

Do Software Development Projects Dispersed Across the Globe Deliver the Goods?

Section: Technology & Information Systems

By: ROY KHENG

Publication: **Knowledge@SMU** 03/11/2007

Page: N.A.

No. of words: 1100

In today's world, the international flow of investment, people and ideas are increasingly commonplace. Multi-billion dollar infrastructure projects and complex business-to-business software development are implemented across borders and time zones, spanning the globe. Enterprises seek out cost-effective talent no matter the location. Constantly expanding business networks make higher demands on computer systems and networks, compelling technology companies to assign software development duties across many dispersed centres in the US, India, China and elsewhere.

In actual practice, does this distributed development model work well? Singapore Management University information systems professors Narayan Ramasubbu and Rajesh Krishna Balan examined this question in an award-winning paper, "Globally Distributed Software Development Project Performance: An Empirical Analysis," presented recently at a joint meeting of the European Software Engineering Conference and the ACM SIGSOFT symposium.

Commenting on the purpose of their research, Ramasubbu said, "For some customers the primary motive for outsourcing is savings because of labour cost differences. For some it is human bodies as they don't have enough capacity -- more than cost [considerations], they want to get the project done within a tight schedule. And for others it is capability; for example an insurance company's vendor might have done something for their competitor and the vendor might have some knowledge [they could use]. However, our interest was not primarily about outsourcing but the effects of locating software development projects in multiple sites, across borders and time zones."

More Time-consuming

Ramasubbu and Balan developed empirical models of software project performance and verified these against data and information gathered from forty two large-scale, real-world projects. The projects were executed by a major software company across two development centres located in India and the US.

Over a two-year period, the authors reviewed both qualitative and quantitative aspects of the company which had attained Level 5 of the Capability Maturity Model (CMM, instituted by the Software Engineering Institute at Carnegie Mellon University), the highest rating possible, indicating mature and well documented processes and structures. The company also implemented uniform human resource practices across its global sites.

CMM is a process model based on software best practices which are effective in large-scale projects. Software code size was measured in terms of function points over the entire project code base.

According to the researchers, code size is a widely used control variable for software quality models as it captures both the magnitude of the project and most of the complexity involved in developing the application.

Measuring Dispersion and Quality Management

The research model the authors used looked at project performance from the perspectives of work dispersion and quality management. They measured work dispersion between the two development centres while quality management was measured along three dimensions - prevention-based, appraisal-based, and failure-based approaches.

The prevention-based approach towards quality management looked closely at activities such as training that was carried out primarily to reduce the occurrence of errors. The authors computed a score based on the percentage of total development effort spent on training, project planning and configuration management activities.

The appraisal-based approach examined progress assessment, performance and the quality of intermediate artifacts at various stages of development. A score was assigned based on the percentage of the total development effort spent on peer reviews of requirements, design, status reviews and code inspection.

Finally, the failure-based approach looked at testing conformity of the product against customer specifications and the subsequent error correction process. A score was assigned based on the percentage of the total development effort spent on unit tests, module integration tests and systems tests.

Beyond the hard data they gathered, the researchers were also present in the field to observe the software development process using an ethnographic observation procedure. As they explained, “To complement our understanding of the software processes and culture at the sites, we conducted structured interviews with two senior business development managers, four project managers, and with ten randomly selected project team members.”

A key finding of the study was that dispersion, even in high maturity environments, had a negative impact on productivity. This loss of productivity was further compounded and increased exponentially as dispersion increased. According to the authors, what this means for companies who practise dispersed project development is that the marginal increase in yield is much higher as dispersion increases. “Firms should seriously consider this effect when initiating distributed development and increasing the number of sites,” said Ramasubbu.

Mitigating the Negatives

The news isn’t all bad, however. For enterprises engaged in dispersed software development, there are structures and processes they can put in place to mitigate the negative aspects of dispersed software development. From the quality management perspective, the study showed that appraisal-based approaches have the highest positive effect on software development productivity with failure based approaches coming in next. Prevention based approaches seemed to have no statistically significant effect.

In terms of conformance to clients’ specifications, the researchers found that failure-based approaches have the largest effect on conformance quality followed by prevention-based approaches. Appraisal-

based approaches had no statistically significant effect on conformance quality.

Bottom line

Overall, the authors found that no one approach can be a panacea for the problems associated with dispersed software development. The choice of approach would depend heavily on the context of the project as different quality management approaches have significantly different effects on the various dimensions of project performance. The researchers also caution that even though the study was conducted in a high maturity environment, and data was tracked for a two-year period, not all the subtleties of software development were captured.

According to the authors, ‘internationalising’ a project has many consequences. Dispersed project teams must contend with distance, culture and language differences. In terms of the basic systems development life cycle, internationalised projects tend to consume more time. Behavioural research has also indicated that distributed projects using remote teams, even with the support of advanced communications technology, suffer decreased productivity due to problems with administration and synchronisation.

The study findings indicated that the more project sites, the lower the productivity. Firms therefore should seek to mitigate these drawbacks by using structured processes and appropriately balancing their quality management approaches. As more companies outsource scores of functions that make up day-to-day operations in the modern business world, firms need to take a close, hard look at the implications of locating projects across their global locations.

Ramasubbu advised, “The bottom line is that firms need to account carefully for the inevitable reduction in productivity and quality when determining whether to move software production to a second or even third location. Firms that institute high standards of software development will have a higher chance of mitigating the effects of dispersion.”