

Visual Testing Of Software Using Screen Recording

Blueberry Software

August 2011



Visual Testing Of Software Using Screen Recording

August 2011

Abstract

Software testing is a time-intensive process which is often overlooked during the planning of a software project. Yet this is a critical stage to ensuring product quality, and resolving hard-to-fix issues can easily turn an otherwise successful project into a costly endeavour. This article examines a visual testing tool designed specifically to address the communication problems that are commonly encountered between software developers and testers.

Introduction

Professional software development companies understand the need for strict regimes of software testing and quality assurance to ensure a product is fit for purpose before it is sold.

There are a number of ways to go about implementing these processes - however, they can all be affected by problems that increase costs and reduce efficiency. Furthermore, market pressures can force a project to be developed with unskilled programmers or insufficient time or money, resulting in increased errors, particularly when designing complex software.

Common problems stem from a lack of communication between developers (charged with fixing faults) and testers (charged with identifying faults), and they are caused by two scenarios:

- The first is that, quite simply, information and descriptions of faults may get lost in translation when working in a multi-national team, or when outsourcing testing-related tasks to foreign countries.
- The second scenario, which is more common, is that the tester is not sufficiently clued up with the deployed technology to convey the details of the problem accurately.

This is particularly likely in black-box testing, where testers of software are not required to provide insight into the programmatic cause of a fault and are therefore unlikely to be as technically-minded as their developer colleagues.

Another problem with traditional methods of software testing is the replication of faults.

In order to implement a fix for a software failure, the developer needs to confirm that it exists in the first place. This is done by mimicking the set of operations performed by the tester, which led to the fault.

It is a time-intensive process, and suffers from its own problems:

- What if the fault does not occur under the same circumstances for the developer?
- Does he assume the tester described the fault incorrectly?
- What if it's a platform-dependent issue that affects the tester's system but not the developer's?

The replication of faults is not an absolute science, and the ambiguity and uncertainty it could bring will cost time and money to overcome.

Visual Testing

The goal of a visual testing tool is to provide developers with the ability to examine what was happening at the point of software failure. Ideally, the testing tool should present the data in such a way that the developer can easily find the information he requires, and the information is expressed clearly. Visual representation is clearly the most effective way of conveying this information in practice, but finding a visual test tool that meets these requirements is one of the main problems in implementing this sort of methodology.

One solution is to use a screen recording tool designed specifically for software developers and testers, such as BB TestAssistant.

At its core, it's built on the idea that showing someone a problem (or a test failure), rather than just describing it, greatly increases clarity and understanding.

Programs like BB TestAssistant enable a software tester to record the entire test process – capturing everything that occurs on their system in video format. These videos can be supplemented by real-time input via webcams (which appear as picture-in-a-picture) and audio commentary from microphones.

A screen recorder for developers provides solutions to two fundamental problems mentioned earlier:

- Quality of communication is increased dramatically because testers can show the problem (and the events leading up to it) to the tester as opposed to just describing it.
- The need to replicate test failures will cease to exist in many cases. The developer will have all the evidence he requires of a test failure and can instead focus on the cause of the fault and how it should be fixed.

In the case of BB TestAssistant, Blueberry Software has developed a purpose-built platform on which to conduct software testing, and thereby extend the functionality of their visual testing tool to include such useful features as QA system integration (complete with open API) and parallel reviewing of system event logs and external log files.

BB TestAssistant integrates fully with QA systems such as JIRA and Bugzilla. In other words, after recording a test, the software tester is able to use BB TestAssistant to automatically create defect reports which comply with industry-standard QA system formats.

Of course, not everyone uses the same QA systems. For this reason, the integration API has intentionally been left open to make it easy for programmers to enable integration with other defect tracking systems.

Screen recorder based visual test tools like BB TestAssistant allow developers to review system event logs and application log files in parallel to the video of the test. This means developers can draw from multiple sources of information relating to the software failure, such as:

- the video itself,
- any edit features such as webcam,

- annotations and audio commentary; and
- properly synchronised event logs and log file information in order to solve the problem.

The inclusion of synchronised access to log files and system event logs means that the developer is able to analyse what's going on inside the program at the same time as viewing the tester's descriptions. This combination of low-level detail with high-level overview of the problem assists the developer in pinpointing the cause of the error quickly.

