Global-Software Development Lifecycle: An Exploratory Study

Alvin W. Yeo
Faculty of Information Technology
Universiti Malaysia Sarawak
94300 Kota Samarahan, Sarawak MALAYSIA
alvin@fit.unimas.my

ABSTRACT

This study was conducted to explore the efficacy of the global-software development lifecycle (global-SDLC), which comprises design, implementation and usability evaluation phase. A spreadsheet was adapted using the global-SDLC process to accommodate a number of cultures. The design and implementation phases were efficacious. However, in the usability evaluation phase, the usability evaluation techniques were only efficacious when participants, who were experienced computer users and participants who were familiar with the experimenter, were employed. Explanations, from cultural literature such as Hofstede, are presented and implications of these findings on the usability evaluation phase and the global-SDLC are also described.

Keywords

Global-software development, internationalisation, localisation, usability evaluation, Hoftede's cultural dimensions

INTRODUCTION

More countries are adopting information technology (IT) with the expectation of reaping the benefits of IT. However, before IT can be utilised by a particular community, software and hardware technologies may have to be customised before they can be used in that target community. For example, software in English must be adapted to Chinese if it is to be useful to the majority of Chinese. By adapting software, more cultures (in particular those which are lagging in the software development industry) can also reap the benefits of IT. Many large United States (US) software companies—US is the largest software exporter in the world—earn more than half of their revenue from outside the US. These software exporters are becoming aware of the huge markets that exist in the world. Thus, it makes financial sense to develop software for the global market. By accommodating more countries, software companies can earn more revenue from these international markets, especially, from Asia (such as, China and India, the two most populated countries in the world). In order to capture the diverse global software market, the software must be able to accommodate the different cultures. This global-software development process is a complex process, and may incur high costs as well as involve significant resources. There is, therefore, a need for software companies to identify strategies to ensure successful, effective and efficient adaptation of software to the diverse international markets.

Global Software Development Techniques

A two-step *internationalisation* and *localisation* process is recommended for developing software for the global market [14,20] Internationalisation is the process that separates the software into two components, a culture-independent and a culture-dependent component. The culture-independent component, known as the generic core, contains the bulk of the software and is devoid of culture-sensitive elements. Localisation is defined as the process of providing the culture-dependent component of a particular target culture. (In this paper, software adapted to a particular culture means that the software is customised for a group of people, who feel, act, and think similarly.) The culture-sensitive elements, comprising dialog messages, error messages, and menu names are localised (translated) and stored in a *message file*. There is a different message file for each market/culture. If the software is required in a new language, only the localisation process takes place; there is no modification of the software's generic core (culture-independent component of the software). Maintenance is also easier as only the generic core is modified. Thus, to realise software for the global market, the global-software development lifecycle (global-SDLC), which includes internationalisation and localisation, is utilised.

Global-Software Development Lifecycle

There are three phases in the global-SDLC. These phases are design (which incorporates internationalisation and localisation), implementation, and usability evaluation [9]. In the design phase, software developers first identify the target markets. The goal of the design phase is to derive a generic-core (the program) via internationalisation, and

SIGCHI'01, March 31-April 4, 2001, Seattle, WA, USA.
Copyright 2001 ACM 1-58113-327-8/01/0003…$5.00.