requirements fully understood by the software engineering team and its customers?
4. Have customers been involved fully in the definition of requirements?
5. Do end users have realistic expectations?
6. Is the project scope stable?
7. Does the software engineering team have the right mix of skills?
8. Are project requirements stable?
9. Does the project team have experience with the technology to be implemented?
10. Is the number of people on the project team adequate to do the job?
11. Do all customer/user constituencies agree on the importance of the project and on the requirements for the system/product to be built?

If any one of these questions is answered negatively, mitigation, monitoring, and management steps should be instituted without fail.

**Top ten strategy** Boehm suggests to derive and consider only the *10 top most dangerous risks*

| **Risk** | **Risk reduction techniques** |
|---|---|
| Personnel shortfalls | Staffing with top talent; job matching; teambuilding; training and career development; early scheduling of key personnel |
| Unrealistic time and cost estimates | Multiple estimation techniques; design to cost; incremental develop.; recording and analysis of previous project; standardiz. of methods |
| Developing the wrong software functions | Improved software evaluation; formal specification; user surveys; prototyping; early user manuals |
| Developing the wrong user interface | Prototyping; task analysis; user involvement |
| Gold plating | Requirements scrubbing; prototyping; cost-benefit analysis; design to cost |
| Late changes to requirements | Stringent change control procedures; high change threshold;incremental development |
| Shortfalls in externally supplied components | Benchmarking; inspections; formal specifications; contractual agreements; quality assurance procedures and certification |
| Shortfalls in externally performed tasks | Quality assurance procedures; competitive design or prototyping; contract incentives |
| Real-time performance shortfalls | Simulation; benchmarking; prototyping; tuning; technical analysis |
| Development technically too difficult | Technical analysis; cost-benefit analysis; prototyping; staff training and development |

Table 3: Boehm's categories for project risks, and possible reduction techniques

### 4.4 Risks to the schedule

The PERT strategy requires three different estimations for each activity in order to derive an estimation for the expected duration ($t_e$):

- *Most likely time* ($m$)
- *Optimistic time* ($a$)
- *Pessimistic time* ($b$)

The expected duration is then computed as:

$$t_e = \frac{a+4m+b}{6}$$

Computing the *standard deviation (s)* it is possible to derive a quantitative measure of uncertainty. For the purpose of the study we need to conduct the standard deviation can be computed using the following formula:

$$s = \frac{b-a}{6}$$

$s$ can be used as a ranking *measure of the degree of uncertainty* or risk for each activity. PERT diagrams can be fruitfully used to derive the likelihood of meeting targets. Figure 8 shows the PERT diagram derived using the data reported in Figure 7.
Successively after having filled the PERT diagram as reported in Figure 9 it is possible to derive the likelihood of meeting targets. For such a case it is necessary to derive the $z$ value.

$$z = \frac{T-t_e}{s}$$

The probability of meeting a target should be derives from the values of the function plotted in Figure 10.

