

Preamble

This material is released under a Creative Commons License agreement <http://creativecommons.org/licenses/by/2.0/uk/>. In informal terms we, the authors, give you permission to do most things with the material provided you:

- a) Acknowledge us as the authors of the original material, and retain the acknowledgement in copy of the material, and in any of your material based on ours,
- b) Do not pretend you wrote it,
- c) Do identify any changes you make as your work, not ours.

The material includes opinions of the authors, and may be different from work published by others. We have quoted and referenced work from third parties, and in a few cases have knowingly used or adapted ideas or content from third parties. Where appropriate we ask for permission from the third parties.

However we make mistakes, as most people do, and may have made mistakes in this work. If you spot something you feel is wrong, or simply something you feel could be improved please let us know via this email address nft.introduction@commercetest.com

We are Stuart Reid and Julian Harty, and are the authors of this material: **An introductory course on non-functional software testing**. If you wish to create derived works you need include the following acknowledgement: at the start and end of the derived works: **This work is based on original material by Stuart Reid and Julian Harty**. We like to know whether our work is being used so please tell us how you're using the content.

Finally, this work is still in a draft form so you may find inconsistencies, gaps, incomplete content, etc. We hope you will find the content useful anyway and we are very happy to receive suggestions for improvement to the email address mentioned above.

Acknowledgements

Thank you to Isabel Evans of Testing Solutions Group who suggested a number of improvements and changes to this section.

Software Usability

Unusable or hard-to-use software makes for unhappy users. Usability specialists help people to understand and resolve usability issues. As with many non-functional areas, usability is best addressed in the design of software and other products. It is better to design usability in from the start rather than test the software and find expensive changes are needed.

Usability testing is performed by reviewing software and other product designs, by testing the product against usability requirements, and by measuring the usability of the product in use. Any issues discovered in the testing can then be addressed to make the software more usable, and the users will benefit as a result.

Accessibility testing is sometimes considered to be synonymous with usability testing, and they share similar goals. However accessibility focuses on special needs and requirements of less able users, while usability includes able and less able users.

Usability definitions

Here are some definitions on usability and usability testing you may find useful.

Usability definitions

ISO/IEC 9126-1 (2000)

“Usability: the capability of the software product to be understood, learned, used and attractive to the user, when used under specified conditions.”

“Quality in use: the capability of the software product to enable specified users to achieve their goals with effectiveness, productivity, safety and satisfaction in specified contexts of use.”

Usability testing definition

*“Usability testing shall use a model of the software/system that specifies its usability requirements including any required usability design standards to which the software/system must conform. Usability requirements shall specify the usability goals for the system. Usability goals will be based on **system goals** (the reason for having the system, the difference it is to make for the organisation or individual, its purpose and the tasks it will aid), and the **contexts of use** for the software/system (who is to use the system and the environment in which it is to be used, user characteristics and user tasks).”*

<http://www.testingstandards.co.uk/usability.htm>

What is unique about usability testing?

Usability testing is all about how well, or otherwise, users can use software to perform tasks. The testing needs to consider both the tasks and the context of use: for example how adept the users are with computers, the environment within which they will use the software, and the physical and mental demands the product makes on the user.

Comparing usability testing with...

Functional testing

Functional testing concentrates on whether the software performs the tasks correctly, it doesn't test how well the user is able to use the software to perform the tasks. So even if the user had to do 1000 things to complete a task, if the task completes and generates the expected result it would have met its functional requirement, however the software may fail to meet its usability requirements.

See [Jakob Nielsen 2001a] for a great explanation of what usability is all about.

Other non-functional testing

Usability intersects with many non-functional aspects of software e.g.:

- Installability – Can the software be installed correctly?
- Usability – Can this user install this software in this environment without having to retrace their steps in the installation process when the user is not familiar with the installation mechanism?

- Security – Only authorised users can access the updates section of the products' web site.
- Usability – Can this valid user authenticate themselves to reach the updates section of the products' web site within 5 keystrokes or 5 mouse-clicks?