Agile Methods

Visual testing is particularly well-suited for environments which deploy Agile methods in their development of software. Agile methods require greater communication between testers and developers, which plays to the strengths of a program like BB TestAssistant for creating very detailed records of failures with very little effort.

Agile methods of software development encourage collaboration within small teams, as well as making the process as personal and informal as possible: any member of a department should be able to contact any other if they have a question or concern.

Visual testing using a screen recorder tool is particularly well suited to Agile methods as it provides a personal touch, including webcam and voice recording on videos.

Exploratory & Ad Hoc Testing

Ad hoc and exploratory testing are important methodologies for checking software integrity, because they require less preparation time to implement, whilst important bugs can be found quickly.

In ad hoc testing, where testing takes place in an improvised, impromptu way, the ability of a test tool to visually record everything that occurs on a system becomes very valuable. This is equally the case with exploratory testing, which is a more structured approach to software testing and is usually carried out by skilled, independent testers.

Both of these test methods rely on an adaptable and learning approach to testing. For this reason, it is sometimes difficult to replicate a failure faithfully from memory and report it effectively.

Using a screen recording tool eliminates the need for testers to remember exactly what they did before they hit a bug, so they can concentrate on performing the tests in a more informal, creative way.

Acceptance and Usability Testing

A screen recorder tool is also ideal with customer acceptance and usability testing, because it can be used by anyone involved in the development process. For the customer, it becomes easy to provide detailed bug reports and feedback, and for program users, the tool can record user actions on screen, as well as their voice and image, to provide a complete picture at the time of software failure for the developer.

Examples of use include:

- By developers to demonstrate current functionality of the product to the customer representative.
- By dedicated software testers to produce thorough test reports on the product, highlighting potential failures for developers to fix.
- By the customer representative to test the product according to their expectations and also to convey additional, desired functionality: "it would be helpful if the program did x when I clicked y".

BB TestAssistant also complements version control software like Subversion, in that it can be used to document development progress by showing how the product performed and behaved on a given date or on completion of a given iteration or sprint.

Summary of Benefits

- Visual testing, using a screen recorder program like BB TestAssistant, negates the requirement for QA staff to have special training or skills in order to create detailed movies that show defects.
- Visual testing improves tester-developer communication by enabling the problem to be shown quickly and clearly, so simplifying the testing process.
- It is applicable to different kinds of testing:
 - Usability Testing – where you can see exactly how your user intuitively interacts with the program via the program's support for webcams, microphones and system event-logs.
 - Automated Testing – leave the program recording an automated test session to obtain a permanent record of exactly what happened while you were away.
- Its use in projects complements Agile methods of software development by allowing you to track iteration progress and involve customer representatives in a more natural fashion.

BB TestAssistant Feature Summary

- **Event log:** The Windows event log, external log files, key presses and mouse clicks are viewed side-by-side with the movie to show the status at any instant in time.
- **QA system integration:** BB TestAssistant integrates fully with current industry-standard QA systems such as JIRA and Bugzilla. The integration API is open, so it's easy to add support for a preferred QA system as well.
- **Record everything:** Everything the tester sees on their screen will be recorded by BB TestAssistant, even complex Windows Aero animations.
- **Editing:** Once the video has been recorded, you can edit it from within BB TestAssistant to include annotations and audio commentary (which can also be supplied in real-time). You also have access to other video editing options such as clipping, cropping and quality adjustment.
- **Skip to edits:** Each edit made by the tester acts as an anchor within the video that the developer can choose to skip to. This is particularly useful in long videos, or where the developer knows exactly which part he's looking for.
- **Projects:** This new UI feature allows users to define projects. A project holds common configuration settings related to a single application or set of applications which a tester is working with.
- **Remove Inactive Periods:** This feature allows the user to identify and remove periods of inactivity within a movie.
- **Hide Other Processes:** This security / data protection feature allows the user to restrict BB TestAssistant to record only a specific set of processes.
- **Export to Word:** This feature allows a user to mark important points in a movie and add notes, then automatically produce a Word document containing screenshots of all these points, together with the notes.