

Marijan Krašna

Ivan Rozman

Faculty of Electrical Engineering
and Computer Science Maribor

Laboratory for Information Technologies

UDK: 681.32.06

Original Scientific Paper

Understanding Processes During Execution of Software Engineering Project Plan

Software engineering project management is specific project management problem. Lack of the historic data influence on risk while building project plan. Project plan consists of activities and tasks which had to be performed to achieve project goal(s). The author of the project plan can make a complete work breakdown structure needed, but estimation of the effort to complete tasks is still high risk area. In the literature articles can be found explaining fall of the quality projects. Reasons found seem big enough only at the first sight. These fictitious reasons inspired us to uncover new comprehension of processes during project execution. The question appears: is the project in execution phase stable or not? If it is stable then with the perfect plan there are no disturbances. Implication: No profound control needed. On the other hand it may be unstable and control loop is needed to ensure stability. Many authors use statistics to identify processes that significantly interrelate with stability of the project in execution phase and all of them omit interrelations among processes. At this point of time we present new theory and claim that processes are not stable and worst of all they are even hyper-sensitive. Hyper-sensation means that a small change of one process can cause a huge change of one or many of other processes. Since this is not necessarily true each time that a change occurs, statistics fail to foresee this kind of relations. The benefits of this new theory described in the article are that the project manager could foresee mutual disturbances that may occur while something goes wrong.

Key words: understanding processes, software engineering, project plan.

1. Introduction

Computer science is the fastest developing area of science ever known. New technologies emerge on daily bases. Methodologies for software development are changing. Use of software tools and code generators are wide spread among enterprises dealing with software development. The only anchor remaining in this

area is project management. Since 1984 when ANSI/IEEE Standard 730-1984 specifies Software Quality Assurance Plan that still holds, there has been no revolutionary improvement in software engineering project management activities. Planning means estimation of working activities needed to fulfill project goal(s). If we take a look at some standards, we can surely claim that planning is becoming an engineering discipline. On the other hand, articles can be found describing the dawn of brilliant planning and organizing; and the sunset of the same. Problem lies in understanding processes during project execution. *We claim that processes during project execution are interrelated and even more they are hyper-sensitive.* That theory can explain why some of project carefully planned and controlled went down the drain. The aim of this article is to open a new dimension for understanding disturbances during execution of software engineering project plan.

2. Present situation

In the literature many aspects of software engineering project management can be found. It is not our intention to describe things we already know. More important are things that we can not find, or even if we can find them they are not clear enough. One interesting aspect of the software engineering project management is area of motivation. B.W. Boehms has written an astonishing article [Boehm-89] stating that most important object in information system development is individual and it can not be treated like a machine. He also said that expectation of personnel is highest motivation activator. His revised golden rule says "Do unto others as you would have others do unto you - if you were like them". This rule implies understanding how people want to win and then match people's tasks to their win condition. This article does not say anything about implication into opposite side to make equivalence.

Our research goal was to find stability factors during execution of software engineering project management plan. In Walston-Felix [Walston-1977] study of productivity there are 29 factors that affect development process of information systems. None of them were related to human motivation. Boehm's Cocomo [Boehm-81] model has 15 factors grouped into 4 groups. Five factors are related to personnel attributes, but only to their experiences and capability.

P.J.Shank [Shank-93] was the first to find that motivation is important. Since his research was made for US Aerospace Defense Organization where personnel are carefully selected before they become hired, his research was much more narrow than for an ordinary project organization.

All factors in different studies were selected from large list of potential factors and with help of statistics (correlation) group of factors with high correlation were unified into one factor. That's how statistics work.

Another problem is historic data. The computer science is a very fast developing branch of science. That means many of historic data are out of date (for example, different approach in software development life cycle and programming techniques). Question not without relevance is whether we can adopt data from another organization or not and level of risk involving their adoption.

3. Presentation of new theory (hyper-sensitive project management)

We found many areas in software engineering project management specified sufficiently enough to make a good project plan. The starting point of our research was the question weather it is possible to make a perfect project plan. We tried to deny previous statement. Assuming that perfect project plan exists we searched the factors that would crash the plan. We put this problem on many project organizations to see if project organization could be stability factor. The most interesting result was that inter-personal relation, and motivation are the highest risk to the stability of the project. The explanation is simple. Individual is the crucial factor in software development. Mental activities during execution of software engineering project plan can not be planned (at the present state). It becomes obvious that psychology and sociology would become necessary in software development processes.

Model of processes during execution of the software engineering project plan was made. In the model we put all objects (project resources) that are involved in software development. Scheme of our model is presented in figure 1. We named project resources as objects because we add attributes, and state transition function that changes states of these attributes to project resources.

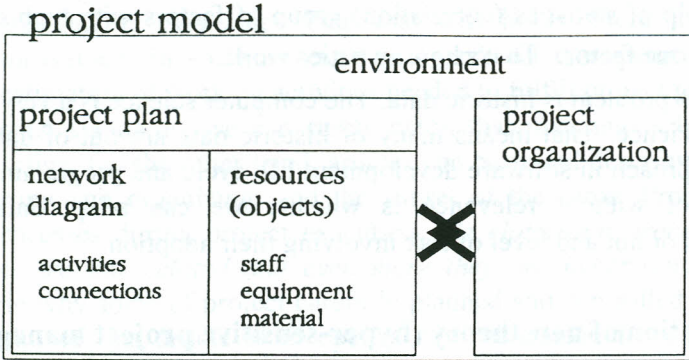


Figure 1: Scheme of project model

Attributes have various states depending on type of attribute. Since states are not constant during development of software we introduce state transition function. State transition function depends on other objects and environment. From the presented scheme we took universal object. Then we add environmental influences, and inter objects connections to it. New presentation is shown on figure 2.