| Activity (Precedents) | Activity Durations (weeks) | | | | |
|---|---|---|---|---|---|
| | Optimistic (a) | Most likely (b) | Pessimistic (b) | Expected te | Standard deviation (s) |
| T0.1 | 1 | 2 | 4 | 2,16 | 0,5 |
| T0.2 | 3 | 4 | 8 | 4,5 | 0,83 |
| T0.3 (T0.2) | 2 | 4 | 6 | 4 | 0,66 |
| T1.1 (T0.1) | 1 | 2 | 4 | 2,16 | 0,5 |
| T1.2 (T1.1) | 1 | 2 | 4 | 2,16 | 0,5 |
| T1.3 (T1.2) | 4 | 8 | 13 | 8,16 | 1,5 |
| T1.4 (T1.3) | 8 | 12 | 17 | 12,16 | 1,5 |
| T2.1 (T1.1) | 3 | 4 | 5 | 4 | 0,33 |
| T2.2 (T0.1) | 3 | 4 | 4,5 | 3,91 | 0,25 |
| T2.3 (T2.1,T2.2) | 2 | 4 | 6 | 4 | 0,66 |
| T2.4 (T2.3) | 10 | 12 | 14 | 12 | 0,66 |
| T2.5 (T2.4,T3.6) | 14 | 18 | 22 | 18 | 1,33 |
| T2.6 | 3 | 4 | 7 | 4,33 | 0,66 |
| T2.7 (T2.5,T2.6) | 3 | 4 | 6 | 4,16 | 0,5 |
| T3.1 (T0.3) | 2 | 4 | 7 | 4,16 | 0,83 |
| T3.2 (T3.1) | 7 | 8 | 9 | 8 | 0,33 |
| T3.3 (T0.1) | 8 | 12 | 13 | 11,5 | 0,83 |
| T3.4 (T3.1) | 3 | 4 | 5 | 4 | 0,33 |
| T3.5 (T3.4) | 3 | 4 | 5 | 4 | 0,33 |
| T3.6 (T3.4) | 6 | 8 | 11 | 8,16 | 0,83 |
| T3.7 (T1.4,T3.2,T3.3,T3.6) | 12 | 16 | 19 | 15,83 | 1,16 |

Figure 7: Activity time estimates

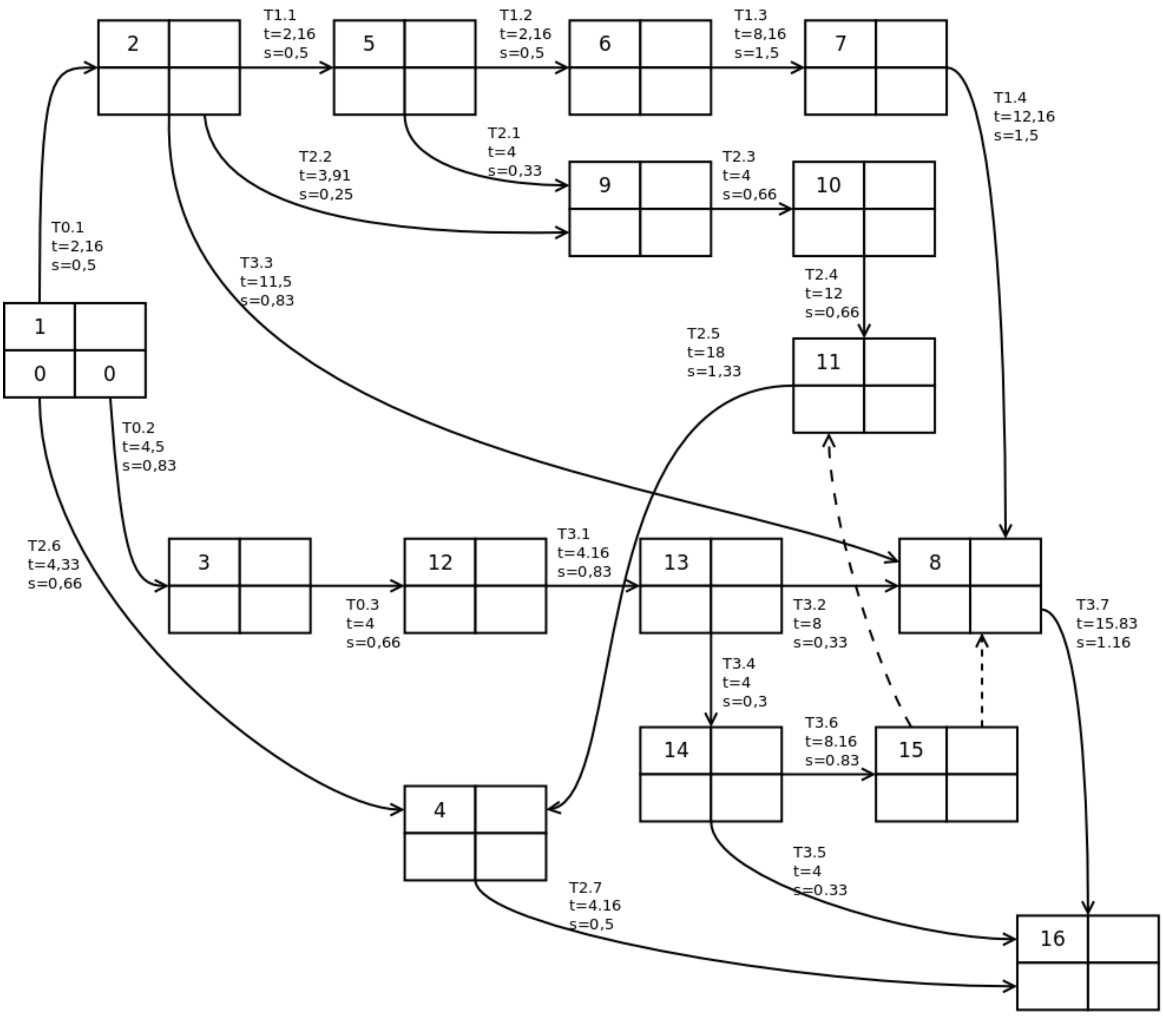


Figure 8: PERT network

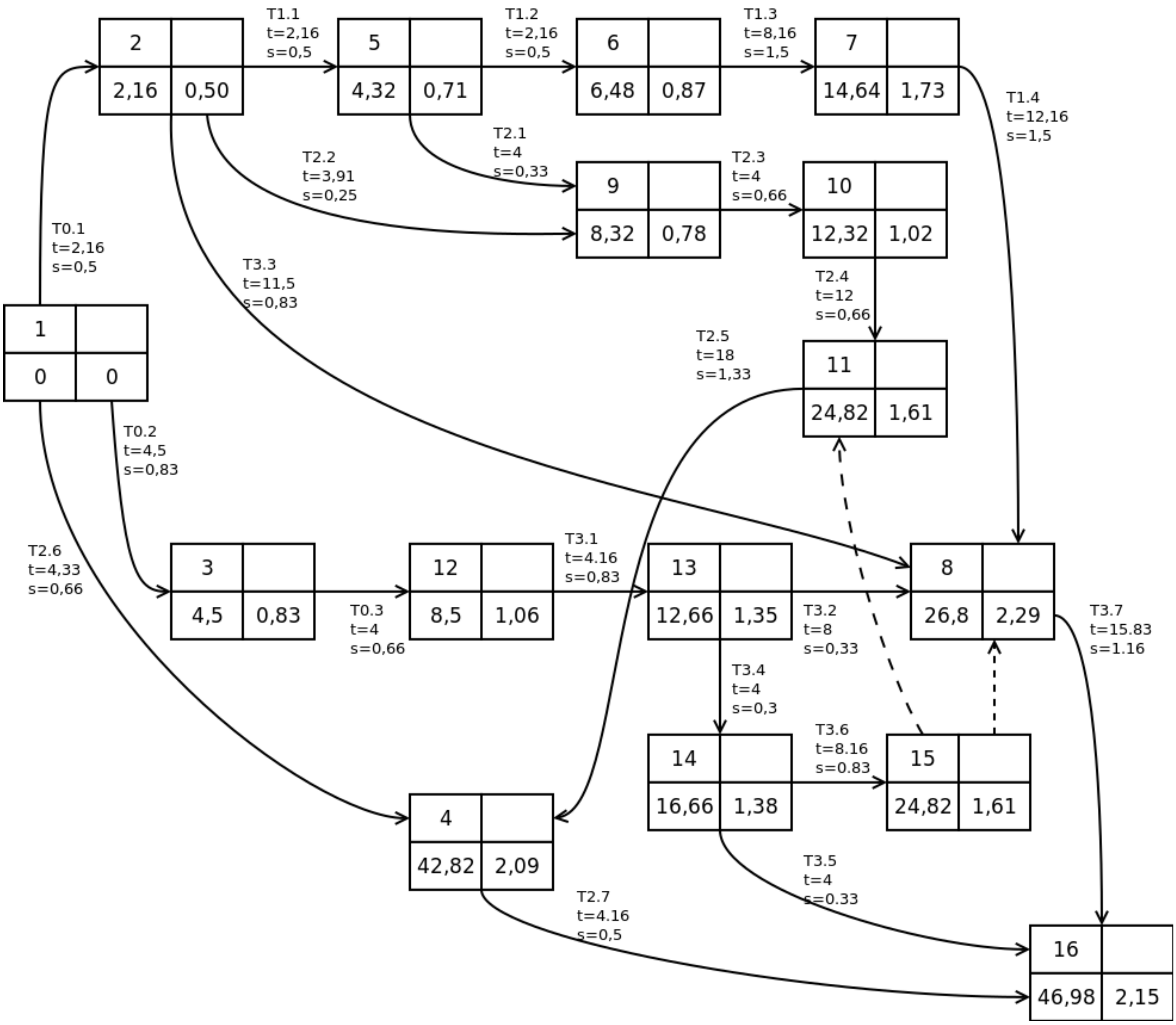


Figure 9: PERT network after a forward pass and calculation of the standard deviation

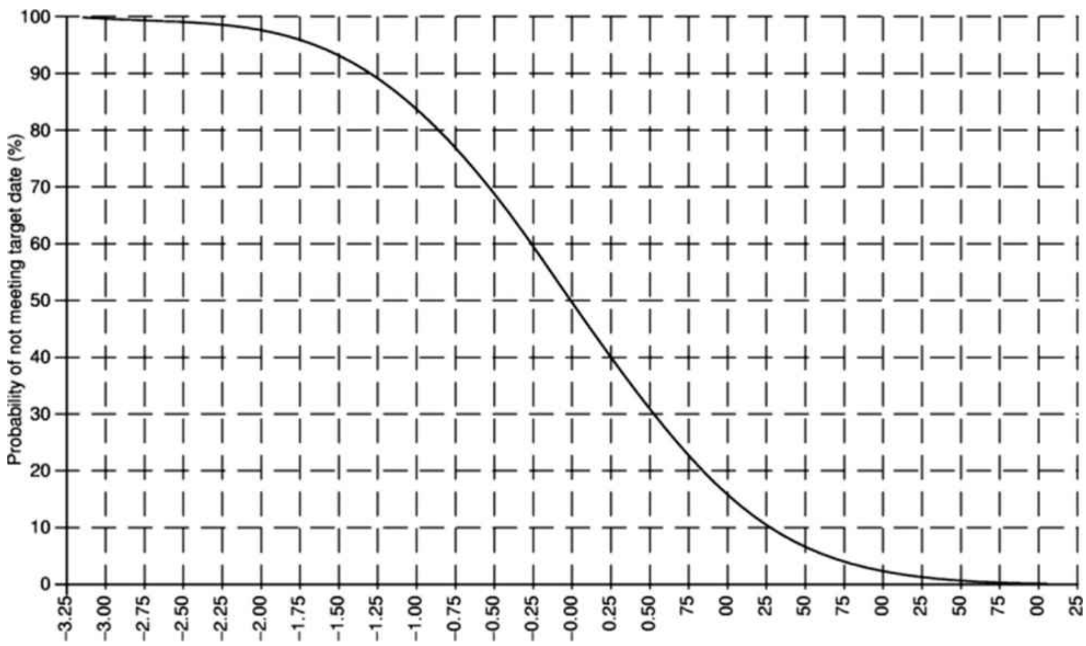


Figure 10: z for a normal distribution

# 5 Resource Allocation

## 5.1 Introducing resources in planning

- Plans discussed so far somehow assumed that *resources where not limited* and no *constraints on their usage are considered*

### 5.1.1 Resources

A *resource* is *any item or person* required for the execution of the project

When considered in planning (*resource allocation*) they result in the definition of a number of scheduling:

