Difficulties in Software Programming

- Complexity: The bigger a program the more complex it is
- Change: Changes (New features, bug fixes, new interfaces) erode structure of system
- Competing Objectives: Correctness, performance, maintainability, usability, security, backward compatibility, rubustness, understandability, verifiability, reusability, portability, etc.
- Constraints: Money, time, resource limitations

Functional Testing Focused on input/output behaviour of method. Test if function returns correct known output for certain input. Include corner cases.

Structural Testing Look at source code to determine test cases that cover most of it.

Automatic Test Case Generation Automatically generate test cases for every possible path the program can take.

Static Program Analysis Compute all possible program executions and use mathematical reasoning.

Requirement Elicitation

Requirements

Describe the **users view**, the *what*, not the how.

Functional:

- Functionality: in->out mapping, abnormal situation handling, validity checks
- External interfaces (GUI, ICP, interfaces, hardware communication): Valid range/accuracy/tolerance check, units of measure, screen/data/command formats, description of purpose.

Nonfunctional:

- Performance
- Attributes: portability, correctness, etc.
- Design constraints: Required standards, environment