


Buzzle.com[®]

Intelligent Life on the Web

[Home](#) [Topics](#) [Featured Articles](#)

Search

Software Testing

[Software Testing Tutorial](#)
[Software Testing Life Cycle](#)
[Software Testing - Bug Life Cycles](#)
[Manual Software Testing Process](#)
[Manual Testing Tutorial](#)

Ads by Google

Improving Podcasts

Agile, .NET, and Java podcasts from the experts at Improving.

www.improvingpodcasts.com

Software Testing Methodologies

There are different types of software testing methodologies used in the field of software testing and quality assurance. In the following article, we'll take a look at various software testing techniques and methodologies that are in practice today.

 +22
Recommend

Software testing is an integral part of the software development life cycle (SDLC). Testing a piece of code effectively and efficiently is equally important, if not more, to writing it. So what is software testing? Well, for those who are new to software testing and quality assurance, here are few useful facts.

Software testing is nothing but subjecting a piece of code to both, controlled and uncontrolled operating conditions, in an attempt to observe the output and examining whether it is in accordance with certain pre-specified conditions. Different sets of [test cases](#) and testing strategies are prepared, all of which are aimed at achieving one common goal - removing bugs and errors from the code and making the software error-free and capable of providing accurate and optimum outputs. There are different [types of software testing](#) techniques and methodologies. A software testing methodology is different from a software testing technique. We'll take a look at few of the software testing methodologies in the latter part of this article.

Software Testing Methods

There are many different types of testing methods or techniques used as part of the software testing methodology. I have enlisted a few of these.

- [White box testing](#)
- [Black box testing](#)
- Gray box testing
- Unit testing
- Integration testing
- Regression testing
- [Usability testing](#)
- Performance testing
- Scalability testing
- [Software stress testing](#)
- Recovery testing
- [Security testing](#)
- Conformance testing
- Smoke testing
- [Compatibility testing](#)
- System testing
- Alpha testing
- Beta testing

The software testing methods described above can be implemented in two ways - manually or by automation. Manual software testing is done by human software testers, who manually check the piece of code, test and report bugs in it. In case of automated software testing, the same process is performed by a computer by means of automated testing software such as WinRunner, LoadRunner, Test Director, etc.

Software Testing Methodologies

These are some of the commonly used test methodologies:

- Waterfall model
- V model
- Spiral model
- RUP
- Agile model
- RAD

Let us have a look at each of these methodologies one by one.

Waterfall Model

The waterfall model adopts a 'top down' approach regardless of whether it is being used for software development or testing.



The basic steps involved in this software testing methodology are as follows:

1. Requirement analysis
2. Test case design
3. Test case implementation
4. Testing, debugging and validating the code or product
5. Deployment and maintenance

In this methodology, you move on to the next step only after you have completed the present step. There is no scope for jumping backward or forward or performing two steps simultaneously. Also, the model follows a non-iterative approach. The main benefit of this methodology is its simplistic, systematic and orthodox approach. However, it has many shortcomings since bugs and errors in the code are not discovered until and unless the testing stage is reached. This can often lead to wastage of time, money and other valuable resources.

V Model

The V model gets its name from the fact that graphical representation of different test process activities involved in this methodology resemble the letter 'V'. Basic steps involved in this methodology are more or less the same as in waterfall model. However, this model follows both a 'top-down' as well as 'bottom-up' approach (you can visualize them forming the letter 'V'). The benefit of using this methodology is that, both the development and testing activities go hand-in-hand. For example, as the development team goes about its requirement analysis activities, the testing team simultaneously begins with its [acceptance testing](#) activities. By following this approach, time delays are minimized and optimum utilization of resources assured.

Spiral Model

As the name implies, spiral model follows an approach in which there are a number of cycles (or spirals) of all the sequential steps of the waterfall model. Once the initial cycle gets completed, a thorough analysis and review of the achieved product or output is performed. If it is not as per the specified requirements or expected standards, a second cycle follows, and so on. This methodology follows an iterative approach and is generally suited for large projects having complex and constantly changing requirements.

Rational Unified Process (RUP)

The RUP methodology is also similar to the spiral model in the sense that entire testing procedure is broken up into multiple cycles or processes. Each cycle consists of four phases namely; inception, elaboration, construction and transition. At the end of each cycle, the product/output is reviewed and a further cycle (made up of the same four phases) follows if necessary. Today, you will find certain organizations and companies adopting a slightly modified version of the RUP, which goes by the name, Enterprise Unified Process (EUP).

Agile Model

This methodology follows neither a purely sequential approach nor a purely iterative approach. It is a selective mix of both approaches in addition to quite a few and new developmental methods. Fast and incremental development is one of the key principles of this methodology. The focus is on obtaining quick, practical and visible outputs rather than merely following the theoretical processes. Continuous customer interaction and participation is an integral part of the entire development process.

Rapid Application Development (RAD)

The name says it all. In this case, the methodology adopts a rapid developmental approach by using the principle of component-based construction. After understanding the different requirements of the given project, a rapid prototype is prepared and is then compared with the expected set of output conditions and standards. The necessary changes and modifications are made following joint discussions with the customer or development team (in the context of software testing). Though this approach does have its share of advantages, it can be unsuitable if the project is large, complex and happens to be extremely dynamic in nature, wherein requirements change constantly.

This was a short overview of some commonly used software testing methodologies. With applications of information technology growing with every passing day, the importance of proper software testing has grown multifold.

By Tintin

Last Updated: 11/30/2011

Ads by Google

[Agile, Scrum, XP in India](#) Agile Training, Agile Team Coaching Agile Implementation Consulting, www.binaryessentials.com

[Learn Testing with Squish](#) for both testers and developers. Squish course, learn best practice www.kdab.com/training

[SOA Virtualization](#) Intelligent Virtual Services for SOA Testing www.grid-tools.com

[Website Designing](#) Get Website, Portal & Online Store 24X7 Support Call Now 8588074767 www.lbtpl.com

[ETL Testing - iQA](#) Complete ETL Testing and QA Tool to test your data warehouse www.icedq.com

[BA in 3D Animation & VFX](#) Get University Degree in 36 months Learn Animation & VFX from MAAC www.MAACindia.com

[Software Testing Courses](#) Find Top Training Centers in India. Get Info on Courses, Admission, Fees. www.Shiksha.com/Software-Testing

Like This Article?

22

Like

9

Follow: