

Collaborating Over Project Schedules

Suzanne Soroczak^{1,2} and David W. McDonald²

¹Information Systems and Technology Group, Intel Corporation, Hillsboro, OR 97124

²The Information School, University of Washington, Seattle, WA 98195

suzanne.m.soroczak@intel.com, dwmc@u.washington.edu

ABSTRACT

Numerous case studies and ethnographies have shown project management in software engineering to be a collaborative activity. However, project management "tools of the trade" do not readily support collaboration. As a result, project management breakdowns can occur. This paper discusses the issues of collaborative project management and makes recommendations for future project management tool development.

General Terms

Management, Documentation, Standardization

Keywords

Software Engineering, Project Management, Collaboration, Project Teams, PERT Chart, Gantt Chart, Work Breakdown Structure (WBS).

1. INTRODUCTION

Software development entails many activities including requirements gathering, design, coding, documenting, testing, and debugging. One activity that weaves together the entire process is project management. Many different project management methodologies have been developed, tried, and documented. Yet the typical tools of project management have remained largely unchanged. In this paper we examine project management practices in software development teams as reflected in the prior literature. This paper reports on a survey and analysis of case studies and ethnographies of software development teams from a project management perspective.

In the next section we provide a brief introduction to current project management tools and practices. Next we discuss a number of project management breakdowns identified in the literature on software development teams followed by the social aspects of managing and planning software development in a globally distributed environment. In section four, we outline issues with implications for the development of a new generation of project management tools. We close with ideas for future research in distributed collaborative project management.

2. PROJECT MANAGEMENT PRACTICE

The Project Management Institute (PMI) defines project management as "the art of directing and coordinating human and material resources throughout the life of the project by using

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Conference '04, Month 1–2, 2004, City, State, Country.

Copyright 2004 ACM 1-58113-000-0/00/0004...\$5.00.

modern management techniques to achieve predetermined objectives of scope, cost, time, quality, and participant satisfaction." [17] The PMI and the many practitioners of project management have built a book of knowledge (PMBOK) or "best practices", which aims to guide project managers in the art of managing projects. The best practices are meant to be a general guide to practice, applicable in many domains. In software development, project management methodologies are seen as one of the software development models used in conjunction with one or more management methods and techniques [18]. Developing a project task list and schedule is viewed as project management *practice* [16].

2.1 Project Management Tools Of The Trade

Project management tools are used for planning, scheduling, tracking, and controlling projects – the essential activities of managing a project. The most common project management tools are the project task list and the project schedule. A project task list is an enumeration of individual tasks and subtasks to be completed for the project. A project schedule is "a permanent record of a set of tasks to be executed, along with their predicted durations and completion times." [20] A schedule usually contains task attribute data and specifies who has responsibility for each task and information about the dependencies between tasks.

In Whittaker and Schwarz's paper on scheduling mediums, they identify the important functions that schedules serve. Schedules serve as a joint to-do list, allowing people to coordinate future action. They can be seen as a type of contract of the work promised to be executed. Schedules also "provide information without the overhead of interrupting other team members or calling a group meeting, serve as an external communication tool to people outside of the group, and can assist individuals in organizing their own work." [20]

Project teams make use of a variety of software tools for creating and maintaining project tasks lists and schedules. Fox and Spence's survey of project managers identified the top "tools of the trade" in use in software companies as MS Project, Project Workbench, and MS Excel [8]. Interestingly, project managers rely on nontraditional project management tools, such as MS Excel, as readily as more project-focused tools. This is especially true for projects of a short duration (less than six months), where the start-up costs associated with project-focused tools may dissuade people from their use. [8] As well, emergent technologies like SharePoint, Blogs, and Wikis are being adopted for *ad hoc* project organization and communication [11].

2.2 Creation of the Project Schedule

According to the PMBOK, a project schedule begins with the work breakdown structure (WBS). The WBS is an enumeration of project tasks. In software development methodologies, this task list is traditionally created using a top-down approach, in which tasks are broken into smaller more manageable subtasks. This

approach strongly appeals to management because it seems orderly, predictable, and facilitates the allocation of resources [6]. While the top-down approach seeks to decrease task complexity, it increases project complexity because at some point all of the subtasks have to be integrated. Additionally, tasks may be prioritized and scheduled based on a risk or user-driven attributes like feature requests in iterative development cycles.

