# Monitoring GSD Projects via Shared Mental Models: A Suggested Approach

Matthew Bass
Siemens Corporate Research, Inc
755 College Road East
Princeton, NJ 08540
+1 609 734 3567
Matthew.Bass@Siemens.com

## ABSTRACT
Team cognition research suggests that the degree to which teams have developed shared mental models is a significant factor in the performance of the team. Research in the software development domain has similar findings. This research is not, however, reflected in most commonly used project management practices. In geographically distributed software (GSD) development difficulty with team coordination is the norm. This paper looks at these issues, the research into team mental models, and suggests how project management practices may incorporate the findings to help address coordination issues in GSD.

## Categories and Subject Descriptors
D.2.9 [**Software Engineering**]: Management - *Productivity and software teams*

## General Terms
Management, Measurement, Programming teams and Human Factors.

## Keywords
Global software development, shared mental models, project management.

## 1. INTRODUCTION
Team cognition research suggests that the degree to which teams have developed shared mental models is a significant factor in the performance of the team [1][2] [7]. Similar research in the software development domain has made similar findings [13] [16]. This research is not, however, reflected in most commonly used project management practices. In geographically distributed software (GSD) development difficulty with team coordination is the norm. This paper looks at these issues, the research into team mental models, and suggests how project management practices may incorporate the findings to help address coordination issues in GSD.


*GSD'06,* May 23, 2006, Shanghai, China.


It has been well documented that coordination problems abound in projects with geographically distributed teams [9] [9] [11] [12]. It is difficult to gauge if teams have adequately understood and are able to act on information. Additionally no "right" model for coordination has been found. Just because an approach has been successful in one project doesn't mean it will be sufficient in another project. As a result of these difficulties tasks often take longer, morale suffers, and quality problems are not uncommon.

Common management practices use a document or artifact driven approach. In other words managers track the production and delivery of artifacts against the project management plan. The identification of these artifacts is typically driven by the organizational processes (at least in more mature organizations). Ideally these processes are based on what has worked well in past projects. This approach seems to work well in organizations that have many projects with similar characteristics. This is often not the case, however, in GSD.

A trend that seems to be gathering momentum in Siemens is to adopt an agile approach. Such an approach has very short iterations (typically 4 – 6 weeks) by the end of which portions of the system are delivered. The requirements can change between iterations and progress is measured by the incremental delivery of systems of value. This approach relies on the very close interaction among developers and between developers and customers (or their proxies). It is more often in this case that you have an implicit design that all the developers are familiar with (as they work so closely together) with frequent re-factoring as needed. This approach has also been successful in Siemens with smaller teams that are collocated.

Agile approaches have issues when organizations begin to distribute the work. The nature of the developing organization changes drastically and practices that worked with groups of collocated like minded people may no longer work with teams that don't know each other, are from different cultures, may not be familiar with the domain, and can't communicate directly without difficulty. Likewise when you attempt to adapt an agile approach to distributed teams it is difficult to coordinate across teams. The groups that are consistently successful within Siemens are groups that have been developing systems in a

distributed manner for a long time; they typically have long-term relationships with outsourcing partners, and have learned through trial and error what works and what doesn't. These lessons came with a high price, however. There was often an initial reduction in productivity and an increase in costs before the optimal balance of overhead and risk was determined.

What is it about GSD that is inherently different than traditional collocated software development? The primary difference identified has been the lack of informal and ad-hoc communications. The impact of such a lack of communication has been shown to be great [8]. The strategies that have been employed to deal with these issues both in research and in practice seem to fall in one of two camps:

- Decouple the work. This usually comes in the form of decomposing the system into "loosely coupled" components or subsystems, or attempting to decouple various aspects of the lifecycle (e.g. requirements from design and build)
- Augment the lack of face-to-face communication with other means of coordinating. Various collaborative technologies have been used and introduced, different often more formal means of specifying artifacts have been suggested, and augmented processes have been developed.

The problem is that in practice project management has no means of monitoring the effectiveness of such practices. Looking at past projects in Siemens we have not been able to say with any certainty that a given practice works, or even that a given practice doesn't work. We have seen projects that seem to defy logic and are successful despite practices that were (to our way of thinking) incompatible with GSD. In other cases projects that seem to "be doing everything right" failed.

