Global Software Development: Who Does It?

Andrew Begel
Microsoft Research
Redmond, WA 98052, USA
andrew.begel@microsoft.com

Nachiappan Nagappan
Microsoft Research
Redmond, WA 98052, USA
nachin@microsoft.com

Abstract

In today's world, software development is increasingly spread across national and geographic boundaries. There is limited empirical evidence about the number and distribution of people in a large software company who have to deal with global software development (GSD). Is GSD restricted to a select few in a company? How many time zones do engineers have to deal with? Do managers have to deal with GSD more than individual engineers? What are the benefits and problems that engineers see with GSD? How have they tried to improve GSD coordination? These are interesting questions to be addressed in an empirical context. In this paper, we report on the results of a large-scale survey of software engineers at Microsoft Corporation. We found that a very high proportion of engineers are directly involved with GSD. In addition, more than 50% of the respondents regularly collaborate with people more than three time zones away. Engineers also report that communication difficulties around coordination are the most critical, yet difficult to solve issues with GSD.

1. Introduction

Global Software Development (GSD) is a field of research that has grown tremendously over the last decade [8, 9]. Herbsleb and Moitra [9] attribute the acceleration of GSD to (i) capitalization of the talent pool and resource usage wherever needed; (ii) business advantages of new markets; (iii) quick formation of virtual teams to capitalize market needs; (iv) improvement of time-to-market by utilizing "around-the-clock" development, and (v) flexibility to capitalize on merger and acquisition opportunities globally. More importantly, Herblsleb and Moitra outline the major dimensions of problems in GSD:

- **Strategic issues:** determination of projects that are disjoint architecturally, as much as possible.
- **Cultural issues:** understanding various cultures – norms and practices.
- **Inadequate communication:** due to difference in time zones and the lack of immediate response to questions.
- **Knowledge management:** Sharing product and domain knowledge between teams.
- **Project and process management issues:** synchronization between project and product management deadlines.
- **Technical issues:** due to bandwidth problems, problems in replicating code bases in different geographical locations.

With the advent and rapid expansion of GSD, it is surprising to note that there has been little empirical evidence on the extent to which engineers in a company encounter it. Does the entire company deal with GSD at some level, or is it only a select minority, perhaps just senior management? We assess these questions by conducting a large-scale survey to engineers working for Microsoft Corporation.

The organization of this paper is as follows. Section 2 discusses the related work and Section 3 our research method. Section 4 and 5 discuss the results and observations and Section 6 the threats to validity. Section 7 concludes our paper.

2. Related Work

Fred Brooks in the classic *Mythical Man-Month* [6] book states that in software systems: schedule disasters, functional misfits and system bugs arise from a lack of communication between different teams. This statement, though not in a GSD context, has gained even more significance with GSD's rapid spread. In this section, we present related work from companies that have used distributed GSD extensively.

Sengupta et al. [11] show the use of tools for distributed requirements management. Based on semi structured interviews of 30 engineers in the US, Netherlands and India, they identify potential areas for research impact on GSD, like development of collaborative environments, reverse-engineering, and maintenance of informal knowledge in a human-independent way. Battin et al. [4] describe their experiences with GSD at Motorola developing a 3G cellular system with 20% of the required staff in the US and the remaining 80% in Tokyo, Beijing, Singapore, Bangalore, and Adelaide. Bass et al. [3] report on collaboration experiences at Siemens and the problems learned with key learning's in people and communication-related aspects of collaboration. Agarwal et al. [1] discuss the role of full-time quality analysts at Tata Consultancy Services (TCS) and experiences from their pilot projects. Similarly, Narayanan et al. [10] compare two-offshore structuring models at Wipro Technologies with different organizational structures to enable manager consider various alternatives to select the best model suited for them.

We are unable to find exactly what proportion of the overall company has to deal with GSD from the reports of these studies. Thus, we conducted our own study to address this issue.

