



Unlearning for testers

Today I want to present a non-technical advice: unlearning.

The traditional definition* of unlearning stands for giving up present habits and/or knowledge:

- verb** (used with object)
 1. to forget or lose knowledge of.
 2. to discard or put aside certain knowledge as being false or binding: to unlearn preconceptions.
- verb** (used without object)
 3. to lose or discard knowledge.

* -Taken from Dictionary.com

As we can see from this definition, there are negative and positive sides of unlearning.

How to benefit from unlearning?

- o Use it as a brainstorming tool
- o Use it as a problem-solving tool
- o Use it for retrospective analysis and learning

As an example, below I put a few core assumptions about software development and testing that have already been unlearned or currently being unlearned in the industry.

- o One needs requirements and test cases to do testing
- o Testing can only be conducted after a build is deployed on test environment
- o Verification tests passed -> no bugs
- o A role of written documentation is critically important in software quality assurance
- o Software quality can be unambiguously defined by requirements
- o No bugs found -> no bugs in software
- o No bugs in software -> customer satisfaction
- o Programmers do programming, testers do testing
- o Years of experience with tools, or techniques, or technologies is the most important hiring criteria

Filed under: [Heuristics](#)

[Leave a comment](#)

Comments (0)

Trackbacks (0)

(subscribe to comments on this post)

No comments yet.

Leave a comment

Name (required)

Mail (will not be published) (required)

Website

[« SQL Injection cartoon and walkthrough](#)

[Crash test your application »](#)

Categories

- [Accessibility](#) (2)
- [Automation](#) (17)
- [Availability](#) (1)
- [Bias](#) (7)
- [Bug Reports](#) (18)
- [Career Tips](#) (6)
- [Compliance](#) (1)
- [Database](#) (3)
- [Disaster Recovery](#) (2)
- [Documentation](#) (14)
- [Estimation](#) (1)
- [Exploratory Testing](#) (24)
- [Free Tools](#) (68)
- [Heuristics](#) (47)
- [Internationalization](#) (4)
- [Learning about the product](#) (7)
- [Mind Mapping](#) (9)
- [Performance Testing](#) (33)
- [Playing Well With Others](#) (39)
- [Practicing Testing](#) (8)
- [Regression Testing](#) (2)
- [Security Testing](#) (13)
- [SEO](#) (1)
- [Skilled Bug Investigation](#) (13)
- [Stress Testing](#) (1)
- [Technical Tricks](#) (1)
- [Test Data](#) (15)
- [Test Management](#) (33)
- [Test Oracles](#) (2)
- [Test Planning](#) (31)
- [Test Theory](#) (4)
- [Testability](#) (1)
- [Testing mobile](#) (6)
- [Testing Techniques](#) (14)
- [Time Savers](#) (27)
- [Tools under \\$100](#) (9)
- [Uncategorized](#) (2)
- [Unit Testing](#) (2)
- [Usability](#) (12)
- [Web Testing](#) (27)

Authors

- [Albert Gareev's website](#)
- [Anne-Marie Charrett's blog](#)
- [Jonathan Kohl's blog](#)
- [Julian Harty's website](#)
- [Karen Johnson's blog](#)

Pages

[About](#)