Is there than another option? If there was some way to measure the effectiveness of the currently employed coordination mechanisms would that give us an indication of whether progress is likely much earlier? Research both in the team cognition and software development domains suggest that development of shared mental models is an important factor for a successful team. Agile practices seem to have understood this and have included practices in the "agile manifesto" [17] that seem to be aimed at developing shared mental models within a team.

If a project manager were to be able to measure the extent to which multiple teams coordinating on a given task have been able to develop a common understanding of that task, would that not give an accurate indication of the effectiveness of the current practices? If the research mentioned above holds true then long before teams are able to produce a meaningful output (the current artifact being measured) they would need to develop a shared mental model around that task. If it was recognized that the mental model were not developed management would be in a position to take appropriate corrective action before production began rather than continuing with production (in order to meet the schedule) only to introduce further delays in the form of rework and bug fixing.

# 2. RESEARCH ON TEAM COORDINATION

There is a long history of research into the existence and importance of shared mental models in the cognitive physiology arena (literature in this area dates back to 1934 [15]) [1] [2] [3] [4] [5] [6] [7]. More recently work has been done exploring these ideas within software development [13] [16]. The results and recommendations have been fairly consistent. In this section we will look at the relevant aspects of both bodies of work.

## 2.1 Team Cognition Research

Work into the existence and importance of shared mental models dates back to 1934 [15]. More recent work has focused on understanding the team processes underpinning distributed and ad-hoc work [1] [2]. Much of the focus of this research has been civilian (police and emergency crews) and military teams who often need to operate in complex and dynamic environments. These teams need to cooperate in high stress environments with ill defined goals, limited ability to meet face to face, and have to deal with unstructured problems [1].

One of these key principles in this work, *situational awareness* refers to an individual's understanding of what is happening in their environment and ability to be able to predict what will happen in the future [5]. Research suggests that if people have a high degree of situational awareness then they are more likely to be able to engage in effective teamwork [6]. Pascual found that 83% of those surveyed found it more difficult to establish and maintain situational awareness in distributed teams than in collocated teams [2].

A closely related concept is that of a *shared mental model* as defined in [7]. Shared mental models refer to a common mental model held by a team that provides them "a set of organized expectations for performance from which accurate, timely predictions can be drawn" [7]. This allows teams to understand and predict the needs of others and take appropriate action even when overt communication isn't possible. A number of studies indicate that development of a shared mental model is an important aspect of team performance [3] [4] [5]. The more recent aim of this work in the command and control arena is to suggest additional team and leadership training as well as tools to promote means of developing and sustaining shared mental models [1] [2].

There have been a number of findings worth mentioning. It is certainly no surprise that communication is both integral and critical to effective teamwork [2]. It has also been established in this work that communication is much more difficult when the parties are geographically distributed. Misunderstandings were reported as "regular occurrences", team leaders were unsure whether their communications were properly understood without face to face communication, and cultural and linguistic barriers were reported as key communication challenges [2].

## 2.2 Shared Mental Models in Software Development

In addition to the work into team cognition in general, some studies have been done looking at coordination and the role of shared mental models specifically in software development. Robillard et al have studied cognitive activities in software development teams [19]. They have developed an approach for studying collaborative software development [18], observed teams during design review meetings, coded the interactions amongst team members and studied the results [20]. The conclusions reached by Robillard mirror those found in some of the studies mentioned above. Robillard has determined that shared mental models (he calls the process of developing a shared mental model *cognitive synchronization*) are an essential part of software engineering [16]. The approach that Robillard suggests is to introduce a step in the software engineering lifecycle called "synchronization meetings". This is a meeting where the explicit goal is to synchronize the mental models of the participants.

Espinosa et al. [13] also looked at the role of coordination and shared mental models in developing software using distributed teams. They found that work familiarity (the extent to which an individual is familiar with a task at hand) and the development of a shared mental model were important factors in the ability of teams to coordinate. Furthermore they found an inverse relationship between the coordination of teams and the development time of tasks (the more coordination the less time it took to complete the task). The results of these studies were consistent with the findings of the other studies mentioned above indicating that the team cognition plays a similar role in software development as it does in other areas such as command and control.

# 3. CURRENT APPROACHES IN PROJECT MANAGEMENT

Most current project management approaches do not explicitly recognize the role of "cognitive synchronization" in software development. When it comes to monitoring and controlling projects, managers typically use document or artifact driven approaches. That is they monitor the delivery of artifacts against a predefined schedule. These artifacts are typically things like requirements artifacts (e.g. a Software Requirements Specification) design artifacts (e.g. an Architecture Overview document), test plans, code deliverables, and so forth. Progress of the project is determined by the timely delivery of these artifacts.

