

Global Software Team and Inexperienced Software Team

G

Kim Man Lui

The Hong Kong Polytechnic University, Hong Kong

Keith C. C. Chan

The Hong Kong Polytechnic University, Hong Kong

INTRODUCTION

Given that the number of qualified programmers cannot be increased drastically and rapidly, software managers in most parts of the world will likely have to live with a human resources shortage in this area for some time. One way of dealing with this shortage is to form global software teams in which members are recruited from all over the world and software is developed in a distributed manner. Forming such a global software teams can have many advantages. In addition to alleviating the problems caused by scarcity of human resources, programmers on a global team would be free to work without being confined by physical location.

Although forming global software teams may increase the size of the pool of programmers that can be recruited, both team quality and software quality are issues of great concern. Some software companies would prefer to establish a global software team with software programmers in developing countries, such as China, Poland, and South Africa (Sanford, 2003). Given the tremendous salary gap between skilled and unskilled developers or between developed and developing countries, it is not difficult to see that maintaining a team with a proportion of less experienced members significantly reduces running expenses (Figure 1). On the other hand, however, it would present the problem of managing inexperienced programmers.

This chapter shares our experience of managing inexperienced software teams in China. To simplify our discussion, we deal separately with the two topics of inexperienced software teams and global software teams. However, it should be noted that a global software team can be composed of both inexperienced and experienced software subteams. We categorize the problems in these two types of software teams which will help software managers learn more how to manage the two types of software teams.

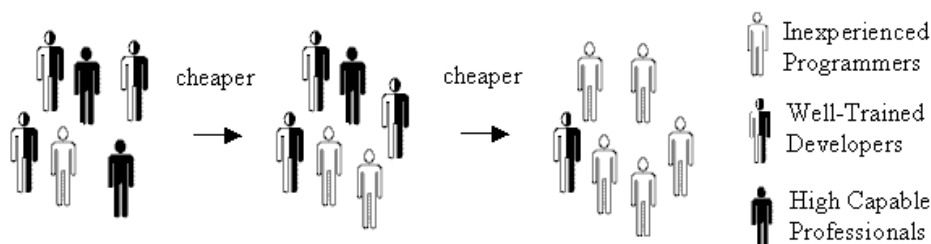
BACKGROUND

This section reviews real cases that have driven the formation of an inexperienced software team and a global software team. The motivation behind managerial decisions to build such teams is both financial and environmental.

Discovering Developing Areas: Inexperienced Software Team

Active rural industrialization involves manufacturing plants moving from more developed regions to less developed ones so as to exploit the lower costs of land, labor, and distribution channels (Otsuka, 2001). In less developed regions it is an

Figure 1. In the software world, the proportion of highly professional to less experienced teams may fall as companies operate under the constraints of tighter cash flows and for reasons of cost replace more experienced programmers with junior programmers and seek to avoid the costs associated with the professional development of senior programmers.



easy matter to recruit labor for manufacturing, but these plants also require management information systems (MIS) and it is not at all easy to find and recruit the IT professionals that are required to develop an integrated, customized MIS.

In developing countries, the demand for IT professionals in larger cities is currently so high that it is almost impossible for any manufacturing plant in a rural area to recruit people. As a result, programmers in poor rural areas are usually inexperienced. Even though the alternative of employing software expatriates might sound reasonable, it is feasible to recruit one or two highly qualified programmers from developed regions but it is not practical to recruit a team of them. Instead of in-house development, we might evaluate a third-party solution. The additional expenses incurred in purchasing vendor products, in consultancy services, annual maintenance, version upgrading, training, traveling, and so forth, outweigh the savings that are sought by setting up in the country in the first place.

In less developed areas, many programmers do not receive proper training in computing. In addition, the turnover rate is typically high. As soon as they have received even a little training, many workers will seek a job with better career prospects in a more developed city. The result is that the project manager always has to work with inexperienced programmers and in a constant mood of crisis management as the high turnover rate is aggravated by resignations without notice as people tender their resignation and leave on the same day. Clearly, the process of handing over work is unsatisfactory, teams are constantly under-staffed and the working environment suffers, making work elsewhere an even more attractive option.

