7 Web Application Testing

Giuseppe A. Di Lucca, Anna Rita Fasolino

Abstract: Web applications are characterised by peculiarities that differentiate them from any other software application. These peculiarities affect their testing in several ways, which may result in harder than traditional application testing. Suitable methods and techniques have to be defined and used to test Web applications effectively. This chapter will present the main differences between Web applications and traditional ones, and how these differences impact the testing of Web applications. It also discusses relevant contributions in the field of Web application testing, proposed recently. The focus of the chapter is mainly on testing the functionality of a Web application, although discussions about the testing of non-functional requirements are provided too. Readers are required to have a general knowledge of software testing and Web technologies.

Keywords: Web engineering, Web application testing, Software testing.

7.1 Introduction

In the last decade, with the wide diffusion of the Internet, a growing market request for Web sites and applications has been recorded. As more and more organisations exploit the World Wide Web (WWW) to offer their services and to be reached by larger numbers of customers and users, the request for high-quality Web applications satisfying security, scalability, reliability, and accessibility requirements has grown steadily. In such a scenario, testing Web applications to verify their quality became a crucial problem.

Unfortunately, due to market pressure and very short time-to-market, the testing of Web applications is often neglected by developers, as it is considered to be time-consuming and lack a significant payoff [11]. An inversion of this trend may be obtained if testing models, methods, techniques, and tools that allow testing processes to be carried out effectively and in a cost-effective manner are available.

Although Web application testing shares similar objectives to those of “traditional” application testing, there are some key differences between testing a traditional software system and testing a Web application: the specific features exhibited by Web applications, and not included in other software systems, must be considered to comprehend these differences.
A Web application can be considered as a distributed system, with a client–server or multi-tier architecture, including the following characteristics:

- It can be accessed concurrently by a wide number of users distributed all over the world.
- It runs on complex, heterogeneous execution environments, composed of different hardware, network connections, operating systems, Web servers, and Web browsers.
- It has an extremely heterogeneous nature that depends on the large variety of software components that it usually includes. These components can be built by different technologies (i.e. different programming languages and models), and can be of a different nature (i.e. new components generated from scratch, legacy ones, hypermedia components, COTS, etc.).
- It is able to generate software components at run time according to user inputs and server status.

Each aspect described in the previous list produces new testing challenges and perspectives. As an example, effective solutions need to be identified for executing performance and availability testing to verify a Web application’s behaviour when accessed concurrently by a large number of users. Moreover, as users may utilise browsers with different Web content rendering capabilities, Web applications must be tested to make sure that the expected application’s behaviour using different Web browsers, operating systems, and middleware is the one expected. Another critical feature of a Web application to be specifically tested is its security and ability to be protected from unauthorised access. The different technologies used to implement Web application components influence the complexity and cost of setting up a testing environment required to test each component. In addition, the different mechanisms used to integrate distributed components produce various levels of coupling and inter-component data flow, impacting the cost for being tested effectively. As for the existence of dynamically generated software components, the issue here is to cope with the difficulty of generating and rerunning the same conditions that produced each component.

Finally, Web application testing also needs to take into account failures in the application’s required services/functionality, to verify the conformance of the application’s behaviour to specified functional requirements. Considering that the components of a Web application are usually accessed by navigation mechanisms implemented by hyper-textual links, a specific verification activity also needs to be devised to check link integrity, to assure that no unreachable components or pending/broken links are included in the application.