The definition of the artifacts mentioned above is usually part of the organization's development processes (at least in more mature organizations). These artifacts ideally have been demonstrated to be useful in past projects and are assumed to be suitable for current projects as well. The characteristics of GSD projects, however, are often radically different then that of past projects (unless the organization has a long history of GSD projects of similar makeup). As described in numerous reports [8][9][13], these differences can have a drastic impact on the outcome of the project.

When GSD projects are managed using artifact driven approaches it is possible (in fact common) for projects to go off track and still be able to make a number of their deadlines. There is often a high level of management pressure to make deadlines and teams will do whatever they can to make interim deadlines. By the time it becomes obvious to managers that a project is not on track it is often quite late in the game and the effort and expense to rectify the issues could be too great.

Because of these issues mentioned above some projects within Siemens (and in some cases entire organizations) are turning toward agile approaches. Agile approaches are appealing because the managers get to see concrete results regularly [21]. In fact in many ways the agile practices are focused on optimizing the extent to which the team shares a common mental model [17]. This has been working well inside of a single team. There is no means, however, for synchronizing across teams. In fact the means that managers use in an artifact driven approach are often difficult to implement in a multi-team agile environment. Siemens has had some limited success in collocated multi-team agile projects, but agile project management for distributed teams is quite difficult.

# 4. MANAGING VIA SHARED MENTAL MODELS

As mentioned previously the established and emerging approaches are largely aimed at identifying *the* practice or set of practices that will work in projects. The problem is that most of these approaches work well for particular projects and not so well for others. We do not currently have an understanding of all of the factors that influence the extent to which a particular practice will work in a given context. Work such as [22] recognizes this and suggests that a dynamic approach be adopted that provides a means for adjusting the processes and practices mid stream. Managers still, however, need a mechanism for determining the health of their project at any given point in time. If managers were able to monitor the extent to which teams were able to develop and maintain a shared mental model around a given task, they would have a more direct measure of the effectiveness of the current practices than currently exists.

Looking at the research it appears to be clear that when teams work together to complete a given task they first need to come to a common understanding around what the task is, what their own role in the task is, and how the cooperating team may respond to a given situation. Development of this mental model is done well before completion of the task. There are several ways in which managers might measure either directly or indirectly the extent to which a shared mental model exists. One is to monitor coordination through social network analysis [23]. Through administration of a lightweight web-based survey (typically less than 10 minutes per individual) on a bi-weekly basis, the social networks of the project can be monitored. These networks indicate things like:

- Who in the project does an individual coordinate with and how often

- How aware is an individual in the activities, skills, and roles of others in the project
- To what extent, about what, and how are teams coordinating
- Who is involved with particular kinds of decisions in the project

If project managers have an idea about what the interdependence of tasks are, and how these tasks are planned, then they have an expectation about where and when coordination should occur. If coordination is not occurring as planed, then perhaps management should investigate to discover why not. Likewise if coordination suddenly spikes, perhaps it is worth investigating why the spike occurred. Is it the case that there is a recognized misunderstanding and efforts are being made to resolve these efforts?

Another option is to adopt a form of syncmeetings as suggested by Robillard [19] and use their protocol analysis techniques to determine the extent to which teams are on the same page. If during these meetings, there is a high degree of cognitive synchronization exchanges then it can be assumed that the shared mental model is not yet fully developed. One could imagine that if these meetings were held regularly then trends would emerge that would indicate progress in this direction. It could also be the case that sudden spikes could reflect a change or issue in some aspect of the task.

There may be (and likely are) other means by which managers could monitor this aspect of the project. This would take a fundamental shift in perspective on the part of managers in order to explicitly pay attention to these concerns. In addition a set of tools (some of which are quite simple or exist already) would be needed to ensure that monitoring of the synchronization activities was lightweight and non-disruptive to the project. If this practice became ingrained in a culture and a language and vocabulary around this topic emerged it would also provide a more structured way to determine the appropriate time for a given project to transition from one phase of development into another.

## 5. NEXT STEPS

There are several prerequisites to being able to put such an approach into practice. We would need to have a more concrete notion about what the tasks are around which teams need to cooperate. In addition we would need a better understanding of the specific nature of interdependence between aspects of the project. For example in what ways are various work allocation units of the architecture coupled to other work allocation units. What are the implications of these couplings from a coordination perspective?