We would need to create different and distinct test cases to test the usability aspects of these non-functional requirements.

Special stakeholders

The end users have a particular interest in whether the software is usable *for them*.

Among the operations and support groups, the support or help desk will have an interest in the usability of the software – the better the usability, the fewer help desk calls will be required.

The customer will also have an interest in usability: good usability includes the effectiveness and efficiency of task completion with which users can complete their tasks. If the effectiveness and efficiency of task completion is high that gives a greater throughput of work done correctly, first time, which has positive commercial implications.

Via accessibility, many countries have legislation that applies to usability of software. So if a case can be made that a disabled user was discriminated against owing to poor usability then whoever is responsible for the software could be in trouble.

Benefits and risks associated with usability testing

The business or corporate body (the sponsors) that decided the software was necessary has a vested interest in the software being usable for their body of users, as the users are likely to be more productive and content if the software makes it easy, fast and safe for the users to complete their tasks. Therefore the sponsors can see the benefits of usability testing if it is able to identify usability problems that compromise how quickly, easily and safely the users can perform their tasks.

There are both benefits and risks associated with usability testing that involves actual users.

Benefits include:

- Direct and indirect feedback to the other stakeholders e.g. the development and design teams, on what works and doesn't work for these users
- Users feeling they matter as they are being consulted and involved in the process

Risks include:

- Upsetting the users by ignoring, or seeming to ignore, their complaints and feedback
- Spending too much time trying to address issues raised in the feedback where the issues are unrepresentative, unrealistic, misleading, contradictory, or ones that would disappear over time

E.g. a new user-interface for a replacement software product may initially take longer for users to learn while they adapt to the new behaviour; however once the users have adapted they may be more productive and even happier than they were with the old software...

Usability testing, especially when it includes accessibility testing, can help demonstrate that those responsible for the software met their legal requirements for providing a usable and accessible product or service.

Costs of getting it wrong

Here is a good example of the costs of getting usability wrong:

On IBM's Web site, the most popular feature was the search function, "because people couldn't figure out how to navigate the site," said Carol Moore, IBM's vice president for Internet operations. The second most popular feature was the "help" button, because the search technology was so ineffective.

IBM's solution was a 10-week effort to redesign the site, which involved more than 100 employees at a cost estimated "in the millions."

The result: In the first week after the redesign, use of the "help" button decreased 84 percent, while sales increased 400 percent.

Good Web Site Design Can Lead to Healthy Sales - Bob Tedeschi - 1999
<http://bmrc.berkeley.edu/courseware/cs160/fall99/readings/E-Commerce%20Report%20good-web-design.htm>

Potential conflicts

With functionality

When we need to decide between functionality and usability there is a trade-off as both compete for a limited set of resources such as time, developers, etc. However if we're able to take a slightly different viewpoint and consider how to implement requirements so they also meet usability requirements the conflicting demands are likely to reduce since it's generally possible to implement functionality so that it is usable (and accessible).

The earlier in the SDLC one considers usability, and other non-functional attributes, the more likely are the chances of delivering usable functionality.

Functionality that's not usable could be considered as a waste of time and resources to implement.

With other non-functional attributes

Usability and performance may conflict, especially when more computing resources are needed to deliver usability e.g. Windows Explorer provides a richer user-interface and more information than command line utilities when performing activities such as copying files, creating folders, etc. However the performance suffers significantly (which is one reason why technophiles continue to use the command-line utilities). Potentially, if the user-interface was very rich, with lots of options, usability could decrease as users might get confused, or lost in the user-interface.

Both the performance and security sections include comments on the potential conflicts between usability and performance and security respectively.

Example of specifying usability requirements

The following example is based on a fictional product that requires users to register before they can perform other tasks. The product will be used by at least 100 users during the beta test phase.

The usability requirements need to ensure different types of users can use the system successfully (e.g. novice and experienced). In order to test the usability of the interface, you need to ensure there are suitable requirements, and to create test cases where novice users and experienced users are separately asked to perform typical activities.

