

Seeing What Isn't There - The Use of Heuristics

By Stewart Kidd, Software Tester

July 29, 2014

Have you ever been about to document a defect discovered in an application under test and wondered how to best compose your words? You want to make sure the defect is clear to the stakeholders, making sure it gets the attention it deserves. If you, like me, sometimes have trouble articulating your thoughts and being persuasive, then a useful tool is “heuristics”. An example of a heuristic is “Consistent with the product’s history”, meaning the present version of the product is consistent with past versions of itself. Derived from the Greek “heuriskien” that means “find or discover”, heuristics are best suited for situations where the issues are not black or white, but occurring in the implicit areas that are somewhere in-between, and where there can be a definite lack of specifications. As Michael Bolton pointed out in “Testing Without a Map”, ““Completeness” is entirely dependent upon perspective and context. Even so-called “complete” specifications contain much that is implicit. Some specifications are not supplied in formal documents, but come to you through e-mails, conversations, or through your own inferences.”¹

**“best suited for situations
where the issues are not
black or white”**

The dictionary definition of a heuristic, or “rule of thumb”, is a guideline serving to indicate or point out and encourage a person to learn, discover, understand or solve problems on his or her own, as by experimenting, evaluating possible answers or solutions. They can add weight to your findings and really turn a defect report into a persuasive document. As the Association for Software Testing² website points out, “The key goal of the bug report author is to provide high-quality, well-written, information to help stakeholders make wise decisions about which bugs to fix when.” You are doing all this to make sure that the bug reports do not wind up being just neutral technical reports.

**“They can add weight to
your findings”**

There are several heuristics that are applicable to testing any kind of product, whether it is software or a service. A heuristic is simply a guideline used to determine whether a given test may pass or fail.

Since heuristics are a tool, they do not come with a guarantee that they will give you the right answer. Sometimes, certain heuristics can contradict other valid heuristics. They can only point you to a potential problem and, in doing so, aid in making a decision. They are not comprehensive. Heuristics help us recognize

problems but they don't help us solve them; they are something to consider. There are plenty of other ways to decide whether a product is acceptable or not.

The original list of heuristics that testers are familiar with comes from James Bach³. To easily remember the heuristics, James came up with the mnemonic "HICCUPSF" which stands for:

- **Consistent with the product's history** - The present version of the product is consistent with past versions of itself, meaning a product's features and functionality should be consistent with its past behaviour.
- **Consistent with the products image** - The product is consistent with the image its makers want to project to its customers or user. This is also known as "branding". Customers can build strong emotional attachments to products so the experience should be seamless from version to version.
- **Consistent with comparable products** - The product is consistent with a comparable one; i.e. its closest competitors. You want to have a rich feature set that is equal to or, ideally, better than your competitors.
- **Consistent with claims** - The product must behave the way the marketing team claims it will. These claims can be made through literature, specifications, help files and conversations or emails.
- **Consistent with user's expectations** - Is the product consistent with what we think the user wants? What they can reasonably expect?
- **Consistent with purpose** - This would include both explicit⁴ (precisely and clearly expressed) and implicit⁵ (suggested though not directly expressed) purposes of the product. Microsoft Word offers a rich set of formatting features. Notepad does not. The two applications serve different purposes which must be kept in mind while testing.
- **Consistent within product** - Each feature of the product is consistent with comparable features in the same product i.e. 'look and feel' is consistent.
- **Consistent with statutes, regulations and binding specifications** - Does the product abide by applicable laws and statutes? Does it comply with legal requirements and restrictions? "These differ in that they are imposed on developers by outside organizations."⁶
- **Consistent with familiar problems** - Does a problem from an earlier version of the product still exist? Has it been deferred because it is irrelevant, obscure, or has it been mistakenly considered as having no customer impact?

I can remember testing changes made to a major financial institution's website and finding some glaring differences between the English and French versions of the same site. The development of the English site had been contracted to one company, while the development of the French site had been contracted to another. This had obviously been done to make sure the content was accurate in both sites and official languages, for users to

better understand financial matters that concerned them. Being developed independently, functionality was slightly different, embedded links were in different places, etc. When I documented these defects and submitted them for review, I argued that there was no consistency within the product which could adversely affect the user's perception of the company. As the major stakeholders agreed with my conclusions, the defects were upgraded to critical and fixed before the site changes went live. Consistency within the product was important, especially in a bilingual country.

Using these heuristics, testers can see not only what is there, but also what might be missing. Something expected that is missing might threaten, or at the very least be detrimental to, the value of the product. To summarize, heuristics are invaluable tools that help testers provide developers, stakeholders and everyone else with as much well written information as possible to make informed decisions about which defects to fix. They have been in use for as long as there have been products to test which is a testament (pardon the pun) to their relevance.

¹<http://www.developsense.com/articles/2005-01-TestingWithoutAMap.pdf>

²<http://www.associationforsoftwaretesting.org>

³http://en.wikipedia.org/wiki/James_Marcus_Bach

⁴<http://www.oxforddictionaries.com/definition/english/explicit?q=explicit>

⁵<http://www.oxforddictionaries.com/definition/english/implicit?q=implicit>

⁶<http://www.testingeducation.org/BBST/bugadvocacy/BugAdvocacy2008D.wmv>



About The Author

Stewart Kidd is a Software Tester with Professional Quality Assurance with 5 years of experience in the IT industry, all of it spent in software testing. His experience includes functional and performance testing, test planning, test management and regression testing. Stewart has worked in a variety of industry verticals, including: forestry, healthcare, insurance, banking, gaming, transportation, security intelligence and public utilities. Stewart has been with PQA for the past 2 years and is always looking forward to his next testing challenge.