

## Modeling of software project

Alojz Masar<sup>1+</sup>, Pavol Tanuska<sup>1</sup> and Renata Masarova<sup>1</sup>

<sup>1</sup> Faculty of Materials Science and Technology in Trnava, Slovak University of Technology in Bratislava

**Abstract.** Approaches to the investigation of software development lifecycle are mostly based on empirical observations, statistical data and people experiences. Results of these investigations come under different areas. Different conditions of each investigation and dissimilarity of the result's areas makes comparison and categorization of these results very tricky. We introduce a model of the software development based on the principles of system theory with respect particular features of software project and specific requirements of practical use.

**Keywords:** engineering, model, project, management

### 1. Introduction

The term engineering project can be seen from different points of view - the project which is focused on engineering design or organization of engineering design work that is overseen by a project manager. Software projects fulfil both of these views. This kind of project is realized mainly by technical staff, but project management is not taken as an engineering discipline. This contradiction leads to some misunderstandings between project managers and realization teams. Achieving a common view is crucial for the success of the project.

Project management and software engineering are widely described. Project management is the one of the oldest areas of people interest. PMBOK [1] and PRINCE 2 [2] represent accepted standards of it. Management of software producing work is a particular area between management and software engineering. It is described by a range of methodologies or approaches, i.e. waterfall [4], RUP [5], RAD [6] or agile approach [7]. The software engineering represents technical approach to the software development. This area is the most coloured by diversity of technologies and technical factors. Summarization of its principles can be found in different publications, i.e. Software Engineering: A Practitioner's Approach [8] or Software Engineering [9].

#### 1.1. Overview of requirements

Interconnecting such different areas is not easy task. Understanding on a vague level is usually not so difficult. Problems grow from details. Many terms have shifted meaning in each of the mentioned areas; some of them are specific for one area and some are interpreted in a totally different way. An example of this is the understanding of the term task. Task represents amount of required or completed work. Project managers are considering work of project team. On the other hand, a developer naturally refers to the tasks carried out by software system. Therefore clearly expressing a semantic context is very important.

Terms are usually defined by words. But different languages and environments use words in a different way. Another way to express facts and ideas is to draw them. Pictures are universal language around the world. But depicting details can lead to very complex drawings and can take a long time. Context

---

<sup>+</sup> Corresponding author. Tel.: + 421 903 754 002; fax: +421 906 068 299.  
E-mail address: masara@stonline.sk.

represented by main ideas and facts has to be expressed graphically and details by words. These combinations lead to understandable and relatively simple graphical overview and details expressed by words embedded to this frame have clearer meaning.

The knowledge about project and its outcome (software) is superficial at the beginning. Over time, this knowledge deepens. The model has to reflect this process. It must provide instruments to express project on different levels of knowledge and details.

The main goal of the model is to achieve a common view of the project. Therefore used concepts have to be understandable by each group participating on the project in short time.

## 2. Model

Software project in fact fulfil the definition of a system. It has inputs and outputs, control, processor and environment. It deals with feedback and must have a defined interface (boundary) for proper operation. Because it interacts with the environment, it is an open system. It can adapt to changes in the environment and tries to affect it. The project consists of at least two elements which have a common objective (project manager and team of specialists). They react to each other and communicate with each other. One of them has the control function. Therefore the project is self organized system as well. The project as a whole brings a new quality, which cannot be achieved by the separate elements. This indicates that the project is a complex system.

The concept of a control system is usually known and clear for all professionals working in the system control area. Therefore it is a perfect unifying element of user's, project manager's and developer's viewpoints. This concept allows us utilize a broad range of experiences, which have been gained in modelling systems area. But project has some specific features. Therefore we have to point them out and take them into account.

### 2.1. Information

Every group participating on the project has its own responsibility, knowledge and sphere of activity. The glue, which connects them and allows their cooperation, is information. The information on the project is occasionally simple. Usually it is compound. Therefore we set the type of information by its main meaning.

We distinguish information, which type is unknown or not expressed. Typed information are:

- Announcement – information related to activities or facts without any other meaning. We define particular announcements related to errors and suggestions. Suggestion does not contain obligation. It is only recommendation. Announcements may not have one of these subtypes (i.e. reports).
- Directive – instruction with obligation
- Artifact – information related to the outputs produced by activities of the project. It describes some outputs or represents them.

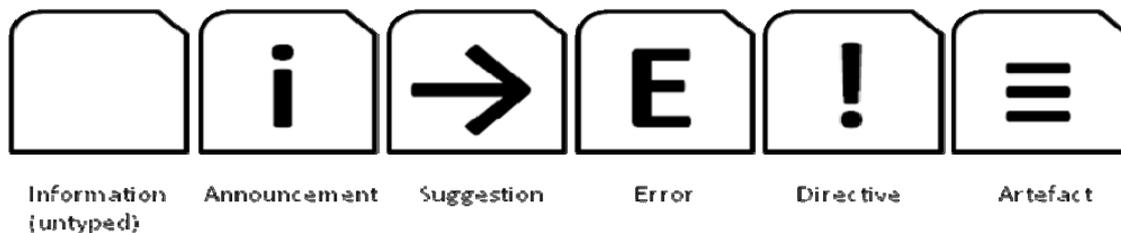


Fig.1: Graphical representation of information types

### 2.2. Components

The elements of the systems realize activities. The activities of the project are realized by people or by tools. The important difference between them lies in uncertainty. The tools are created to be deterministic. They produce the same result at the same conditions. The result of human work is only predictable but the result is not guaranteed.

People usually work in groups (teams) on the project. The tools usually realize set of activities and are assembled from multiple parts. Therefore the term element is little bit confusing in this context. We prefer the word component. Component represents one person, one simple tool or group of them and acting as a whole. A component can contain other components. So the component is like a subsystem of the system (subproject). This allows modelling elements of the project on different levels of abstraction or knowledge.

Components are characterized by the type of duty they have in the project, by their relationships with other components or with the environment and can be characterized by their individual structure and organization. The component has only one type of the duty. The project must be compound at least by one managing component and one process component. Of course, it may contain additional components of these or the other types.

The component can have named values assigned to it. These properties express either the connection of the component with modelled reality or level of competence of the component to perform its tasks. The property is an ordered pair of name and value. The type of value is not restricted. The property describes details of the element. Every element has properties name and description. The name is the unique name of the element in the model and the description is textual depiction of the element. The properties can be expressed graphically as well. The name is drawn in or near by the element as a simple text. Others can be drawn near or in the corresponding element as name:value pairs. For example, the physical location of specialists' team or performance of this team can be expressed by the properties. Important property of the component is its ability to perform its tasks. It is expressed by the productivity of the component, which is a positive number. The value 0 expresses that the component is not able perform any task. The value 1 means that the component consumes all its possibilities for the tasks. Value greater than 1 indicates that the component is able to perform extra tasks.

The managing components manage the other components, process information produced by other components about their activities and produce information for other managing components or send them to the environment. The duty of process (or execution) components is mainly related to the produced software. They process inputs related to the produced software, transform them to the (partial) results by the managing instructions from managing components, send results to the other components or to the environment and produce information about their activities for the managing components. The process component can manage only itself. Therefore they are like a black box from this point of view. The validation components are responsible for checking or validating outputs of the other components and for reporting defects, deviations or lack of conformity with the expected results. They never produce results or manage other components.

Type/ Behavior	Unknown	Managing	Process	Validating
Unknown	Component (general)	Managing component 	Process component 	Validating component 
Certainty	Component certainty 	Managing component 	Process component 	Validating component 
Uncertainty	Component uncertainty 	Managing component 	Process component 	Validating component 

Fig. 2: Graphical representation of component

### 2.3. Communication

Components communicate with each other or with the environment. Two components are in a relationship if they can communicate with each other. An element of the environment may or may not be determined. The element of the environment is not a component. It is meant as a property of the communication or just described by the communication. The source of the communication is the starting

point of communication. It initiates communication because it needs to send information. The target of communication receives sent information. Both, the source and the target of communication, are the component or environment.

The communication has two aspects. The communication channel determines the way and the direction of the communication. The transmitted information represents the content of the communication. The information is always created by the component or by the environment. The communication, or more precisely the channel, is unidirectional. This means that communication (transmission) occurs only in one direction. This restriction is not practical if the two components need to communicate on both sides. For simplicity, we say that the communication is bi-directional if the two communications between the same components use identical types of channels and transmit the same type of information. One piece of information can be transmitted from one component only by one channel at one time. But the same kind of information can be transmitted by different types of channels. Therefore, it is clear that multiple channels can be assigned to one communication, but only one of them can be used at one time. All channels of the communication must have the same direction.

Depending on whether the target component must be ready to receive the information, the communication is direct or postponed. The target component must be ready for direct communication at the moment the information is sent. For postponed communication, the information will be delivered as soon as the target component will be ready to receive it. If the target component is ready at the moment of dispatch, the transmitted information is delivered immediately. An example of the postponed communication may be a mail communication. The post office is the channel and a letter is the information. An example of the direct communication is a telephone call. A telephone line is the channel and content of the conversation is the transmitted information. At this moment we disregard the fact that the call can be realized only with approval of the recipient.

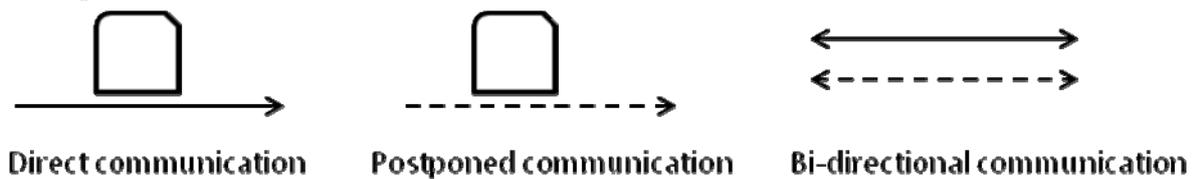


Fig. 3: Graphical representation of communication

Modelling communication between several components only by binary relations is practicable, but not practical. Therefore we define a communication point. The communication point connects multiple channels that are each associated with one (miscellaneous) component. These channels have to be the same sort and they must be able to transmit the same form of information. We define two types of communication points. The distribution point allows the transmission of the same information from one source component to multiple target components. Therefore, communication realized by the distribution point is one-way communication. There can be multiple source components for one distribution point, but only one can send information at one time. The source component may also be a target element. Then there must be an incoming communication from the distribution point. This type of communication point is used only for postponed communication. Contrary, the meeting point is used for direct communication. This point connects multiple bi-directional channels. The information transmitted thru this point is immediately delivered to each connected component. If any target component is not ready, the transmitted information is never delivered to this component.



Fig. 4: Graphical representation of the communication points

## 2.4. Information

We have mentioned that the type of information is defined by the meaning of the information. On the other hand, the information embodies two constituents. The form is a perceptible or electronic representation of the information. It may be sound, image, physical or electronic document, etc. The message is an object of information. The message of the information can be empty. The form can be unknown.

Two pieces of information can be composed together. This can be done either by merging the messages and creating a common form or by creating new information with two independent parts. The example of the first method is merging of two files. A compact disc burned with these files is the example of the second one.

The information can be transmitted only if the form of the information and the channel are compatible. Sometimes it is better not to be too strict. Therefore the model defines a universal form. This kind of the form is compatible with each type of the channel. This simplification is advantageous even if the form does not matter to us.

## 3. Conclusion

The proposed model allows us to express the project from an unusual perspective. This approach permits utilizing well known results of control system theory and project management area. The elements are designed to allow their mathematical formalization. This opens the door to transferring a modelled project to the more rigorous form. The model even allows for a comparison between the project and the developed software systems which are modelled in a similar way.

The main advantage of our model is including uncertainty. This feature allows expressing randomness or doubts in a vague manner. Together with possible formalizations it offers a possibility to involve advanced mathematical tools like stochastic processes.

We are convinced that this approach leads to a more systematic examination of projects and to a better understanding of findings. It naturally divides the study of objective characteristics of the projects which are expressed (directly or indirectly) in a structure of model elements and subjective characteristic which are usually expressed by the named properties of these elements. This paper presents the base principles only. There is a lot of work to create the fully featured modelling system. Improvement of the capabilities and explanatory power of the model can be achieved by elaboration of metric system and by specification of element properties. We are convinced that examination of models of known cases or methodologies will lead to confirm some of the known empirical results on a more abstract basis. We hope that this way, new coherences between projects and producing software systems will be uncovered.

## 4. References

- [1] *A Guide to the Project Management Body of Knowledge*. Project Management Institute, 2008
- [2] *Managing Successful Projects with PRINCE 2, 2009 edition*. TSO, 2009
- [3] ISO/IEC 12207:2008(E) *Systems and software engineering — Software life cycle processes*. International Organization for Standardization, 2008
- [4] W. Royce. Managing the development of large software systems. In: *Proceedings of IEEE Wescon. 1970*, pp. 382-338.
- [5] P. Kruchten. *The Rational Unified Process: An Introduction*. Addison Wesley, 2003
- [6] J. Martin. *Rapid Application Development*. Macmillan, 1991
- [7] A. Cockburn. *Agile Software Development*. Addison-Wesley, 2001
- [8] R. S. Pressman. *Software Engineering: A Practitioner's Approach*. McGraw Hill, 1997
- [9] I. Sommerville. *Software Engineering (8th Edition)*. Addison Wesley, 2006
- [10] M. Fowler and K. Scott. *UML Distilled Applying the Standard Object Modeling Language*. Addison Wesley, 1997