Also there should be end-to-end requirements for these users to perform different types of transactions. Note that such requirements are related to performance requirements – so be careful they do not conflict. Performance will be used to measure the response times of the software, whereas the usability tests might measure how quickly the users can complete a task. The performance of the interface will affect usability if the system does not respond sufficiently quickly. And if the performance is so poor that the user gets distracted or disheartened the usability will be perceived as even poorer.

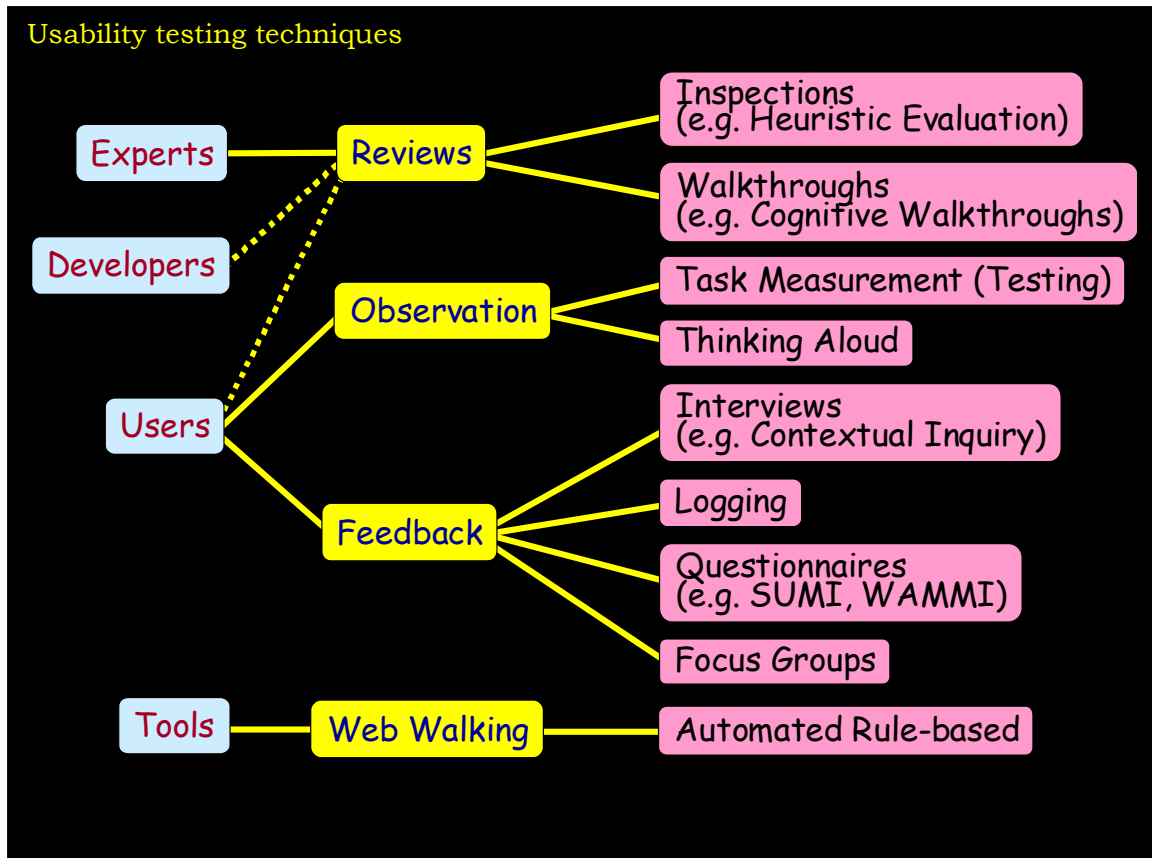
We have provided one example of a typical usability requirement, in practice there are likely to be significantly more.

