International Journal of Information Technology and Business Management
28th Feb 2015. Vol.34 No.1

ISSN 2304-0777 www.jitbm.com

STREAMLINING COLLABORATIVE SOFTWARE PROJECT DEVELOPMENT: THE CONCEPT OF THE MEETINGS-FLOW

Chung-Yang Chen, Ph.D.
Dept. of Information Management, School of Management
National Central University, Taiwan

Abstract

In today's integrated software and business project environment, managing stakeholders' involvement has been challenging. To help address this issue, this paper draws an attention on project meetings, and propose the meetings-flow (MF) concept for streamlining the stakeholder involvement. This paper serves as an introductory part of the entire MF research in software development and collaborative project development. In this paper, we primarily focuses on reviewing the role of meetings in software development, and presenting a software company's case to demonstrate the usage of MF, in particular for modeling the critical group proceeding of the collaborative software development.

Keywords: software process modeling, collaboration, meetings-flow

The Human Side of Software Development

Contemporary software development (SD) heavily requires the participation of various stakeholders and parties in accomplishing ad-hoc project tasks. Project activities such as feasibility analysis, presentation rehearsal, requirements exploration, critical artifact review and acceptance, project monitoring and control, change control, conflict resolution, etc. are performed in the form of group discussion and social presence to announce, brainstorm, negotiate, reach consensus, leverage peer pressure, and present or report works under public scrutiny. Stakeholders' involvement in software development, though not technical, contributes to the success of a project.

The importance of stakeholder involvement has been evidenced by many studies, e.g. (Faraj and Sambamurthy, 2006; Natale and Ricci, 2006; Marchewka, 2010; Standish Group, 2007). Unfortunately, managing people is not easy, and it gets more difficult in software development that particularly requires teamwork among the stakeholders (Crocitto and Youssef, 2003; Dennis and Garfieldll 2003; Faraj and Sambamurthy, 2006; Hong et al., 2004; Natale and Ricci, 2006; Probert, 1997).

According to Standish Group's chronicle reports, the stakeholder involvement problem continues to majorly cause software projects to fail (Marchewka, 2010; The Standish Group, 2007). These human-side issues of stakeholder involvement in collaborative software development should be emphasized and further integrated into mainstream methods and tools (FinstAM, 2003).

In a collaborative project that develops integrated products and processes, since it involves complex people configuration and participation, the stakeholder participation become *the* critical path (Chen, 2011; Roberts et al., 2002). From the process aspect, known project management and software process standards, such as the Project Integrated Management in the PMBOK and the Integrated Project Management process area in CMMI, suggest this kind of *people* issue be handled by proper planning and institutionalizing various group involvements throughout the development of a project (SEI, 2010; PMI, 2008). Hence, software projects ought to have a communicative venue for people to effectively distribute information and collaborate, and should sustain the communication venue throughout the development. This “people” refers to project stakeholders of the project team, the software organization, the suppliers, and the project customers and users.

Managing the Human-centered SD: a Focus on Meetings

Meetings are conceivable in serving as this communicative venue. In the integrated and cooperative software development environment, stakeholders’ participation and communication are usually done in the form of meetings (Verner and Evanco, 2005; Gallivan and Keil, 2003; Teasley et al., 2002; Rising and Janoff, 2000; Davision, 1999). According to many reports, meetings enable and facilitate participation in sharing inspiration, leveraging expertise and consolidating information (Hass, 2006; Newell, 2004; Gorse and Emmitt, 2007; Hass, 2006; Wenger *et al.*, 2002). Meetings are also helpful in codifying and preserving substantial group or team actions (Orlikowski and Yates, 1994). The codification (i.e. meeting minutes and subsequent supporting information and documents) becomes the group memories that support the collaborative development of a project.

The study of meetings has been a major topic in project management or information technology related literature. Much literature focuses on the subject of joint application development, group support systems (GSS/GDSS). They are mostly administrative and internal behavioral studies of effectively operating group action inside a meeting, or building computerized tools for running a meeting. In other words, although many studies and tools promote the contribution of meetings to ad-hoc group actions; these isolated meetings are at the micro level in meeting management. A more holistic and collective perspective with regard to the interconnectedness of previous, current, and future meetings may be further needed in order for forming the "group flow (Csíkszentmihályi and Csíkszentmihályi, 1992; Martin, 2010)" that streamlines the collaborative development. In this regard, some researchers take a macro approach with an innovative treatment on project meetings for managing stakeholder involvement. Such an approach suggests treating meetings as mutual interdependent entities, and interconnecting them to form *meetings flow*, a macro group process that represents the collaborative proceeding of software development.

