

## Eras of Software Testing

### Eras of Software Testing

Software testing has evolved and improved over the past 75 years, and has seen many changes as a result of changes in technologies, processes and perspectives on testing. Presented below is a table outlining some of the key eras of testing based on David Gelperin and Bill Hetzel's paper "The Growth of Software Testing" published in 1988. I've added in the last three eras myself, based on various textbooks.

| Era  | Description  |
|--|--|
| 1945-1956<br><b>Debugging-Oriented</b>       | This was at the start of the history of programming, and is sometimes called the "Code-and-Fix" era, where there was no testing process, programmers fixed code as they found errors.  |
| 1957-1978<br><b>Demonstration-Oriented</b>   | This was the first time there was a clear distinction between testing and debugging; and the testing focussed on ensuring that the program was doing everything it was supposed to do.   |
| 1979-1982<br><b>Destruction-Oriented</b>     | During this era, the goal was to see what inputs would cause the programs to fail, for example. if we put a text value in a numerical field, or we put in a date of birth after today's date.  |
| 1983-1987<br><b>Evaluation-Oriented</b>      | This era focused on testing as part of a larger quality assurance process; where it was acknowledged that large software systems would inevitably have some bugs in them, but to minimise the number of bugs to a specified rate.                            |
| 1988-2000<br><b>Prevention-Oriented</b>      | In this era, testers were expected to have a very good understanding of the systems that they were testing, and to know which parts of the code would be more difficult to test.   |
| 2001-2003<br><b>Methodology-Oriented</b>     | Testing gained a new prominence and importance in this era with the advent of software development methodologies that put testing at their core, including Test-Driven Development.  |
| 2004-2013<br><b>Automation-Oriented</b>      | In this era, large software testing tools were developed to help the testers do their job by eliminating some of the repetitive tasks, as well as creating large sets of input data, and inputting that data, and checking that the outputs are as expected. |
| 2014-To Date<br><b>Intelligence-Oriented</b> | Finally, we are now in an era where the testing tools are augmented by artificial intelligence that can help the tester figure out what tests are best for each part of the software.  |

Note: These dates are all approximate, and, in reality, these eras don't fit into tidy little boxes, so in practice the eras overlapped significantly, but for the sake of understanding the key evolutions in software testing, this is a really good model.

#PythonMonday © Damian Gordon