Usability

ID	NFR.13
Description	Users must be able to complete the registration successfully first time. (I.e. without cancelling and restarting the process.)
User Requirement	UR.1
Primary Attribute	Usability.
Measure	Task measurement, based on software audit logs and interviews with selected users.
Statutory Requirement	None
Basic Requirement	The ratio of completed registration / started registration is ≥ 0.95 with a sample of > 40 new users. At least 80% of sample users agree the registration process was clear and easy to complete, first time. Less than 5 percent disagree.
Ideal Requirement	$\geq 95\%$ of sample users agree the registration process was clear and easy to complete. None disagrees with the statement.
Current value	Not applicable.
Technique	Analysis of the application's audit log, subtract the number of 'completed registration' entries with those for 'started registration'. User Interviews after a range of able and visually impaired users are using the software for the first time.
Test Requirement	Users are drawn from the body of intended users within the organisation. The software will be installed on representative hardware (specification provided separately). Users will be interviewed within one day of completing the registration process.
Operational Requirement	Continuing measurement based on feedback from users.
Conflicts	None.
Priority	6 (on range 1-10 for this project).

Approaches for usability testing

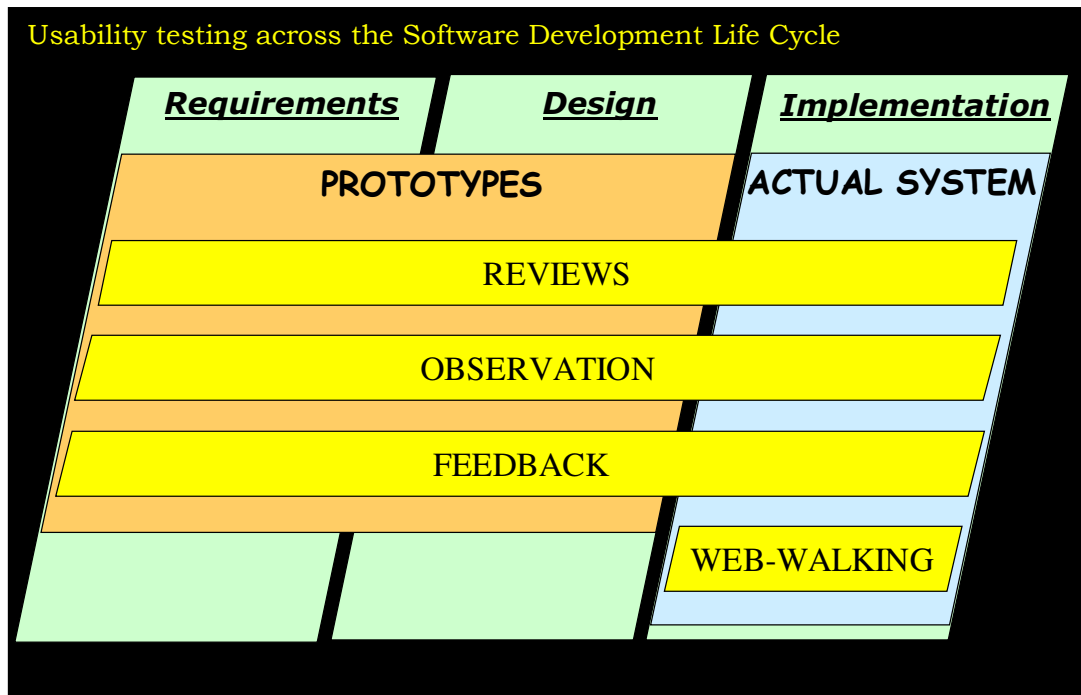
There are at least nine categories of usability testing techniques, which can be grouped into one of four groups – as shown in the following figure.



Users are directly involved in six of these nine categories. Usability testers may be involved in any or all of the techniques. They may also be 'experts' and able to perform, or lead, the reviews.

Usability testing across the SDLC

Usability testing can be used throughout the SDLC, from when the requirements are being gathered until after the software has been deployed. The following figure shows how broadly throughout the SDLC most of the groups of usability testing techniques can be used.



Where is usability testing performed?

Locations range widely, from custom laboratories with one-way mirrors and sophisticated monitoring devices to web-based forms available on the Internet, to field trials where users are interviewed where they work. Some techniques involve, or require, the software to be available for the users to use. If so, test environments, or even the live systems may be used.