3. Research Method

We conducted an anonymous web-based survey over a period of two weeks in October 2007. An invitation was sent by email to 2,830 recipients, randomly selected from a much larger pool of around 28,000 software developers, test developers, and program managers. A 10% sample of each job role directly involved with software engineering was selected, independent of geographic location. Respondents were asked a total of 39 questions on general software development and process. More details on the survey can be found in an earlier report on this work [5]. Part of this survey was used in our research analysis to study the extent of GSD within Microsoft.

4. Results and Observations

Our survey received a total of 511 responses for a response rate of 18%. Our respondent population consisted of around 18% managers of individual engineers and around 10% managers of managers. The remaining were individual engineers. The mean work experience for our population was 8.49 years (median = 7.15 years). The distribution of our respondent population by geographical location is shown in Table 1. Most of Microsoft's engineers are based in the Puget Sound region of Washington State in the US. The rest of the US-based engineers are from North Dakota, Texas and California.

Table 1: Geographical regions of respondents

North America: USA - Washington	81%
North America: USA - California	2.2
North America: USA - Other	2.8
Asia: India	6.1
Asia: China	2.7
Asia: Other	0.8
Middle East: Israel	0.8
Europe	2.0
South America	0.4

In Microsoft, as in almost all other companies, people in the same team work on common projects. Usually teams are defined by the fact that they have a common management chain and work on similar projects. We wanted to determine the physical collocation of teams in Microsoft to better understand the proportion of teams that were not physically collocated. Figure 1 below shows the team collocation information for our respondent population. More than 80% of our overall respondent population is in the same building, indicating that a majority of the teams at Microsoft are collocated. Approximately 7% of teams are distributed across different countries.