- *activity schedule*
- *resource schedule*
- *cost schedule*

## 5.2 Typical resources

Resources can refer to:

- *Labour* to perform project activities - full assignment vs. partial assignment to the project
- *Equipment* to be used to permit the work of labour (desks, workstations ...)
- *Materials* to be consumed during the project
- *Space* in particular if additional personnel units have to be recruited
- *Services* to be used during the project (e.g. confcall services)
- *Time*
- *Money* resource needed to retrieve other resources

## 5.3 Resource requirements

To identify required resources we need to consider each single activity in the plan and define which are the needed resources

| Stage | Activity | Resource | Days | Quantity | Notes |
|---|---|---|---|---|---|
| 1 | | Project Manager | 65 F/T | | |
| | All | Workstation | | 17 | Check for OS licenses |
| 2 | Planning | Senior Analyst | 7 F/T | | |
| ... | ... | ... | ... | ... | ... |

The allocation of resources to activities will lead us to review and modify the ideal activity plan, in particular to take into account the actual availability of needed resources. This activity can be fruitfully carried on using a spreadsheed in which the results of the analysis on the activity network are reported. Figure 11 reports, in a Gantt like format, the information derived from the CPM. The spreadsheed can be used to allocate resources and then to possibly revise the plan to possibly obtain more effective resource allocation plans. In particular Figure 12 reports the value for an allocation of a interchangeable resource which then will not introduce additional constraints.

The initial allocation of resources suffers from the presence of peaks and valley which will make the execution of the project not really efficient, requiring the repetitive allocation and dealloation of resources to the project. In Figure 13 a revised allocation is proposed. The novel allocation has been derived considering all the resources as intecheangeable and also considering the possibility of splitting the activities that, even though they complexively require the same amount of resource. As it can be observed the new allocation of resources does not lead to peaks and valley, and also the maximum request of resources at teh same time has been drastically decreased from 11 to 7. The new allocation can then be considered quite good, given the time dependency constraints among the activities, since the amount of resources required is increasing at the beginning and then decreasing toward the end, without any inversion in the trend. This will permit