Web-walking generally takes place in a test environment to avoid the tool affecting the production systems e.g. by generating orders while navigating through a web site.

What special resources are required?

Specialist resources range from a notebook and pencil (wielded by a usability expert) through to specialised, dedicated usability laboratories.

Web-walking tends to be automated using software tools. These software tools range from free and available online, through to software packages installed on a test machine.

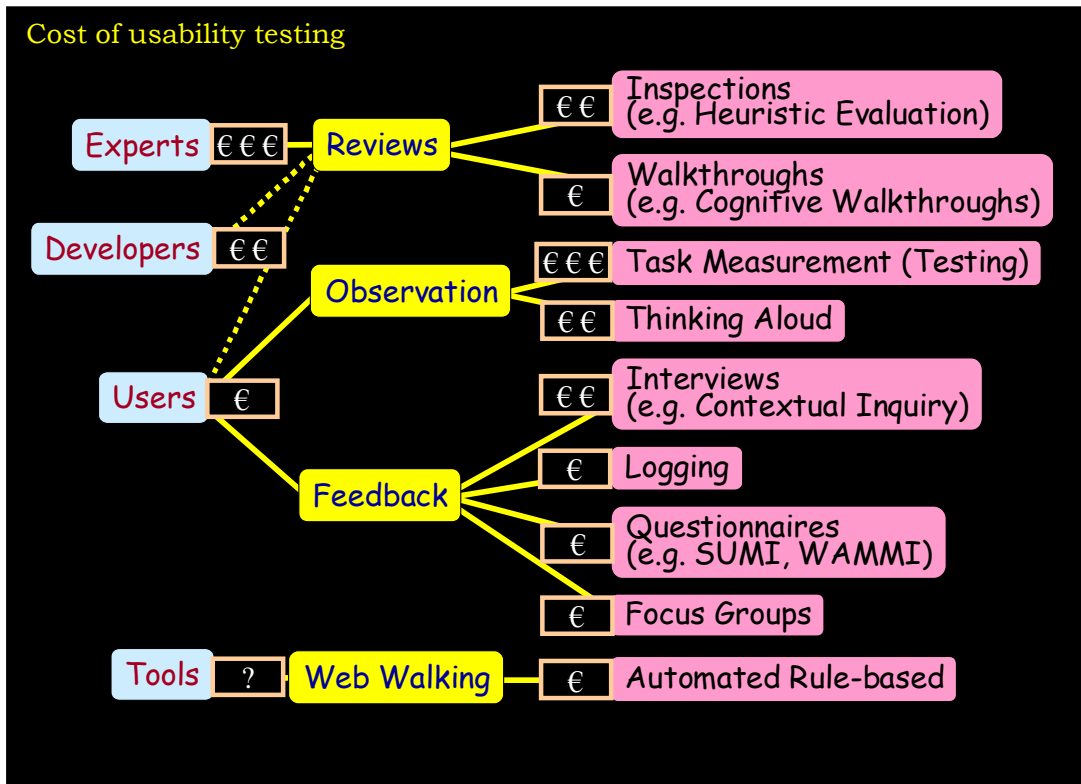
Costs of usability testing

The costs of usability testing depends on a number of factors, including:

- Who / what performs the testing
- The number of people involved
- The testing technique(s) used
- License, or service, fees e.g. for SUMI assessments