The Meetings-flow Approach

The meetings-flow (abbreviated as MF) study is an emerging research. The concept was seen in conducting students' software capstone projects and engineering projects (Chen and Teng, 2011; Chen and Chong, 2011; Chen, 2009). According to these studies, the serialized manner of meetings forms the temporal meetings-flows, indicating how the collaborative development proceeds; the interdependent information sets among the meeting entities form the *contextual* flows, indicating the evolving of group knowledge.

In addition, Chen (2011) reported the usage of the MF in managing a multi-party large-scale engineering project. In the report, the project client used the MF to monitor and participate in the development. The MF synergizes the interconnectivity of project functional meetings and institutionalizes a continuous method and a more natural way of intervention. Such a meeting-oriented process was also recognized in the study as a new type of project's critical path—a path of showing the *people and communication* bottleneck of the project. This communication critical path differs from the traditional CPM (Critical Path Methodology) that solely focuses on the technical work path. According to Chen's argument (p.12) in the report, while a technical job may be easily done, it often requires substantial efforts to brainstorm the ideas prior to the work, taking considerable time for reviewing the work, and spending much effort on convincing the clients to accept the work.

Based on the aforementioned review on project meetings and current development of the MFA, we explore the applicability of the approach in software project development. Specifically in this paper we preliminarily introduce how the approach is used in a software project and the benefits of the approach may contribute in streamlining the collaborative development of the project.

A Software Company Case

Introduction

Founded in 1985, Environmental Science and Engineering Inc. (abbreviated as ESNE hereafter) is a system integration company in New Taipei City, Taiwan. ESNE develops ad-hoc meteorological software systems for its clients such as Central Weather Bureau, Taiwan Air Force, and other government agencies. In ESNE, the development of meteorological system projects requires various stakeholders or parties to participate in continuous and intensive validation and verification. Therefore, starting in 2008, the company conducted a research project (Liu, 2009) of using the meetings-flow approach for managing and streamlining project's critical collaboration and group communication path.

ESNE's projects are managed in waterfall-like phases: presale (i.e. preparation), development, and transition and maintenance. Therefore, this paper summarizes the company's MF framework as Figure 1 below. For simplifying the presentation due to the page limitation, in the following MF report we use only a phase (the "presale" phase) of a project example.