One may suggest that educating inexperienced people or allocating suitable jobs according to an individual's ability should fulfil the same purposes. However, when the knowledge and experience of staff members is not aligned with the tasks assigned, the learning curve can be steep and long (Amrine, Ritchey, Moodie & Kmec, 1993). Nevertheless, when a staff member becomes well trained in some less developed regions in China, or in a small company in Denmark, for example, the determination of the staff member to look for better job prospects elsewhere will become stronger. Training, therefore, does not provide a promising solution in this case. In contrast with what we might find in well-developed regions, in China, staff who are provided with certified professional programs will leave a company or a less developed region even sooner. Senior managers are disturbed by this phenomenon and say that they are always training another company's staff. The idea of allocating developers according to their skill set is not feasible when all team members are inexperienced. Human resource allocation can therefore be implemented only to a limited extent. Better knowledge management, rather than adopting conventional principles, is required.

Around the Clock: Global Team

A small but ambitious company selling weight-loss and nutritional products, which was headquartered in New York, had a number of small offices of 40 employees in different parts of the world. Being close to customers is always a key to business success.

One to two staff members in each office had the task of providing IT support. When the MIS system needed to be modified to meet requirements for local processing, requests for modifications would be sent to the head office. The result was that more resources were required at the head office to provide ongoing support to the branches. Although a larger software team was thus required at the head office, IT staff at the branches might have time to spare.

The load-balancing problem got worse when the number of branches increased. The question, naturally, was to decide if it was possible to link people to establish a global software team. The team in each site then plays a role more or less as distributed agents following a communication scheme from a coordination agent.

A global software team can even be formed locally, if the team is set up in different locations within the same country or in nearby countries or regions. This means that there may not be much difference in time zones and cultures. In such situations, the term "multisite" or "distributed" software team can be used more generally to describe a software system developed by teams that are physically separate from each other in different cities of a country or in different countries. A multisite team in nearby time zones can be managed with less complexity and fewer challenges than a global team and but has more limited service hours. Clients on the other side of the world want a reply inside their own local business hours but if all teams are in the same time zone, it becomes hard to respond promptly outside office hours. In any case, the management framework required for a global or multisite team should be very similar. To further explore around-the-clock development (see Figure 2) and global development, we realize the intrinsic difference is how synchronization of work-in-progress proceeds. We concluded that challenges of managing around-the-clock tasking widely cover managerial and technical problems of non-around-the-clock global software development.

Around-the-clock development exploits time zone differences as a way to improve time-to-market. But, there has not been a model for this kind of global software development (Carmel, 1999; Karolak, 1998). It is easiest to manage a global software team with less strict synchronization among different sites. In this case, a team that is waiting for a result from another site could work on other tasks for the same project. However, in around-the-clock development, work-in-progress and communications are rigidly synchronized and the progress of a team is tied to the progress of

5 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/global-software-team-inexperienced-software/13801

Related Content

Researching Technological Innovation in Small Business

Arthur Tatnall (2009). *Encyclopedia of Information Science and Technology, Second Edition* (pp. 3292-3297).

www.irma-international.org/chapter/researching-technological-innovation-small-business/14062

Enhancing Patient Care and Care Coordination using Event Notification Systems

Steven H. Ton, Alice M. Noblin, Kendall Cortelyou-Ward and Victor A. Nunez (2016). *Journal of Cases on Information Technology* (pp. 17-27).

www.irma-international.org/article/enhancing-patient-care-and-care-coordination-using-event-notification-systems/159262

Business Process Reengineering for the Use of Distance Learning at Bell Canada

Tammy Whalen and David Wright (1999). *Success and Pitfalls of Information Technology Management* (pp. 186-199).

www.irma-international.org/chapter/business-process-reengineering-use-distance/33491

Certifying Software Product and Processes

Hareton Leung (2005). *Encyclopedia of Information Science and Technology, First Edition* (pp. 381-386).

www.irma-international.org/chapter/certifying-software-product-processes/14266

An Event-Based Data Warehouse to Support Decisions in Multi-Channel, Multi-Service Contact Centers

Andrea Brunello, Paolo Gallo, Enrico Marzano, Angelo Montanari and Nicola Vitacolonna (2019). *Journal of Cases on Information Technology* (pp. 33-51).

www.irma-international.org/article/an-event-based-data-warehouse-to-support-decisions-in-multi-channel-multi-service-contact-centers/216951