Guidelines for Improving the Development of Web-Based Applications

Abdesselam Redouane
3360 Paul Anka Dr., Suite 211,
Ottawa K1V 9S2, Canada
redouane@sprint.ca

Abstract

In this paper we present some useful guidelines for the development of web-based applications. We comment on the current trend of development, its drawbacks, and then propose some guidelines to improve the quality of such practice. These guidelines are being investigated and it turns out to be very convenient and helpful.

1. Introduction

Web-based applications are becoming so popular in our daily life in the sense that it would not pass a day without we use them. These applications range from simple to more sophisticated ones, where millions of dollars, in revenue, are generated. Developing, testing and quality assuring these applications become a challenging task.

Web-based companies have very stiff and stringent conditions. They have limited resources. This will hinder the quality of the product and ultimately the success of these companies. It is usually the case that the few people, who carried out the development, will also perform the testing of the end product. This is a poor practice, as it does not allow the test to be carried out rigorously and it will be certainly biased.

The development of web-based applications can be, at least, divided in two parts. A content presentation part and behind the scene data processing part. The former is carried out by an HTML group whereas the latter by a small group of programmers using any server-side scripting languages like Perl, C, etc… Although communication between these two groups is not a barrier, testing however, the two main parts of a web-based application, is performed by each group separately, leaving the testing of the application from end-to-end an unfinished job.

In this paper, we provide some guidelines, which are being investigated and it turns out to be very convenient and helpful.

2. Guidelines Description

The guidelines are described in the chronological order in which they should be applied and we call them the seven-rules.

RULE 1: Collect the requirements from users and/or management. There should be no intermediate person between the requirements original source (users or management) and developers. We stress this point and encourage direct communication because it is much more productive and very helpful to developers.

RULE 2: Translate informal requirements to a formal or semi-formal specification using any formal notation, which both teams are familiar with, like DSL [1]. By semi-formal, we mean a formal specification augmented with some informal explanation, as it might be the case that both teams (HTML and programmers) are not at the same level of technical expertise. This approach can be very helpful even to the end user during validation of the specification.

RULE 3: Generate, as much as you can, test cases from the formal specification. It should be complete in the sense that it would not leave any component without a proper test case. This list can be long with a site with hundred of pages and containing thousands of links. In this case, a modular test suite is the key solution.

RULE 4: Each team will carry their share of the development. The HTML group will work on their parts and the programmers will do so on their parts.

RULE 5: Each team will do their testing according to RULE 3.

RULE 6: The integrated application, HTML and server-side generated pages, is tested by each team in full against RULE 3. It would be very helpful to have a tool, which manage these tests, where developers can log bugs found and check them as fixed when they have been corrected.

RULE 7: Iterate on RULE 6 until the log is clear of any bugs.

References

[1] A. Redouane, An Investigation of a Definitional Specification Language and the Development of a Support Tool, Ph.D. Thesis, Computation Department, UMIST, 1990.

Proceedings of the Fourth International Workshop on Web Site Evolution (WSE'02)
0-7695-1804-4/02 $17.00 © 2002 IEEE