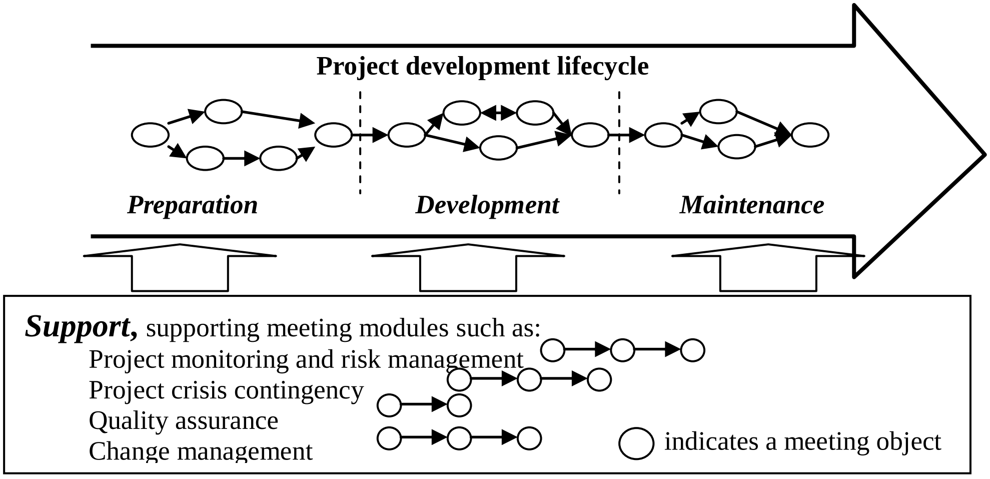
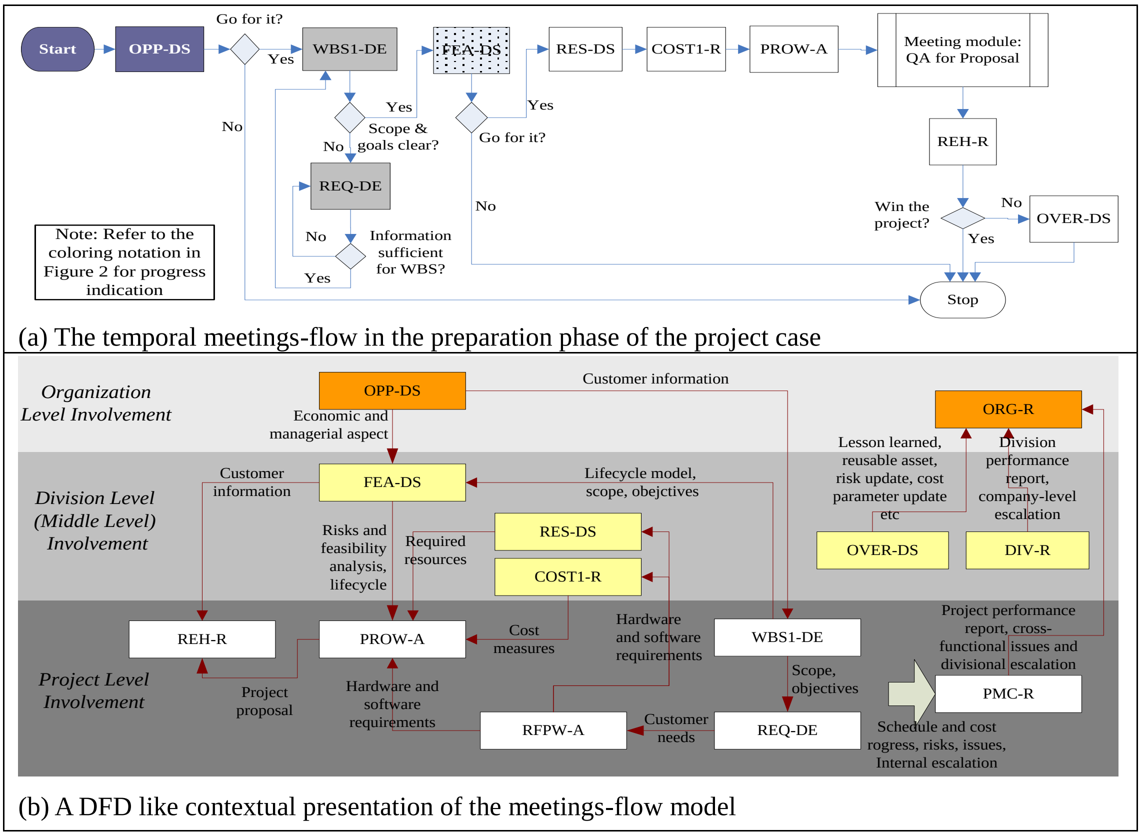

Project development lifecycle
Preparation
Development
Maintenance
Support, supporting meeting modules such as:
Project monitoring and risk management
Project crisis contingency
Quality assurance
Change management
indicates a meeting object

Figure 1: Managing software team processes by managing meetings and their flows

In TKE, meeting entities are identified from any work item on a project's WBS that requires group participation. TKE further defines the participation as group behaviors of brainstorming, review, announcement, reporting, presentation, and negotiation. The meetings are then characterized into various meeting types (classes) with the generic agenda, attending roles, participating roles, degrees of participation, etc. Once identified, the meeting classes are further linked up, according to their corresponding positions on the WBS. The upper part in Figure 2 below demonstrates such a temporal flow.


Start
OPP-DS
Go for it?
Yes
No
WBS1-DE
FEA-DS
RES-DS
COST1-R
PROW-A
Meeting module: QA for Proposal
Scope & goals clear?
Yes
No
REQ-DE
Information sufficient for WBS?
Yes
No
Go for it?
Yes
No
REH-R
Win the project?
No
Yes
OVER-DS
Stop
Note: Refer to the coloring notation in Figure 2 for progress indication
Organization Level Involvement
Division Level (Middle Level) Involvement
Project Level Involvement
OPP-DS
Customer information
Economic and managerial aspect
ORG-R
Lesson learned, reusable asset, risk update, cost parameter update etc
Division performance report, company-level escalation
Customer information
FEA-DS
Lifecycle model, scope, obejctives
Risks and feasibility analysis, lifecycle
Required resources
RES-DS
COST1-R
OVER-DS
DIV-R
REH-R
PROW-A
Cost measures
Hardware and software requirements
WBS1-DE
Project performance report, cross-functional issues and divisional escalation
Project proposal
Hardware and software requirements
Scope, objectives
PMC-R
RFPW-A
Customer needs
REQ-DE
Schedule and cost rogress, risks, issues, Internal escalation