Once a task list is created, the tasks are organized into a task timeline in the form of a Gantt or Program Evaluation and Review Technique (PERT) chart. The Gantt chart, named after H.L. Gantt [9], is widely used to represent projected schedules and actual task progress against time. PERT charts allow teams to manage the interdependencies between tasks.

In Agile software development, tasks are identified using a bottom-up approach. Individual tasks are established and then built up to more complex solutions. Teams are empowered to decide what they will work on and how they will do it. The Agile method, eXtreme Programming (XP), takes the managing activity to the extreme (no pun intended) by specifically banning the use of PERT charts [14]. The philosophy holds that PERT charts are built on the faulty assumption that the tasks of a project can actually be positively identified, ordered, and reliably estimated. Teams using Agile methods may employ other techniques such as whiteboards and stickies to manage task coordination and execution.

3. LITERATURE SURVEY

We surveyed a number of case studies and ethnographies of software development teams to identify project management activities. The development teams ranged in size from as few as 5 to more than 120 developers. The application domains included enterprise Internet solutions, telecommunications software, medical solutions, configuration management tools, government information systems, and commercial single-use software. Teams in these studies were practicing both formal and informal software development methods. The projects employed a variety of development methodologies including: Rapid Application Development (RAD), open source software (OSS) development, traditional or waterfall development processes, and several *ad hoc*, unspecified or unknown methodologies. Collaborative planning activities include brainstorming project task lists, identifying task attributes, such as estimating deadlines, recording task interdependencies, and assigning roles and responsibilities, and reporting task/project status.

3.1 Project Management Breakdowns

Despite the existence of "best practices" and standard project management tools, breakdowns in the activities of managing still occur. Project team members can have misunderstandings about the development process and can become isolated from each other and the rest of the organization, the coordination process may become bogged down so as to incur a schedule delay, and the scheduling tools may not adequately provide for the needs of the project team. The following subsections highlight how current project management tools contribute to these breakdowns.

3.2 Tools Obscure the Process

Because project schedules organize information by tasks, a project team using these tools is forced "to organize work by task and not by person"[20]. This can make it difficult to get a handle on what

other team members are doing and how one's own work fits into the whole process. Additionally, there is no way to link other management artifacts such as requirements documentation, test plans, or application programming interfaces (APIs) to the schedule. de Souza reports that because there was "no formal process to create and maintain APIs in the project plan" and therefore schedule, the APIs were forgotten until the last minute and caused additional schedule delays.[7]

Grinter similarly reports that Configuration Management tools did not create visibility into the development process. "At a higher level of abstraction, removed from the details of individual changes, the developers could not see how their work or other people's fitted together." [10]

Since project schedules contain a list of tasks and offer the ability to assign people to complete those tasks, it should be clear by looking at the schedule who does what. Yet misunderstandings about roles and responsibilities of project team members continue to crop up [4, 5]. This problem is exacerbated by the fact that team member roles and responsibilities can change over the course of the project [2]. Geographical distance and subtask complexity can also exacerbate the problem in large, distributed software development projects.

3.3 Schedule Can Isolate Team Members

In some cases the project manager can become isolated both from members of the project team and/or the rest of the organization. This can happen when the schedules and other work products created or maintained by project managers are incomplete or out-of-date. "Although most managers had developed progress tracking schemes, many were less aware of system status than were their system engineers." [4] How can it be that the person maintaining the schedule is so out of the loop?

Herbsleb found that reliance on documents can lead to impoverished and slow communication [13]. "There is a strong requirement for frequent updates of the schedule, so that it correctly reflects the current state of the project." [20] This can require daily maintenance without which the schedule can quickly become obsolete. Relying on these charts can cause project managers to overvalue the schedule to the process and consequently isolate themselves from the rest of the activities.

3.4 Tools Not Tied to Practice

Project management tools are not adequately tied to the practice of managing a project. Standard tools for maintaining project schedules are not linked to status reporting processes and require that the updates be funneled through a single person. In many cases, project teams reply on email status reports or 'today' messages to support group awareness [1, 12, 20]. This self-reported data may include as much or as little information as the developer wishes and as seen elsewhere, tight deadlines can encourage developers to be sparse with their comments [10]. When received by the project manager via email, the status report data must often be manually entered into the appropriate scheduling system to update the schedule. Consequently activities can feel like busy-work to the project manager and artifacts useless to the project team if the information is behind the times.

