



A Study on Project Management in Open Source Software Projects

Dr. Arul Kumar. N

Department of Information Technology,
IBRA College of Technology,
Sultanate of Oman

Indrani. P

Department of Computing,
Middle East College,
Sultanate of Oman

Abstract: *Project management (PM) in Open Source Software's is a systematic thought process. Based on a volunteer development approach, this paper develops a conclusion that supports the suggestion that the two disciplines are, in fact, linked and continued to be tightly entwined namely development process and release process. Based on this study, this paper proposes that a new Systems for Software Project Management (SfSPM) methodology is required for today's complex meta projects in Open Source software project management. The paper is intended to foster further research and discussion on the need for a project management paradigm shift for complex Open Source projects.*

Key words: *Open Source Software; Software Engineering; Open Source Software Projects; Project Management*

I. INTRODUCTION

Project Management (PM) is generally defined as delivering a specific project within an agreed time frame and within a fixed budget if applicable. To achieve this end to end process, the project typically follows a systematic life cycle process with different stages.

Software Engineering (SE) is also a systematic approach to solve a given problem in to multiple tasks. The purpose of Software engineering is to deliver a system that must interoperate harmoniously, which, in turn, requires a systematic and repeatable process for designing, developing, and implementing the system.

From a methodological perspective, PM and SE are systematic approaches to develop an Software. This commonality can infer that either PM has its origin in systems engineering, vice versa, or some other relationship may be present. This paper explores this relationship to highlight the interconnectivity of PM and SE.

The paper is organized to present a short historical look at PM and Systems thinking. This is followed by a discussion on how the two processes are intertwined. The concluding section will identify the challenges associated with the systematic approach in the complex project processes and proposes the need for a systems of systems PM paradigm shift.

II. PROJECT MANAGEMENT HISTORICAL PERSPECTIVE

Developing a unique activity with distinct task and development process traces itself back to antiquity. History clearly shows that grand cathedrals, impressive aqueducts and the pyramids were built by adopting some form of PM approach. Today, PM is playing a major role in space exploration and large military systems development.

In the last 30 years, PM, as a discipline, has grown from obscurity to a global and extensively written about discipline. PM can be rightfully called an emerging profession. In many ways, Project Management Workshop can be traced to the International Project Management Association (IPMA) and Project Management Institut & 1969 formations. From these early and slow starts, the mid 1980s resulted in general business acceptance and support and recognition how PM contributed to the bottom line. After that, the discipline quickly became a critical means for company survival.

Coincident with the 1980's general business mental mind paradigm shift, the discipline experienced the arrival of PMI's first PM code of ethics, the first PM Professional certification, and documentation of Project Management Body of Knowledge (PMBOK). The discipline continues to experience rapid global growth from industry and academia, as demonstrated by expanding PM research and subsequent PM literature.

One general literature theme is that PM deals with systems and its approach is a systems thinking method.

III. SOFTWARE SYSTEM THINKING HISTORICAL PERSPECTIVE

General software systems theory, software systems thinking, software systems analysis, software systems approach, and software systems engineering are all terms that have been and are used interchangeably, yet describe different functionalities and appears to have different origins. General software Systems Theory origins can be traced based on the review of literature.

Software System Analysis (SSA) origin has been reported to be an outreach of Operations Research (OR), where OR typically focuses on the software system mathematical solution. SSA broadens the analysis to include the human

aspect and soft system science components. SSA really owes its start within the US Air Force efforts to analyze the system impacts of adding intercontinental ballistic missile capabilities. SSA is also identified as a Management By Software Objectives (MBSO) process that analyzes and synthesizes algorithm of the management in a holistic manner. SSA and MBSO both require the development of clear measurable objectives from which alternatives can be developed, analyzed, comparisons and recommendations applied.

From several perspectives, SE is identified with technical specialists that are responsible for the development of hardware, software or a combination of technologies.

SE literature indicates that system engineers are given a predefined objective, typically without any context, and assigned to 'engineering the optimum solution. In today's environment, SE has evolved to include the broader contextual environment. As the International Council on System Engineering states, it integrates all the disciplines and specialty groups into a team effort forming a structured development process, that proceeds from concept to production to operation. SE considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs. It clearly brings out that SE involves more than just hardware, software or a combination of the two, with this grounded understanding of PM and SE, the next section will discuss the correlation between the two.