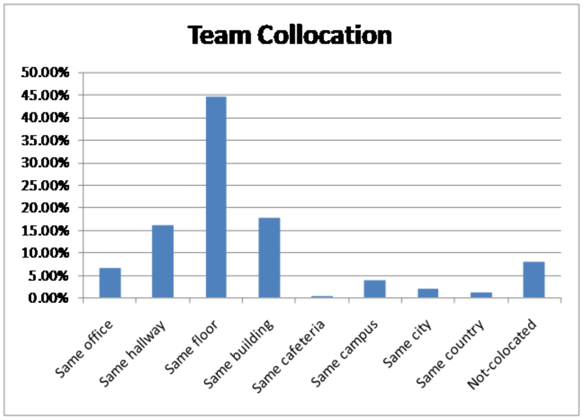

Figure 1: Team Collocation distribution for respondent population

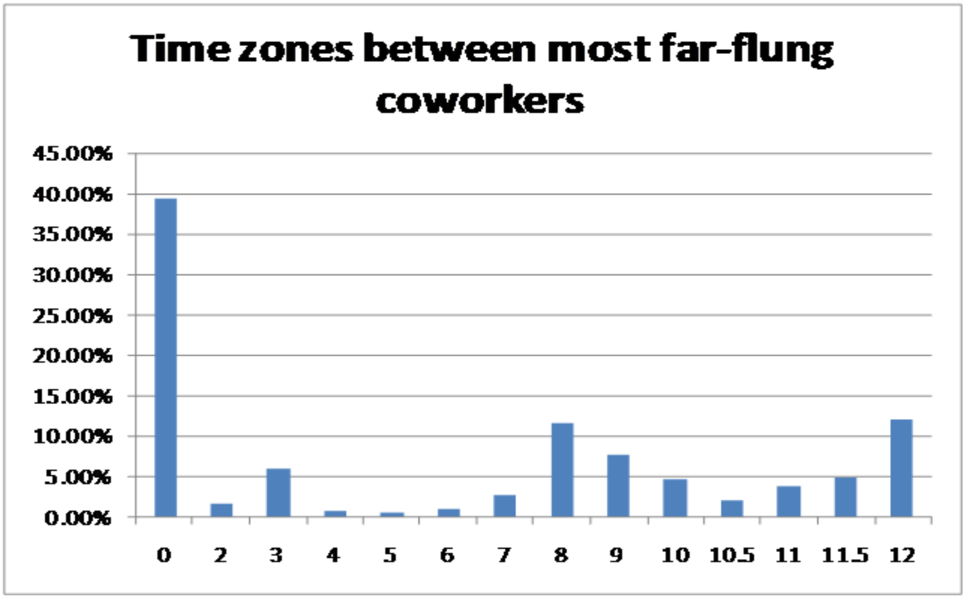

Figure 2: Time zone differences

To determine the prevalence of GSD at Microsoft, we asked in our survey how far the farthest co-worker/collaborator was in terms of time zone(s). This would serve as a good indicator of the proportion of our respondents who would have to work with people outside of their geographical location. Figure 2 shows the results of our question based on our survey respondents (results for zero and one time zone apart were conflated due to a survey recording error). We observe from our survey respondents that more than 50% of the overall population has to collaborate with co-workers greater than three time zones from their own. We call out this number since any worker based in the US with a co-worker more than three time zones away is working with someone outside their own country (Continental United States). A point to note is that this computation would not work if a person has co-workers in other North American countries. Since none of our survey respondents (as shown in Table 1) are from those regions, we avoided that issue. The results are shown with a maximum of 12 hours difference, even though respondents answered with numbers up to 24. We subtracted a respondent's answer from 24 to merge these into a day-agnostic view of time zone difference.

Considering different ways to split the data, 59% of managers have to deal with a co-worker more than three time zones away. 73% of managers of managers have to work with someone more than three time zones away. Developers are less likely on average (42%) to work with those outside their country, while program managers are more likely (63%). Testers are average (54%). Splitting by country, 93% of workers in China, and 84% in India work with those outside 3 time zones, whereas only 48% of workers in the US do. A point to note is that Microsoft has its headquarters in Redmond, WA, USA a suburb of Seattle. Seattle is nine hours away from Europe, nine hours from China, and twelve hours from India, creating a difficult coordination challenge for our software teams.

From the viewpoint of the software development process, there is no difference in the results if we split on whether or not developers are using Agile development [7], but 71% of those who *plan* to use Agile are working with someone more than three time zones way, whereas only 46% of those who do not plan to use Agile are.

5. Coordination in GSD

In addition, to collecting demographic information on GSD, we also asked respondents to report on how distributed development affects their coordination (benefits and problems), and to tell us how they have addressed some of these coordination problems in the past year.

The main benefits according to the respondents for distributed development can be divided into tools, processes and culture. GSD helps induce more rigorous documentation and automated SCM systems, and encourages experimentation with communication tools outside of email such as wikis and mobile devices. It improves software processes by producing better communication about expectations, spreading tribal knowledge, and encouraging a strong release management team. Culturally, it exposes workers to new customs, ideas, and new ways of doing business, and allows employees to make key contacts around the globe who are subject-matter experts in their field. An overall benefit to GSD is that discovering best practices in distributed development helps workers get better at co-located development as well.

Almost all respondents said that the difference in time zone caused the majority of their coordination problems with distributed teams. Many wished for some magical device that could shift time so that despite their time zone, people could video conference with one another at a reasonable time for everyone. Others expressed frustrations with playing “email tag”, where questions and responses could be delayed by over 24 hours. Many workers reported working non-standard business hours to communicate more easily with the other team, but suffered increased tension in their home lives. Respondents scheduled work and meetings outside standard business hours to overlap with the other team’s work hours. As shown in Figure 2, almost 50% of workers work 8 hours (equivalent to a standard US business day) or more away from their colleagues. Extending communication hours helps immensely to unblock distributed coworkers quickly. Meetings are also recorded for later review by team members who could not attend.

Improving coordination amongst geographically distributed teams was an important goal for our respondents. Some ways that workers have improved coordination over the last year are by scheduling visits to the location of the other team. These visits served to improve understanding of work practices, priorities and environment, to let employees ask for help, to resolve misunderstandings, to “put faces to the names,” and to get to know one another. They help to set expectations better than can be done through teleconferencing.

