Some Principles of Testing

Fundamental Principle of Testing

- The computer scientist Edsger Dijkstra once famously said that "*Program testing can be used to show the presence of bugs, but never to show their absence!*" (EWD249), in other words checking that a program works with various inputs and under different conditions might uncover some errors (bugs) in a program, but because we can't check every possible input and circumstances, it's not possible to prove that a program will work in all scenarios using testing alone.

Other Principles of Testing

- *Exhaustive testing is not possible, but optimal testing is necessary*: As noted in the first principle it is not possible to test every possible input and circumstances, therefore we have to prioritise the parts of the program we think are most likely to cause errors and also prioritise the parts of the program that would be most serious (or risky) if they caused an error.
- *Bugs like to hang out with other bugs*: This is sometimes called "Defect Clustering" in other words if we find an error (bug) in part of a large program, we should have a look at the code in that region of the program, because there might be further errors near the original. This is like the Pareto Principle, 80% of the errors are found in 20% of the program.
- *We have to regularly change our approach to testing*: If we use the same tests on all of our programs, they will only locate certain kinds of errors. We need to change our approach for different programs even if they have a similar function, so we can add new tests onto our existing ones, revising existing tests, or just change the whole testing process.
- *Testing isn't just about checking if the code works*: The sooner testing begins, the sooner errors can be located, we should not wait for the programs to be written to start our testing process, we can test the design of our programs before the coding. We should test all of our assumptions, as well as the code.

BIOGRAPHY: Edsger W. Dijkstra

Dijkstra was born in Rotterdam on 11th May 1930 and died in Nuenen on 6th August 2002. He is one of the most influential and important people in the history of computer science. His contributions cover areas including compiler design, operating systems, distributed systems, program design, program verification, software engineering, graph algorithms, and the philosophical foundations of computer science. His testing paper, "*On the Reliability of Programs*", (EWD303) is considered a classic in the field of testing.