IV. PROJECT MANAGEMENT'S RELATION TO SOFTWARE SYSTEM THINKING

One approach to relating PM to SE is to look at the desired output and outcome of the disciplines. A literary review determines that both methodologies are intended to deliver a software system that meets the user requirements. In each case, the standards and processes typically involve a systematic, hierarchical management and development effort to achieve an assigned final release. From this approach, the two processes appear to be related and are, in fact, tightly intertwined and linked.

V. IMPLICATIONS FOR PROJECT MANAGEMENT OF COMPLEX SOFTWARE SYSTEMS

Software System complexity is growing due to increasing rate of technology change, broadening interconnection of software systems and human factors. All software systems, simple or complex, have similar characteristics as each consists of subsystems/elements interacting to produce a purposeful release management.

Research indicates that the greater the number of software subsystems and their interactions, the greater the software system complexity. The complexity increases as a result of increasing interacting and coupling of communication channels.

If the software subsystem interactions are loosely coupled, the software system is described as Detailed Complex, which can be decomposed at a level where the various interactions are well understood. Conversely, if the subsystem interactions are tightly coupled, the system is described as Dynamic Complex where the subsystem interactions cannot be determined through decomposition. Dynamic Complex software systems exhibit emergent behavior which challenges current PM and software systems thinking methodologies of delivering a predefined specific release.

These challenges, are as a result of the software systems, "Emergent structural and behavioral patterns suggest that the consequences and software system patterns cannot fully be known in advance of the software system deployment that we should design any complex software system with the understanding that we will experience structural and behavioral anomalies upon software deployment these must be expected and introduce a level of uncertainty into complex software systems.

PM and software engineering are also challenged by the software systems contextual setting. For complex software projects, the contextual setting is a critical element that must be considered throughout the software project life cycle. Current PM and software systems thinking processes are not designed to support these context rich boundaries.

VI. DISCUSSION

Software engineering faces the challenge of meeting the demands of ever increasing complex software systems. No longer can the software systems engineer focus on just the hard software system science, but must involve the software system science and complex interrelationships of interrelated software systems. Put in another perspective, the International Center of Software Engineering Perspective the global software engineering environment drives a new world view software Systems of Systems (SoS). To address this new world view, a new methodology has been proposed, Software Systems of Systems Engineering (S0SE). The expanding complex worldview requires an evolution of software systems thinking into SoS thinking. Yet, this new software system thinking worldview has many visions.

While the term SoS has no widely accepted definition, the notion is widespread and generally recognized. There is an emergent class of software systems that are built from software components which are large scale software systems in their own software systems which consist of independent, Collaborative components.

The concept of SoS has been extensively examined by various researchers and is broadly acknowledged as a challenging issue due to its high complexity. SoSE is fundamentally about structuring and ordering the analysis, design, or transformation of system solutions that rely on multiple complex systems to function as integrated volunteer development process.

As identified, there is no universal SoSE definition, yet there is an agreement that complexity is increasing and SE is being challenged to meet these demands. At present, it does not appear that a PM SoS world view has been developed. It is proposed that to address this shortfall, there is need for a paradigm shift in PM methodology, similar to SoSE.

Lack of awareness of the importance, value, timing, accountability and organizational structure of SE on programs SE has not adequately demonstrated the ability to operate beyond the technical level of enterprise problems. PM in Open Source Software successful delivery of projects continues to fail with the current process. Current project methodologies are inadequate for Open Source project implementations.

Adequate qualified resources are generally not available within the government and industry for allocation on major programs A new type of systems engineer, skilled in understanding and resolving complex systems problems must be developed and made accessible. (A new type of project manager, who is trained in SoSE PM skills and techniques is required).

Insufficient SE tools and environments to effectively execute SE on programs and implicit recognition of the environment, as well as the supporting tools, must be developed to address emerging SE problems. Current linear PM software tools are inadequate for complex interactive, iterative, emerging projects. New tools are required that can address emerging software system design, development and release concerns.