In some cases, status report and task data was conveyed via face-to-face meeting. Yet we have seen instances where "meeting data was not collected, so that information about decisions, rationales,

and responsibilities was lost." [19] Teams also had difficulties assimilating new members to on-going projects because there was no connection between streams of communication [12]. Meeting data that is collected may be entered into word processing documents or spreadsheets so that it can be emailed to the project team. Again, there may be no ties connecting this data back to the project schedule.

3.5 Unintelligible Project Schedules

Geographically separated development teams are more likely to have different development philosophies and make use of different terminology. For example, the role of the project manager in one group may differ from that in another part of the organization, causing confusion about responsibilities [5]. The terminology used in the task list and project schedule can create a problem of mutual unintelligibility. "For the schedule to be interpretable by all, it must use a shared vocabulary." [20]

"Large groups of 'project size', as Grudin (1994) calls them, can not find out what the status of the project is by social interaction alone." [4] As project team size grows, the number of tasks grows, and Gantt and PERT chart representations make it easy to get lost in the schedule. Large groups relying on the project schedule for status information may require matrix printers and entire walls to display such huge amounts of data. Such large schedules can make even simple navigation of the schedule problematic.

4. DISCUSSION

4.1 The Social Aspects of Managing a Project

In large-scale development teams, membership in teams and work on projects is not static. Team members are likely to be working on more than one project at a time. Additionally, project team structures can change to meet the needs of the organization. The assimilation of new project members becomes a project management activity. Project management tools can aid the assimilation of new team members by providing a contextualized knowledge record of the project.

The collaborative generation of project tasks and task attributes is a social activity and can produce a greater mutual understanding and expectations for the project. Collaborative scheduling empowers team members to determine task deadlines and may improve task deadline estimates. Additionally, making a schedule visible in a social context influences how the schedule is understood and interpreted by project members. Whittaker notes that publicly displayed project schedules support collaborative reflection on the project and task deliverables. [20] In comparing a publicly displayed wall schedule vs. an electronic version, "the board was considered more 'real' and 'credible' than traditional electronic schedules". [20]

4.2 Role of Documented Schedules

The use of formal documents impacts the flow of communication in organizations in many different ways. Charts and graphs can improve the vertical flow of information, because they abstract data to a level which informs management about the progress of activities. [22] Email and other informal written communication can improve the horizontal flow of information. [1, 12] Many development teams use email as a primary method of communication amongst developers quite successfully. Subtask groupings in schedules and APIs can serve to reify organization

boundaries, which may impede the flow of information [7]. Having a clear understanding of the role of project schedules in team communication is a necessity to future tool development.

In summary, project management tools, namely the project task list and project schedule have not changed much since their introduction. Because of the high-level of goal uncertainty in software development these tools may not be meeting the needs of project teams. Software tools for creating project schedules were initially developed as single-user software, and despite new collaborative features continue to be used as such. Unlike standard MS Office offerings, MS Project may not be installed on everyone's computer and so the outputs from the tool such as a Gantt chart are no more interactive than a paper printout. The source data may not be publicly available and therefore difficult to keep current. Scheduling tools give little guidance to support a task structure that is universally understandable. The current state of the project schedule can obfuscate the development process instead of making that process visible to the development team. Finally, these tools are designed to be used in a myriad of contexts so much so that they may be too generic for contextualized use.

5. FUTURE DIRECTIONS

Research contends that there is no one-size-fits-all methodology [3], so why are we using one-size-fits-all tools? Perhaps its time to move away from the PERT chart view of project management. Software development projects are known to be more uncertain than other types. A few methodologies have strived to overcome uncertainty issues with project management tools by avoiding them altogether. In XP, for instance, no formal project schedule is created. In these methodologies informal communication is expected to suffice for managing activities. However informal communication will not be sufficient in large-scale, distributed software development projects.

Based on the data drawn from current project management practices in software development teams, we make the following recommendations in the future direction of project management tools:

- Make the project management process more visible.
- Align tools with planning practices.
- Link tools to other project artifacts.
- Strive for a people-oriented, not task-oriented focus.
- Tool should be accessible by all to promote individual responsibility and collaborative planning.
- Tools should be publicly viewable to promote informal communication, collaborative reflection.
- Make the tools easy to keep update and maintain.
- Try other visualizations for identifying and enumerating tasks and task attributes.

6. CONCLUSION

In this paper we have discussed the social-aspects of project management in software development. Based on an analysis of a survey of the literature, we have made some recommendations for the future development of collaborative project management tools. We hope that future development may be inspired by other visualization techniques such as social network maps and activity rhythms.