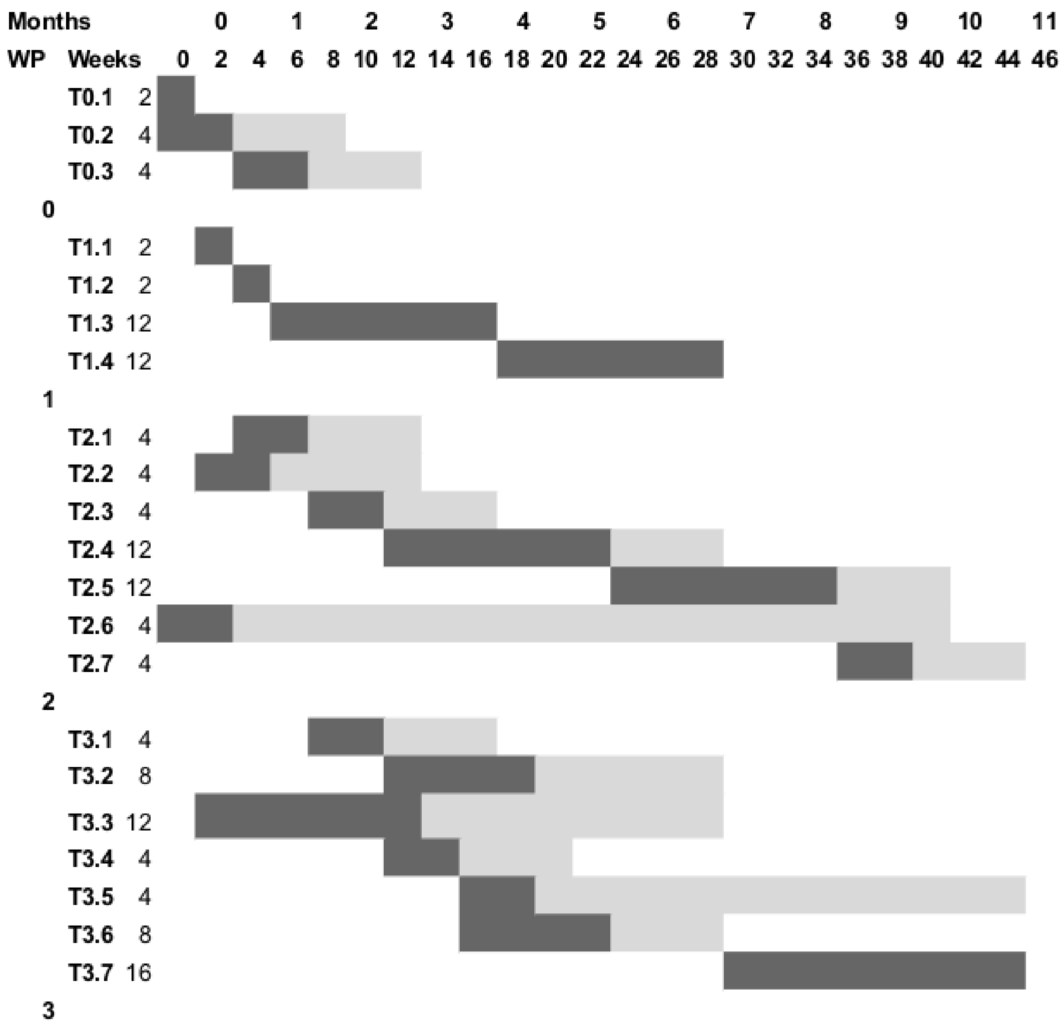


Figure 11: Gantt chart for the project with indication of float

a quite good allocation of resources to the project with no "dead periods" in which is difficult to allocate the resource to other projects, reducing then the overall efficiency of the organization.

The resource allocation reported in Figure 13 has been derived under ideal conditions, i.e. in a situation in which allocation did not introduce any further constraints. This is consequence of the fact that all resources are considered interchangeable. In the more realistc case in which some resources have an identity it is possible that their allocation introduces additional constraints that have to be taken into due account in the resulting plan.

In Figure 13 it is reported the extimation for the whole project. The estimation has been derived multiplying the total amount of resources needed by the cost per unit of resource. Having this information it shouldn't be difficult to derive also a graph for hte cash flow needed by the project. Finally it is worth noting the the allocation procedure considered the data derived applying the Critical Path Method. No evaluations of risks are then included in the resulting plan. It is clearly possible to include also the evaluation of risks, for instance taking into account the data coming from the analysis carried on using the PERT based method.