Software requirements definition, development and project management is not applied consistently and effectively. Requirements are a necessary function of effective SE. However, requirements must be developed from a holistic systems perspective. They must consider the wide array of organizational, managerial and contextual constraints influencing complex software system development not simply technological constraints and requirements. PM must address the emergent nature of complex Meta software system requirements.

As this study points outline, the volunteer Open Source Software Engineering community has seen a need to address the emerging software development concerns.

VII. SoSPM METHODOLOGY

In Open Source Software Development Process, System of Systems Project Management (SoSPM) methodology is required to manage complex software systems level projects and is presented as a guideline to establish a beginning for further discussion and research.

SoSPM is an iterative software development process of top down analysis to bottom up synthesis software development efforts. This iterative development process follows the SoSE meta software system analysis, software system transformation and dynamic equilibrium application of software cycle. The core change from current PM methodology to SoSPM is the continuous iterative required life cycle process. During the planning phase, there is a specific feedback loop that requires the project team to validate the Open Source developed problem statement and purpose as the project team moves into the meta software system synthesis phase among volunteers.

VIII. CONCLUSION

This paper presents the interaction analysis of Software Project Management and software system engineering relationship in Open Source Software Projects. This effort supports the concept that PM is tightly coupled and closely intertwined with SE. Each take up a systematic thought process and goal oriented problem resolution process. The analysis supports the conclusion that PM is an outgrowth of software systems science.

This study also discussed the emergence of complex meta software system PM. From this discussion, it is clear that increasing software project and systems complexity and greater use of meta software system concepts is strongly supported in current literature. Critical project and systems applications and high risk areas involve Open Source contained within complex contextual settings. Software System science, in response to the changing complexity landscape, is evolving towards a proposed SoSE methodology that is postulated as a process that can help minimize the complex meta software system risk.

While SoSE is evolving, the PM body of knowledge has to change to meet the meta software system challenges. To address this gap, a conceptual SoSPM methodology that relies on the new SoSE methodology foundation has been presented. This software process follows the early established basis that PM is an outgrowth and continues to be tightly coupled with system science. SoSPM conceptual methodology is intended to foster further discussions, research and debate on the need for a new PM process for complex meta software system Projects.

REFERENCES

- [1] Aberdour, M., 2007. Achieving quality in open-source software. *IEEE Software*, 24: 58-64. DOI: 10.1109/MS.2007.2
- [2] Coffin, J., 2006. Analysis of open source principles in diverse collaborative communities. *First Monday*.
- [3] Arul Kumar, N. (2012). Release process on quality improvement in open source software project management. *Journal of Computer Science*, August 2012, Vol.8(6), pp.1008-1011.
- [4] Arulkumar, N. & Chandra, S. (2012). The impact of release management and quality improvement in open source software project management. *India Applied Mathematical Sciences*, 6(62), 3051-3056
- [5] Crowston, K. and J. Howison, 2005. The social structure of free and open source software development. *First Monday*.
- [6] Erenkrantz, J.R., 2003. Release management within open source projects. *Proceedings of the 3rd Workshop on Open Source Software Engineering*, May 3-11, Portland, USA., pp: 51-55.

- [5] Feller, J., Fitzgerald, B., Understanding Open Source Software Development. Addison-Wesley, London, (2002).
- [6] Koch, S. and G. Schneider, 2002. Effort, co-operation and coordination in an open source software project: GNOME. Inform. Syst. J., 2: 27-42. DOI: 10.1046/j. 1365-2575.2002.00110.
- [7] Michlmayr, M., 2005. Quality improvement in voluntary free software projects: Exploring the impact of release management. Proceedings of the 1st International Conference on Open Source Systems, Jul. 11-15, Genova, Italy, pp: 309-310.
- [8] Michlmayr, M., F. Hunt and D. Probert, 2005. Quality practices and problems in free software projects. Proceedings of the 1st International Conference on Open Source Systems, Jul. 11-15, Genova, Italy, pp: 24-28.
- [9] Wynn, D.E., Organizational structure of open source projects: a life cycle approach. In: 7th Annual Conference of the SAIS (2003).