(a) The temporal meetings-flow in the preparation phase of the project case

(b) A DFD like contextual presentation of the meetings-flow model

Figure 1: The meeting-oriented group processes in the case company

As the lower part of Figure 2 shows, the project uses DSM (design structure matrix) and a simple DFD-like diagramming tool to frame the information context for the planned meeting entities. The underlying multi-layered communication channel, visualizes the relationship and linkage between meetings (thus participants too) and levels of management. Such meeting-oriented collaborative proceeding, in both the temporal and contextual representations, is regarded as a macro-level group process of the project's development, and the generic content of the meeting types in the flow becomes a reusable group process model for similar projects to follow.

Discussion

Streamlining the collaborative software development: Due to a functional organization structure, employees of ESNE work on technical tasks of individual domains. Previously they argued about the lack of a whole picture regarding the shared vision of the project. The meetings-flow was found to fit in this gap. In a follow-up interview, participants replied that the project's meetings-flow enabled a shared track for people to join in together, streamlining the collaborative development by bringing the right information to the right people at the right time and venue.

One practical issue pertinent to such

a collective planning manner was raised in the case company. Participants recalled that in the beginning, the planning of meetings-flow was challenging, because they tended to regard meetings as unpredictable and nondeterministic events, e.g. how can the MF determine all the unexpected meetings and handle the change of previously concluded agenda in a meeting? But later they realized that the MF was not to cover all the meetings that happen or pop up during the development of a project. In TKE, meetings were defined based on the project's WBS, highlighting a collaborative path through the development. They are different from the communication events on demand.

Increasing meeting effectiveness: According to the company, project members felt that, by modeling collaboration into meetings, the MFA helped reduce the number of meetings overall. We further questioned whether project's communication was hindered due to the reduction of meetings. The responses were positive in two. The first benefit referred to the increased control inside a meeting. Because of the understanding of the contextual relationships (Figure (b)), participants in current meeting were able to track the information from previous meetings. Secondly, they became more careful in producing the meeting's outputs, which would be fed to other meetings.

Handling the dynamics of software development: a future study: As far as this paper presents, the MF approach was used to model the critical group path of the reported project. However, software development is inherently unique due to the characteristics e.g. different lifecycle modes, durations, participants, etc. The MF model in Figure 2 may not entirely fit into other projects. Moreover, due to the dynamic nature and the people factor, collaboration in software development may not go as planned. Although the members in the reported project expressed positive responses as mentioned above, they also concerned the effort devoted to follow the planned flow model as the project deployed. In this regard, the planning of project's meetings-flow should also consider the flexibility of dynamically adjust the flow to meet a project's specific dynamic needs. This echoes to the software process tailoring needs recommended in CMMI (SEI, 2010). This becomes one of our studies in the future development of the MF approach.

Conclusion

This paper is preliminary and conceptual in nature; it introduces an innovative treatment on project meetings and describes the concept of the MF approach to address the stakeholder involvement issue in software development. While the existing PM methods and software tools mostly present a discrete way to manage project development process, here suggests a new concept, a focus on meetings and meeting flows, to manage and streamline the stakeholder involvement in collaborative software development. In the future, the MF would be continually introduced to the society. This would include the methodological development of applying the MF in software development. Specifically, we would focus on how to align and tailor the flow model to meet individual project's needs.

Acknowledgement

We thank the Taiwan National Science Council for financially supporting this research (NSC-94-2213-E-182-004). We thank Environmental Science & Engineering Corporation for providing the needed operating environments of this research.