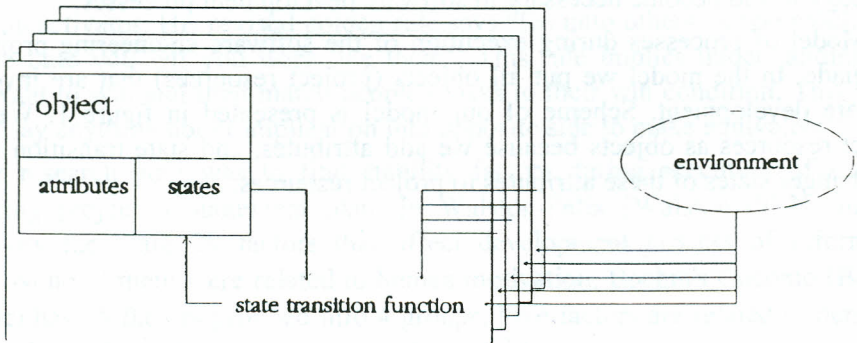


Figure 2: Inter-object connections

Previously mentioned universal object is basic component of our model. Model is further composed in the following manner: we take a project plan (supposedly

perfect) and attach specific object to the tasks of the project plan. Project organization has impact on this model in structure of inter-object connections.

In the model we have information about all objects involved in project execution. Now we must define processes during the execution of software engineering project plan. Processes are multilevel. The highest level consists of a single process and this is execution of project plan. Each activity (task) in the software engineering project plan is process while it is in the state of execution. Presented task-processes can be further decomposed to lower level processes depending on project resources assigned to specific task. Each resource presents different process while specific task is in the state of execution. For concluding the definition of processes during execution of software engineering project plan we can present a hierarchy of processes from top level process to bottom level process.

top level:	<i>software engineering project plan process</i>
middle level:	<i>activity and task process</i>
bottom level:	<i>project resource process</i>

Someone can oppose our decomposition to only three levels of processes since project plan itself can be decomposed into more levels. This is true but for our purposes this decomposition is sufficient. Middle level of any project plan decomposition is to increase visibility across structure of project.

At this point we have project model and processes attached to project model. The only things remaining unclear are inter process connections.

There are two types of inter process connections:

1. trivial connections and
2. nontrivial connections.

Trivial connections are connections that come from the hierarchical structure of processes. These connections make hierarchical tree. Nontrivial connections are those that transform hierarchical tree to network. Nontrivial connections depend on type of project and project organization. Now we are in the position to explore presented model of processes that had to be performed to achieve software engineering project goal(s). Exploration of model means use of model to perform simulation that denotes how this model acts in time.

4. Conclusion

Theory presented in this article has potential to help project managers in better understanding what is happening during execution of software engineering project plan. With simulation, where we can vary states of objects and state transition functions attached to these objects it is possible to foresee dynamics of processes that have to be performed. It is possible to identify stability factors and potential dangerous situations that may cause different losses. We can change objects in that model to present any specific organization and project plan. Simulation can predict response of whole system to any interaction with the model. It is possible to foresee how corrective actions taken to guide misguided project influence the dynamics of processes. In the end we must state that this model is capable to highlight hypersensitization of processes during execution of software engineering project plan.

References

- [1] [Boehm-81] Boehm, B.W. Software Engineering Economics Englewood Cliffs, NJ. Prentice Hall, 1981
- [2] [Boehm-89] Boehm, B.W. Theory-W. Software Project Management: Principles and Examples, IEEE Transaction on Software Engineering, Vol. 15, No. 7, July 1989, pp.902-916
- [3] Conte S.D., Unsmore H.E., Shen V.Y. Software Engineering Metrics and Models, Benjamin/Cummings Publication Company, 1986.
- [4] Frame J.D. the New Project Management: Tools for an Age of Rapid Change, Corporate Reengineering, Other Business Realities, Jossey-Bass Publishers, San Francisco 1994.
- [5] IEEE Standards Collection, Software Engineering, 1994 Edition
- [6] [Shank-93] SHANK, P.J. Modeling as a Tool in Defense Aerospace Software Project Management, Claremont Graduate school, 1993.
- [7] Reifer D.J. Software Management, Fourth Edition, IEEE Computer Society Press, Los Alamitos, California, 1993.
- [8] [Walston-1977] Walston, C. E., and C. P. Felix. A Method of programming measurement and estimation. IBM System Journal 16, 1 (1977): 54-73.

Received: 1995-09-06

Krašna M., Rozman I. Razumijevanje procesa tijekom provođenja projektnog plana programskog inženjerstva

Sažetak

Rukovođenje projektima programskog inženjerstva je poseban problem vođenja projekata. Plan projekta sastoji se od aktivnosti i zadataka koji se moraju izvršiti za postizanje ciljeva projekta. Jedno od pitanja je: je li projekt stabilan u fazi izvršenja ili nije? Ovdje prikazujemo novu teoriju i tvrdimo da procesi nisu stabilni, a najgore od svega je to što su svi oni čak hipersenzitivni. Hipersenzitivnost znači da mala promjena jednog procesa može prouzročiti veliku promjenu jednog ili više drugih procesa. Koristi od ove nove teorije opisane u članku su da voditelj projekta može predvidjeti međusobne utjecaje koji se mogu pojaviti.