7. ACKNOWLEDGMENTS

We would like to acknowledge support by Intel Corporation for this research. We thank our colleagues at the UW iSchool for their input and reflection.

8. REFERENCES

- [1] Bernheim-Brush, A. J., & Borning, A. (2003). 'Today' messages: lightweight group awareness via email. In CHI '03 extended abstracts on Human factors in computing systems (pp. 920-921). Ft. Lauderdale, Florida, USA: ACM Press.
- [2] Beynon-Davies, P., Mackay, H., & Tudhope, D. (2000). It's lots of bits of paper and ticks and post-it notes and things...': A Case Study of a Rapid Application Development Project. *Journal of Information Systems*, 10(3), 195-216.
- [3] Charvat, J. (2003). Project management methodologies: selecting, implementing, and supporting methodologies and processes for projects. Hoboken, NJ: John Wiley.
- [4] Curtis, B., Krasner, H., & Iscoe, N. (1988). A Field-Study of the Software-Design Process for Large Systems. *Communications of the ACM*, 31(11), 1268-1287.
- [5] Damian, D.E. & Zowghi D. (2003) RE Challenges in Multi-site Software Development Organizations. *Requirements Engineering* 8:149-160.
- [6] DeGrace, P. & Stahl, L.H. (1990) *Wicked Problems, Righteous Solutions*. Englewood Cliffs, NJ. Yourdon Press.
- [7] de Souza, C. R. B. d., Redmiles, D., Cheng, L.-T., Millen, D., & Patterson, J. (2004). Sometimes you need to see through walls: a field study of application programming interfaces In *Proceedings of the 2004 ACM conference on Computer supported cooperative work* (pp. 63-71). Chicago, Illinois, USA ACM Press.
- [8] Fox, T.L., & Spence, J.W. (1998) *Tools Of The Trade: a survey of project management tools*. *Project Management Journal* 29(3), 20-27.
- [9] Gantt, H.L. (1919) *Organizing For Work*. Harcourt, Brace, & Howe. New York, NY.
- [10] Grinter, R. E. (1995). Using a configuration management tool to coordinate software development. Paper presented at the *Proceedings of conference on Organizational computing systems*, Milpitas, California, United States.
- [11] Grudin, J. (2006) *Enterprise Knowledge Management and Emerging Technologies. Proceedings of the 39th Hawaii International Conference on System Sciences (HICSS-39)*. 10 pages
- [12] Gutwin, C., Penner, R., & Schneider, K. (2004). Group awareness in distributed software development. Paper presented at the *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, Chicago, Illinois, USA.
- [13] Herbsleb, J.D. & Mockus, A. (2003) An Empirical Study of Speed and Communication in Globally-Distributed Software Development. *IEEE Transactions on Software Engineering* 29(6), June 2003.
- [14] Larman, C. (2003) *Agile & Iterative Development: a manager's guide*. Boston, MA. Addison-Wesley.
- [15] Lin, J., Newman, M.W., Hong, J.I., Landay, J.A. (2000) DENIM: finding a tighter fit between tools and practice for web site design. In the *Proceedings of CHI 2000*. April 1-6, 2000. The Hague, Amsterdam.
- [16] Maylor, H. (2001) Beyond the Gantt Chart: project management moving on. *European Management Journal* 19(1), 92-100.
- [17] PMI. (2006). *Project Management Best Practices (PMBOK)*. Retrieved March 3, 2006, 2005, from http://www.pmi.org/prod/groups/public/documents/info/pp_pmbokguidethirdexcerpts.pdf
- [18] Sorensen, R. (1995, January 1995). A Comparison of Software Development Methodologies. *Crosstalk*, 1, from <http://www.stsc.hill.af.mil/crosstalk/1995/01/comparis.asp>
- [19] Walz, D. B., Elam, J. J., & Curtis, B. (1993). Inside a Software-Design Team - Knowledge Acquisition, Sharing, and Integration. *Communications of the ACM*, 36(10), 63-77.
- [20] Whittaker, S., & Schwarz, H. (1999). Meetings of the Board: The Impact of Scheduling Medium on Long Term Group Coordination in Software Development *Computer Supported Coop. Work* 8 (3), 175-205
- [21] Yates, J. (1993). Genres of Internal Communication. In *Control Through Communication: the rise of system in American management* (pp. 65-100). Baltimore: Johns Hopkins University Press.