One respondent suggested contacting remote teams about his concerns and requirements well in advance so there would be enough buffer time to respond to issues. Another created an email mailing list to promote awareness of source code changes. Two chose point people to be the main communication conduit to the non-collocated team. A more radical tactic was to temporarily move the entire team to work together during the planning and design phases of application development.

6. Threats to Validity

As with all surveys, it is likely that our survey could have been influenced by researcher bias. This is alleviated to some degree by the fact that both the researchers conducting this survey (the authors) belong to Microsoft Research, the research division of Microsoft, and have little motivation to bias the findings regarding GSD either way.

Another threat to validity is that these results are from only one company, Microsoft. This is a general issue with all empirical studies. Researchers become more confident in a theory when similar findings emerge in different contexts [2]. We do not attempt to generalize our results outside of Microsoft, though we intend that our case study contributes preliminary empirical evidence, and hope to encourage its replication in other large companies to build an empirical body of knowledge. Also, the results are not comparable with companies that predominantly work on globally outsourced projects.

7. Conclusions

There has been little empirical evidence on the extent of propagation of GSD activities with companies. This paper presents preliminary results from a large survey deployed at Microsoft Corporation. Based on responses from 511 engineers at Microsoft we conclude that,

- A majority of engineers have to deal with GSD (> 50%).
- GSD is not restricted to senior management, though a higher proportion of managers (59%) and managers of managers have to deal with it (73%).

These conclusions indicate the widespread propagation of GSD activities within Microsoft. This data should serve as a first point in motivating tools focused at common developers and testers for commercial day-to-day usage that would streamline their job functionalities from a GSD perspective.

Contact

Researchers interested in replicating this study should contact the authors to obtain an editable/reusable copy of our survey.

References

[1] R. Agarwal, Nayak, P., Malarvizhi, M., Suresh, P., Modi, N., "Virtual Quality Assurance Facilitation Model", *Proceedings of Second IEEE International Conference Global Software Engineering*, pp. 51-59, 2007.

[2] V. Basili, Shull, F., Lanubile, F., "Building Knowledge through Families of Experiments", I*EEE Transactions on Software Engineering*, 25(4), pp.456-473, 1999.

[3] M. Bass, Herbsleb, J., Lescher, C., "Collaboration in Global Software Projects at Siemens: An Experience Report", *Proceedings of Inter-*

national Conference on Global Software Engineering, pp. 33-39, 2007.

[4] R. D. Battin, Crocker, R., Kreidler, J., Subramanian, K., "Leveraging resources in global software development", *IEEE Software*, 18(2), pp. 70-77, 2001.

[5] A. Begel, Nagappan, N., "Usage and Perceptions of Agile Software Development in an Industrial Context: An Exploratory Study", *Proceedings of Empirical Software Engineering and Measurement (ESEM)*, pp. 255-264, 2007.

[6] F. P. Brooks, The Mythical Man-Month, Anniversary Edition: Addison-Wesley Publishing Company, 1995.

[7] A. Cockburn, Agile Software Development. Reading, Massachusetts: Addison Wesley Longman, 2001.

[8] D. Damian, Moitra, D., "Guest Editors' Introduction: Global Software Development: How Far Have We Come?" *IEEE Software*, 23(5), pp. 17-19, 2006.

[9] J. D. Herbsleb, Moitra, D., "Global software development", *IEEE Software*, 18(2), pp. 16-20, 2001.

[10] S. Narayanan, Mazumder, S., Raju, R., "Success of Offshore Relationships: Engineering team structures", *Proceedings of International Conference on Global Software Engineering*, pp. 73-82, 2006.

[11] B. Sengupta, Chandra, S., Sinha, V., "A research agenda for distributed software development", *Proceedings of International Conference on Software Engineering*, Shanghai, China, pp. 731-740, 2006.