We are currently investigating these and related issues. We have an experimental geographically distributed project that we use as a test bed and data source for some of these ideas and in addition we are in the process of defining some empirical studies with projects at Siemens.

## 6. REFERENCES


[1] Pascual, R. G, Henderson S. and Mills M. *Understanding and supporting team cognition* DERA CHS Year report. May 1998

[2] Pascual, R.G., Mills, M.C., Blendell, C. Supporting Distributed and Ad-Hoc Team Interaction People in Control: *An International Conference on Human Interfaces in Control Rooms, Cockpits and Command Centres:* 21 - 23 June 1999

[3] West, M. Effective teamwork. *The British Psychology Society.* 1994.

[4] McIntyre, I. M. and Salas, E., *Measuring and managing for team performance: Lessons from complex environments.* In GUZZOR, 1995

[5] Zsambok, C.E. and Klein, G. *Naturalistic decision making*. Lawrence Erlbaum Associates, Inc. 1997.

[6] Cook, M.J. and Elder, L. *Decision making planning and teams.* In Computer mediated complex supervisory and decision making in teams. IEE Computer and Control Division, Scottish Engineering Centre, Glasgow. 1997.

[7] Cannon-Bowers, J., Salas, E. and Converse, S.A. *Shared mental models in expert team decision making.* In Castellan, N.J. (ed) Current Issues in Individual and Group Decision Making, Lawrence Erlbaum Publishers. 1993.

[8] Herbsleb, J.D. & Mockus, A. An Empirical Study of Speed and Communication in Globally-Distributed Software Development. *IEEE Transactions on Software Engineering* 2003

[9] Herbsleb, J.D., Paulish, D.J. and Bass, M. Global Software Development at Siemens: Experience from Nine Projects. *Proceedings of the International Conference on Software Engineering (ICSE 05)* (St. Louis, Missouri, May 15-21, 2005)

[10] Herbsleb, J. D., and Grinter, R. E. .Architectures, Coordination, and Distance: Conway.s Law and Beyond,. *IEEE Software (16:2)*, September/October, 1999, pp. 63-70.

[11] Herbsleb, J. D., Mockus, A., Finholt, T., and Grinter, R. E. Distance, Dependencies and Delay in a Global Collaboration,. *In Proceedings of the Conference on Computer Supported Cooperative Work 2000*. New York: ACM Press, 2000.

[12] Carmel, E. *Global Software Teams*. Upper Saddle River, NJ: Prentice Hall, 1999.

[13] Espinosa, J. A., Kraut, R.E., Slaughter, S. A., Lerch, J. F., Herbsleb, J. D., Mockus, A. Shared Mental Models, familiarity, and coordination: A multi-method study of distributed software teams (2002). *International Conference in Information Systems,* Barcelona, Spain, December 15th – 18th, pp. 425-433.

[14] M.E.C. Hull et al., Software Development Processes — An Assessment, *Information and Software Technology,* vol. 44, no. 1, 2002, pp. 1–12.

*[15]* Mead, G.H. (1934) mind, self, and society. *Chicago. University of Chicago Press*

[16] Robillard, P.N., Opportunistic Problem Solving in Software Engineering. *IEEE Software* 2005

[17] Manifesto for Agile Software Development www.agilemanifesto.org

[18] D'ASTOUS P., ROBILLARD, P-N., DÉTIENNE, F., VISSER, W ., A coding Scheme to Analyze Activities in Technical Review Meetings, *PPIG98, Psychology of Programming Interest Group, 10th Annual Workshop,* January 5th-7th, 1998

[19] ROBILLARD, P-N., The Role of Knowledge in Software, *Communications of the ACM,* Vol. 42, No 1, pp 87-92, 1999

[20] D'ASTOUS, P., ROBILLARD, P-N., Characterizing Implicit Information during Peer Review Meeting, 22nd *International Conférence on Software engineering ICSE2000,* Limerick, Ireland, June 4-11, 2000, pp460-466

[21] Schwaber, K., Beedle, M., *Agile Software Development with Scrum*. Prentice Hall 2002

[22] Herbsleb, J.D., Paulish, D.J. and Bass, M. Coordination Profiles: Planning for Globally Distributed Projects. Submitted to: *Special Issues of IEEE Software on Global Software Development* 2006

[23] Browning, T.R. Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Direction. *IEEE Transactions on Engineering Management,* Vol. 48, No. 3, August 2001