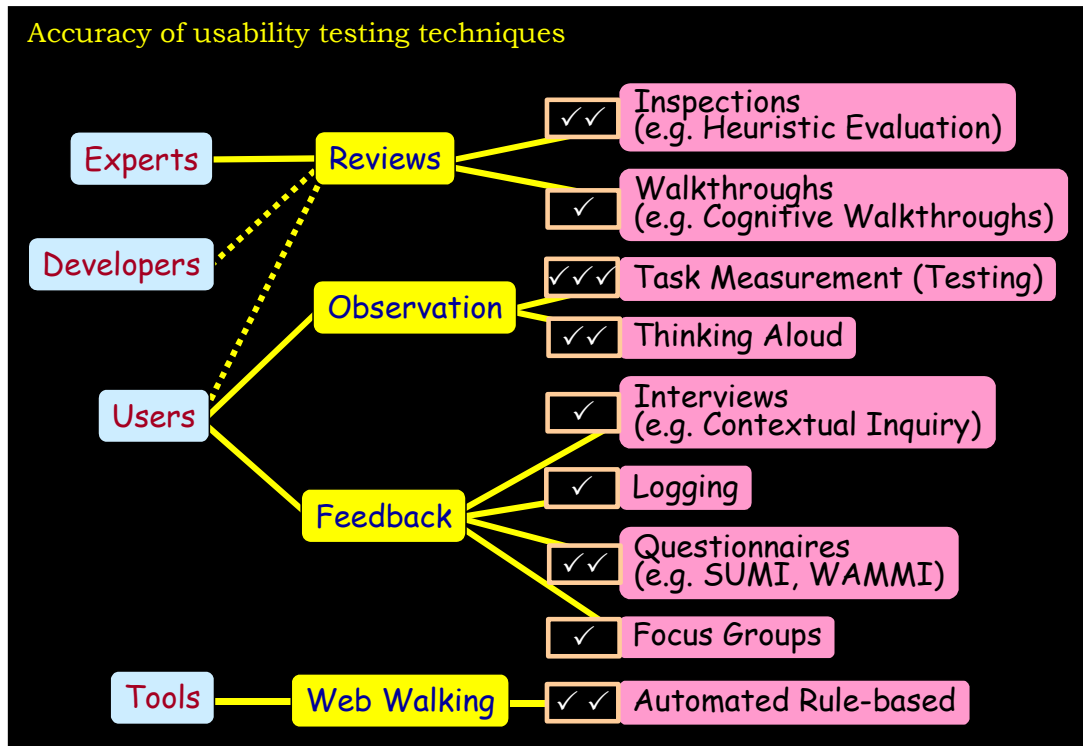
- Cost of software required e.g. for web walking or to collate the results

The visible costs e.g. for tools, license fees, etc. tend to be relatively low. The bulk of the costs are for people and the time they spend involved in the testing. The following figure gives an indication of the comparative costs involved for people and techniques.



Accuracy of usability testing techniques

However we shouldn't consider costs in isolation, rather we need to understand how accurate each technique is so we can look at the return on investment of each technique. The following figure provides comparative ratings of the accuracy.



References

[Jakob Nielsen 2001a]

"Usability is not just a matter of whether or not it is possible for a user to perform a task. It is also a matter of how easy and fast it is for them to do so."

Jakob Nielsen, 2001

www.useit.com

References from books and other printed material

Don't make me think

Steve Krug

New Riders © 2000

ISBN 0-7897-2310-7

A small, easy-to-read book that describes how to provide usable web sites.

Includes practical suggestions of how to test usability quickly and inexpensively.

General references on usability

www.ucc.ie/hfrg - Human factors research group, with links to their work on SUMI, etc

www.wammi.com - Home page for WAMMI usability testing technique

www.testingstandards.co.uk - Information on a number of aspects related to non-functional software testing.

www.usability.serco.com/trump - Good source of information on usability

<http://www.microsoft.com/usability/> - Home page for Microsoft material on usability

General references on accessibility

<http://www.w3.org/WAI/> - The Web Accessibility Initiative, the origin and template for most legislation related to accessibility of web sites.

<http://bobby.watchfire.com/bobby/html/en/index.jsp> - A commercial tool, that also provides limited facilities free-of-charge. One of the well-recognised accessibility tools.

<http://www.sidar.org/hera/index.php.es> - Home page for a free-to-use web site that can check other web sites for conformance with the W3C WAI (see above)

<http://www.bda-dyslexia.org.uk> - Uses the 'ReadSpeaker' utility to read web sites on demand. Useful for testing how a web site 'sounds' to a user e.g. to someone who is unable to read the text directly.

-- End of document --