References

1. Bailetti, A.J., Callahan, J.R., and DiPietro, P. (1994) A coordination structure approach to the management of projects, *IEEE Transactions on Engineering Management*, 41(4), pp.394-403.
2. Crocitto, M. and Youssef, M. The human side of organizational agility, *Industrial Management & Data Systems 103, 6* (2003), 388-397.
3. Ceschi, M., Sillitti, A., Succi, G., and DePanfilis, S. (2005 May/June) Project management in plan-based and agile companies, *IEEE Software*, 22(3), pp21-27.
4. Chen, C.Y. and Chong, P. (2011) Software Engineering Education: A Study on Conducting Collaborative Senior Project Development, *Journal of Systems and Software*, 84(3), pp.479-491.
5. Chen, C.Y. and Teng C.K. (2011) The Design and Development of a Computerized Tool Support for Conducting Senior Projects in Software Engineering Education, *Computers & Education*, 56(3), pp.802-817.
6. Chen, C.Y. (2011). Managing projects from a client perspective: the concept of the meetings-flow approach, *International Journal of Project Management*, 29(6), pp.671-686.
7. Csíkszentmihályi, M. and Csíkszentmihályi, I.S. (Eds.) (1992) *Optimal experience: Psychological studies of flow in consciousness*, The University of Cambridge Press.
8. Damian, D., Izquierdo, L., Singer, J. and Kwan, I. (2007) Awareness in the wild: why communication breakdowns occur, *The 2007 International Conference on Global Software Engineering (ICGSE).*
9. Dennis, A.R. and Garfield, M.J. (2003) The adoption and use of GSS in project teams, toward more participative process and outcomes, *MIS Quarterly*, 27(2), pp.289-323.
10. Faraj, S. and Sambamurthy, V. (2006) Leadership of information systems development projects, *IEEE Transactions on Engineering Management*, 53(2), pp.238-249.
11. FinstAM, T.D. (2003) Project management, an integrated approach, *The British Journal of Administrative Management*, 34(January), pp.24-25.
12. Gallivan, M., and Keil, M. (2003) The user-developer communication process: a critical case study, *Information System Journal*, 13(1): pp.37-68.
13. Hass, M.R. (2006) Knowledge gathering, team capabilities, and project performance in challenging work environment, *Management Science*, 52(8), pp.1170-1184.
14. Heller, T. (2000) If only we'd known sooner: developing knowledge of

organizational changes earlier in the product development process, *IEEE Transactions on Engineering Management*, 47(3), pp.335-344.

15. Hong, P., Nahm, A.Y. and Doll, W.J. (2004) The role of project target clarity in an uncertain project environment, *International Journal of Operations & Production Management*, 24(12), pp.1269-1291.
16. Hudson, V.F. (2007) *The human side*, Industrial Engineer, 39(9) pp.40-44.
17. Humphrey, W.S. (2007) *Managing the Software Process*, Addison-Wesley.
18. Liu, H. (2009) *A study of the meeting-flow through the software project development in ESNE*. Master Thesis, National Central University, Taiwan
19. Marchewka, J.T. (2010). *Information Technology Project Management*, John Wiley and Sons Inc., New Jersey, USA.
20. Martin, P. (2010) *Better Business*, Pearson Publishing Inc., Upper Saddle River: NJ, USA.
21. Natale, S. and Ricci, F (2006) Critical thinking in organizations, *Team Performance Management*, 12(8), pp.272-277.
22. Probert, G. (1997) Projects, people, and practices, *Engineering Management Journal, June 1997* pp141-146.
23. Project Management Institute (PMI) (2008) *Project management body of knowledge (PMBOK)*
24. Rising, L. and Janoff, N.S. (2000) The scrum software development process for small teams, *IEEE Software*.
25. Roberts, T.L., Cheney, P.H. and Sweeney, P.D. (2002) Project Characteristics and group communication: an investigation, *IEEE Transactions on Professional Communication*, 45(2), pp.84-98.
26. Software Engineering Institute (SEI) (2010) *Capability Maturity Model® Integration (CMMI^{SM}) Version 1.3 for DEV*, CMU Press, Pittsburgh: PA.
27. Steele-Johnson, D. (2000) Goal orientation and task demand effects on motivation, affect, and performance, *The Journal of Applied Psychology*, 85(5), pp724-7238
28. The Standish Group (2007). *CHAOS Summary for 2006*, West Yarmouth, MA, USA, http://www.standishgroup.com/press/article.php?id=2
29. Teasley, S.D.; Covi, L.A.; Krishnan, M.S.; Olson, J.S (2002) Rapid software development through team collocation, *IEEE Transactions on Software Engineering*, 28(7), pp671 – 683.
30. Verner, J.M., and Evanco, W.M (2005 Jan/Feb**)** In-house software development: what project management practices lead to success? *IEEE Software*, 22(1), pp86-93.