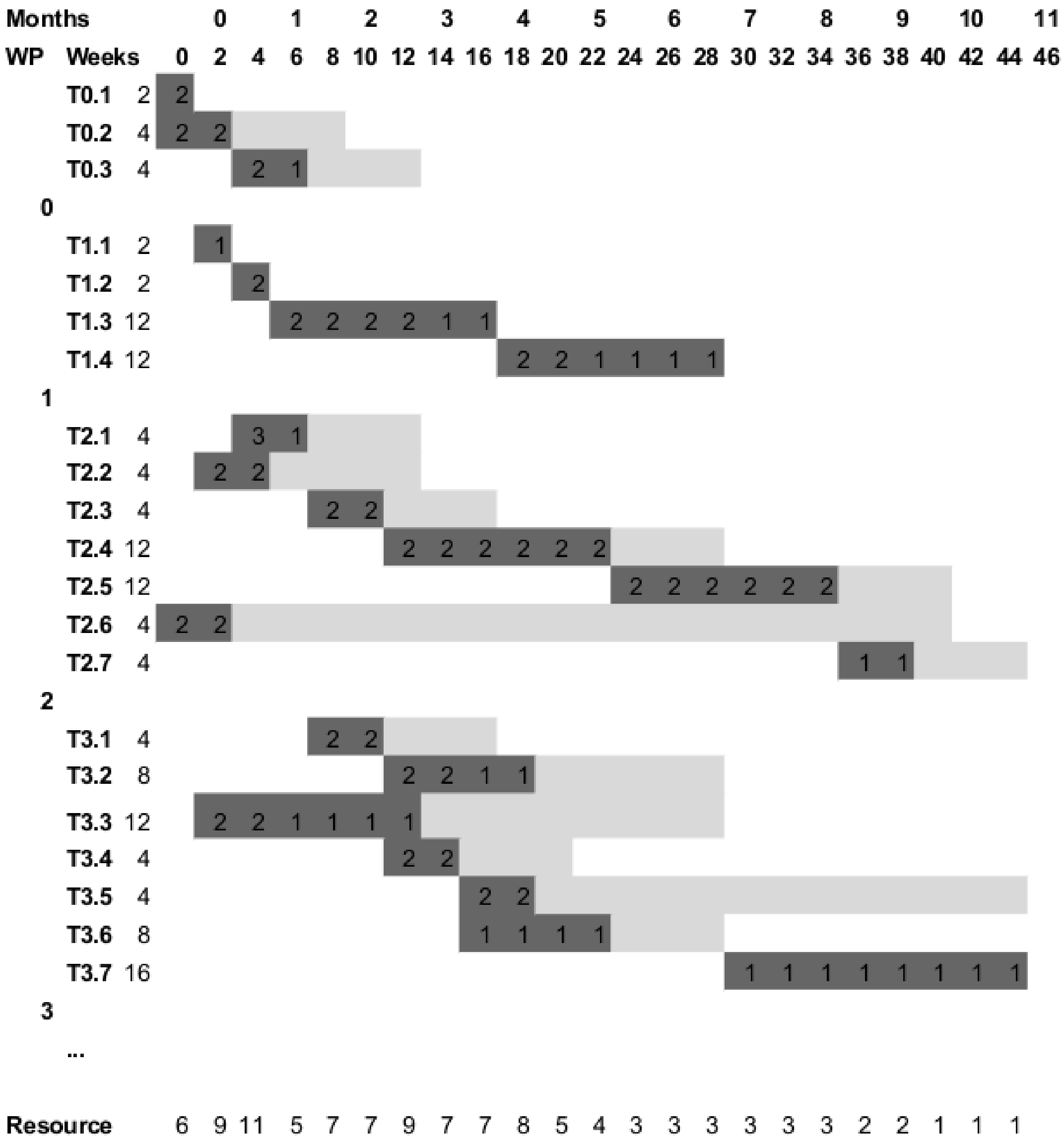


Figure 12: Resource allocation - first iteration

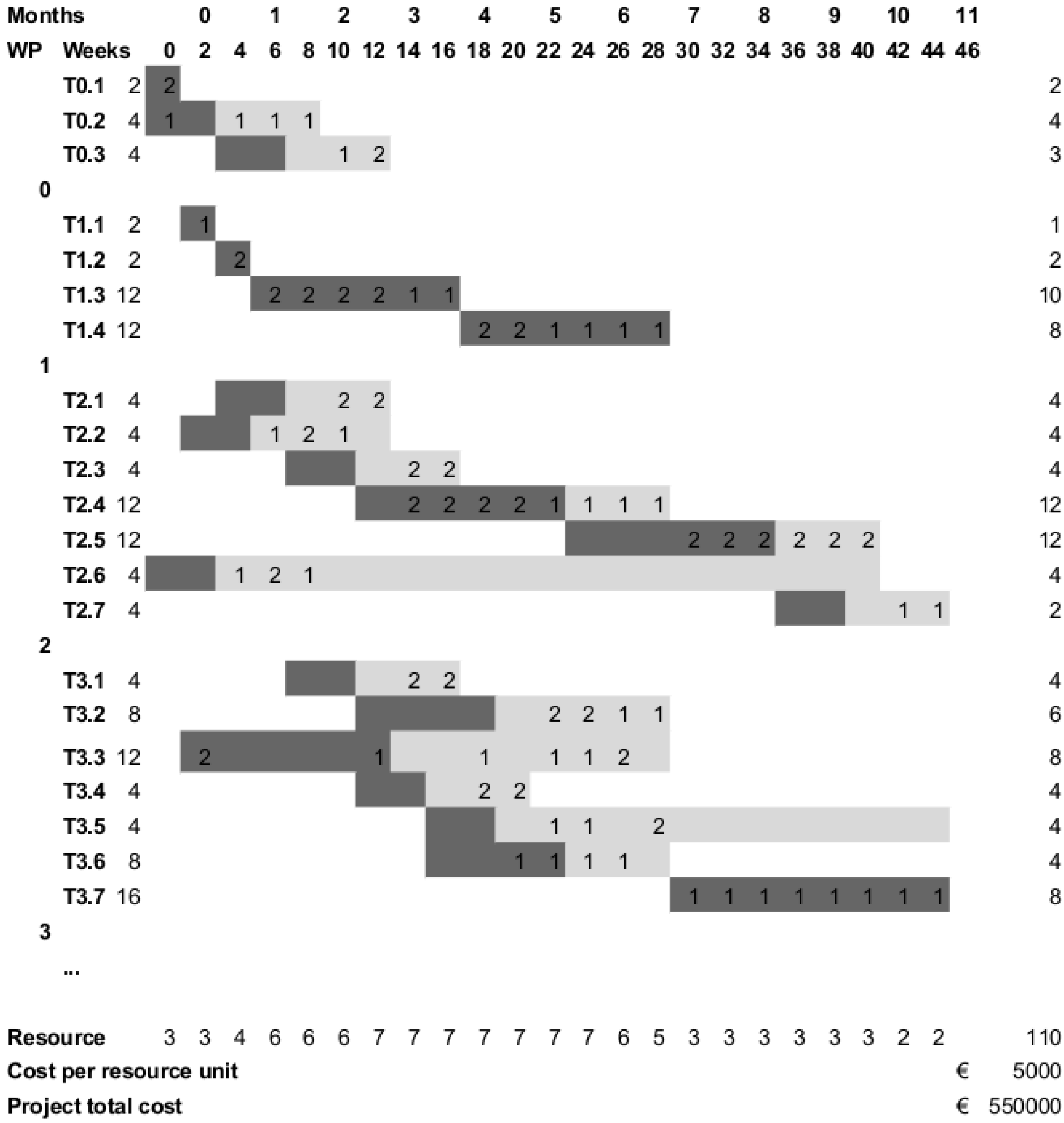


Figure 13: Resource allocation - second iteration

# References


[1] A. Bertolino, G. De Angelis, A. Di Sandro, and A. Sabetta. Is my model right? let me ask the expert. *Journal of Systems and Software*, 84(7):1089 – 1099, 2011.

[2] Frank Budinsky, Dave Steinberg, Ed Merks, Ray Ellersick, and Timothy J. Grose. *Eclipse Modeling Framework*. Addison Wesley, 2003.

[3] Frédéric Jouault, Freddy Allilaire, Jean Bézivin, and Ivan Kurtev. Atl: A model transformation tool. *Science of Computer Programming*, 72(1-2):31–39, 2008.

[4] Dimitrios S. Kolovos, Richard F. Paige, and Fiona A. Polack. The epsilon transformation language. In *Proceedings of the 1st international conference on Theory and Practice of Model Transformations*, ICMT '08, pages 46–60, Berlin, Heidelberg, 2008. Springer-Verlag.

[5] Javier L. C. Izquierdo, Jesús S. Cuadrado, and Jesús G. Molina. Gra2MoL: A domain specific transformation language for bridging grammarware to modelware in software modernization. In *Proceedings of the Workshop on Model-Driven Software Evolution*, 2008.