Gil Zilberfeld on...

Testing times

Unit testing should be an integral part of web development, argues **Gil Zilberfeld**

Unit testing is a practice that's gaining more and more of a presence in the modern software development landscape, especially in agile environments.

A unit test is a short function to test the behaviour of a small unit of production code, producing a pass/fail result. Unit testing is a powerful (and in our opinion, essential) tool for ensuring code quality by enabling developers to catch bugs while still in the development stage. (It's not a bug if the problem was solved before the code reached QA, right?) Used consistently, unit tests can cover a major part of your code and its workflows. They're especially effective in preventing working code from 'breaking' because of modifications, because they pinpoint the problem for the developer as soon as it's introduced. Additionally, when devs know that they have unit tests as a 'safety net', they're not afraid to make functional changes and refactor code for the purpose of readability.

We believe, and most developers agree, that unit testing is a crucial tool for producing working software, with fewer bugs, quickly and consistently.

Better chances

Unit testing does not eliminate all bugs, nor does it make quality assurance unnecessary. But unit testing can increase the chances of new features working correctly, because it becomes the developer's responsibility to verify with unit tests that he or she is delivering working code. It also greatly reduces the time spent by developers and QA staff finding and fixing regression bugs (bugs introduced by changes that break existing functionality). Unit testing can free the development process from the 'one step forward, two steps back' experience that often occurs when you deliver new features, only to find that existing ones have stopped working.

As an application grows, and gets more complex, time spent finding and fixing bugs can increase too, unless some of the burden shifts over to automated testing tools. Unit testing frees up QA resources to focus on new functionality and system-wide testing.

We've mentioned the benefits of unit testing for any software product, but how does this apply particularly to web development? One area to benefit is web app architecture, which involves a server, multiple client applications and environments. The amount of manual testing needed is compounded because of the need to test the client application in multiple browsers, in several screen resolutions, and often in mobile environments too. Unit testing can free up resources to focus more on manual client-side testing. Some unit testing tools provide the extra benefit of 'mocking' or simulating client interactions, so server-side testing can be partially automated.

"Unit testing is a powerful, in our opinion essential, tool for ensuring code quality" Gil Zilberfeld

As well as the technical reasons why unit testing is important to web development, there are business-related reasons why a web-based offering must have the level of quality and confidence in change that it can provide. A newly deployed version of a web platform is instantly exposed to millions of users. A malfunction caused by a bug has the potential to incur significant damage to the business's reputation – and possibly to its income. The broader a site's user base, the larger the potential damage. In the case of a site that processes commercial transactions, a malfunction could even be cause for litigation.

Right first time

Furthermore, as a website is usually not unique in its offering, if something isn't working on the site it's easy for users to move over to competitors in just a few clicks. Customer allegiance is more reliable with software applications, which must be purchased and installed, but a web app enjoys no such loyalty.

Unit testing is a way to ensure developers produce high-quality code, first time, and maintain quality over time despite software updates and product changes. It frees up QA resources to focus on new features and higher-level functionality, and gives product managers and programmers confidence to bring frequent changes into their web application.

Recent advances in automated unit testing tools, mean they're now convenient, richly featured and affordable, making it easy to incorporate unit testing into any software development process. ●

Gil Zilberfeld is product manager at Typemock (www.typemock.com). He blogs at www.gilzilberfeld.com