Chapter XVII

Inexperienced Software Team and Global Software Team

Kim Man Lui and Keith C. C. Chan
The Hong Kong Polytechnic University, Hong Kong

ABSTRACT

Software project management in the 21st century requires that a manager deal either with Inexperienced Software Team or Global Software Team or both. This is because well-developed and less well-developed countries have exploited information technology to various extents. The former requires managing a software team consisting of talents remotely located whereas the latter a team of local inexperienced developers. This chapter assimilates the management challenges involved and explicates how these two types of supposedly different software development are managed by one framework: Plagiarism-based Programming.

INTRODUCTION

Despite the burst of the dot com bubble and the economic slowdown in most parts of the world, the demand for programmers has never been greater. To meet the demand, some companies have started recruiting overseas and some have outsourced software projects offshore. Unfortunately, neither of them seems to be a very satisfactory solution to the manpower-shortage problem. The former requires programmers to physically relocate to a new place, and it takes time for them to adjust to a new environment and, possibly, a new and very different culture. By the time these programmers become productive, they could well be headhunted to work elsewhere. As for the latter, the risks and costs involved in managing an outsourced off-shore project could be very high.

This chapter appears in the book, Knowledge and Information Technology Management: Human and Social Perspectives edited by Angappa Gunasekaran, Omar Khalil and Syed Mahbubur Rahman. Copyright © 200, Idea Group Inc.
Given the fact that the number of qualified programmers will not be increased drastically and rapidly, software managers in most parts of the world will likely have to live with the manpower-shortage problem for some time. In fact, in the U.S. alone, according to the estimation of the Information Technology Association of America in April 2000, 850,000 IT positions were expected to remain unfilled in 2001 (Information Technology of America, 2000). Clearly, this figure is expected to be many times larger if we are to compile some global statistics.

To deal with the manpower-shortage problem, the forming of global software teams, where members are recruited from all over the world and software is developed in a “distributed” manner, have to be considered. Forming such a global software team can have many advantages. In addition to alleviating the problems caused by scarcity of human resources, programmers on a global team are now free to work without the confines of physical locations. If they happen to work in different time zones, they can develop a novel, non-stop, around-the-clock, working style (see Figure 1). A company with a global software team may therefore be able to reduce costs and improve efficiency.

Besides these benefits, forming a global software team also has advantages from the risk-management viewpoint. A software team, if located in one single location, might be brought to a complete halt due to whatever reasons such as strike, political turmoil, natural disasters, terrorist attack, etc. However, for a global software team, even with these incidences occurring in a single location, the team may likely be able to continue to function, as members in other parts of the world may be unaffected. This is also expected to increase customer confidence and trust. They will be pleased to know that support services can continue even in crisis situations.

Other than the idea of forming a global software team, some software managers are advocating for a review of software engineering curriculum to include compulsory/selective course(s) in software engineering for students majoring in other disciplines, especially engineering (Meyer, 2001). By doing so, it is hoped that more programmers can be produced more rapidly here and elsewhere.

While the idea of forming a global software team may increase the size of the pool of programmers that one can recruit, there is always the concern about quality.

Figure 1: Around-the-clock development
A Comprehensive Relational Model of Factors Influencing Knowledge Sharing: An Empirical Study
www.igi-global.com/article/a-comprehensive-relational-model-of-factors-influencing-knowledge-sharing/112063?camid=4v1a