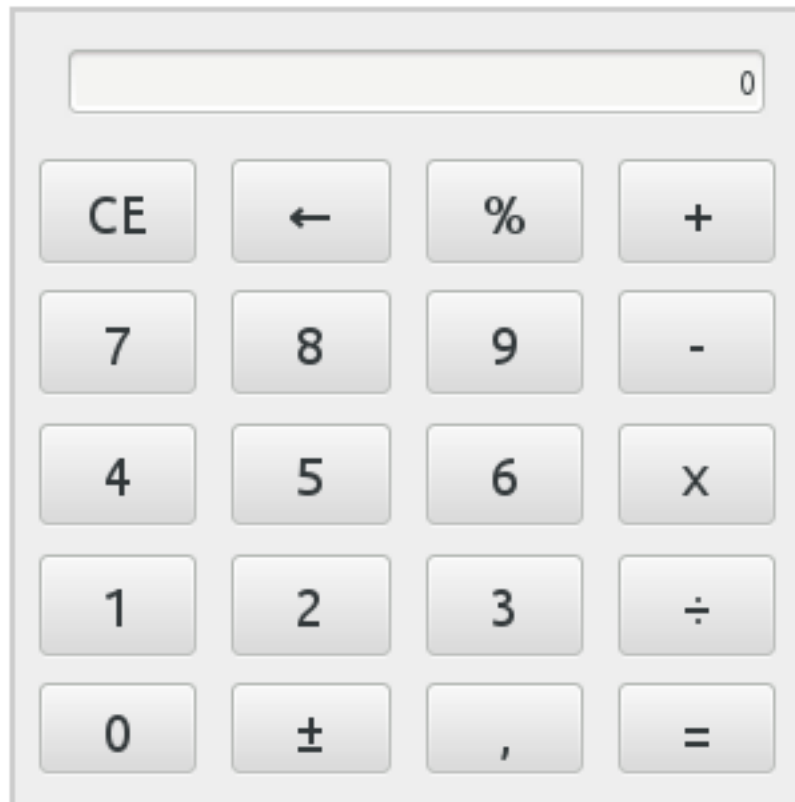


Introduction: Roles of Testers

01219343 Software Testing
Spring Semester 2013

Starter 1

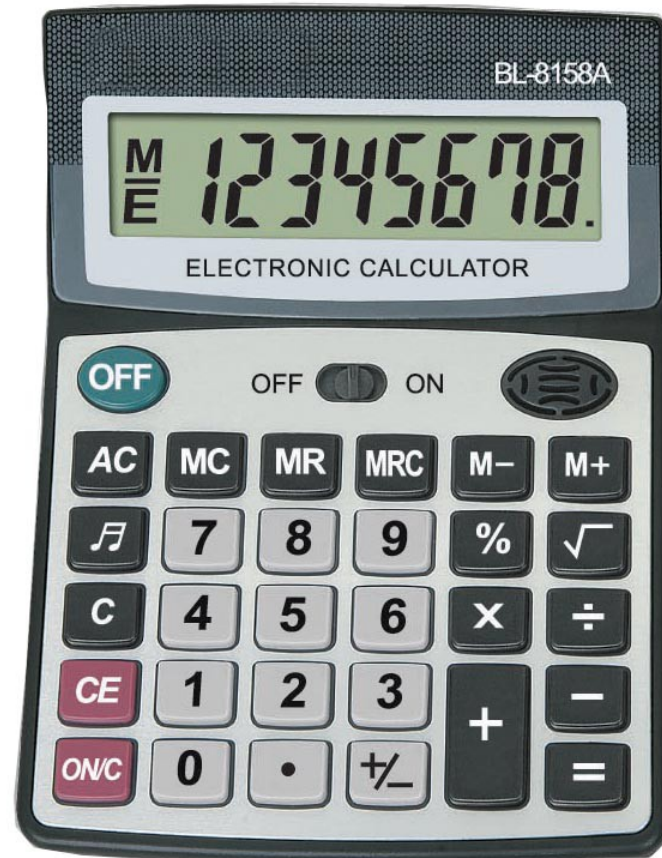
- Test this javascript calculator:



Source: http://www.webestools.com/scripts_tutorials-code-source-8-javascript-calculator-buttons-calculator-keyboard-support-operations-modulo.html

Starter 2

- Test this calculator:



Question

- Why do you test?

Question

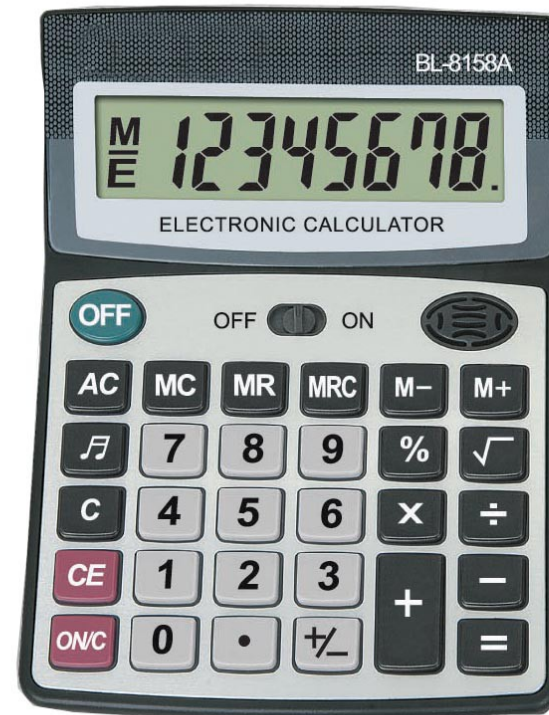
- Given you only one day, what would you do to test these calculator?

Goals

- There are so many "things" to test.
 - "things" → properties of the software/component/etc.
- Can you name a few of these properties?
 - *hint: -ility, -y*

Try again

- How can you test this calculator for:
 - Functionality
 - Reliability
 - Usability
 - Efficiency

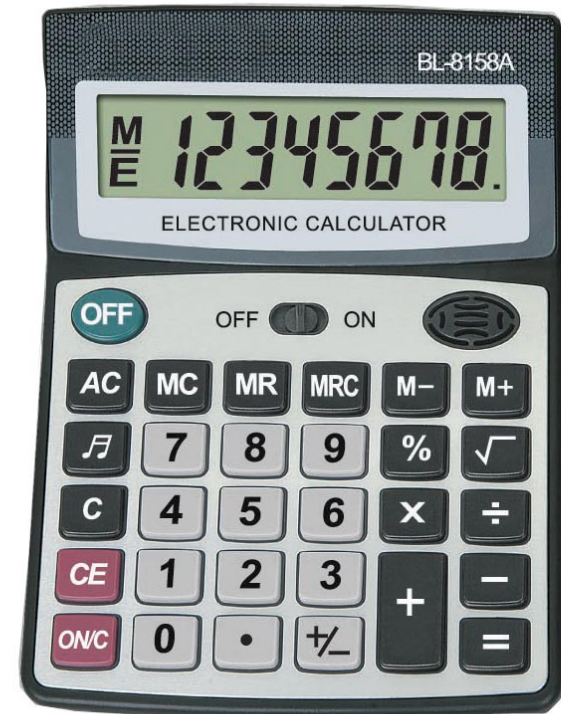


Test cases

- What should you specify in a test case so that anyone can execute and be confident about the required quality of the software?
- A test case should include:
 - step to perform
 - how to check if the software does the thing as you expected
- It should have the "goal", i.e., what the test case intends to check.

Practice

- Go back and refine your test cases.
- For each test case, specify:
 - Goal
 - Steps to perform
 - Expected result

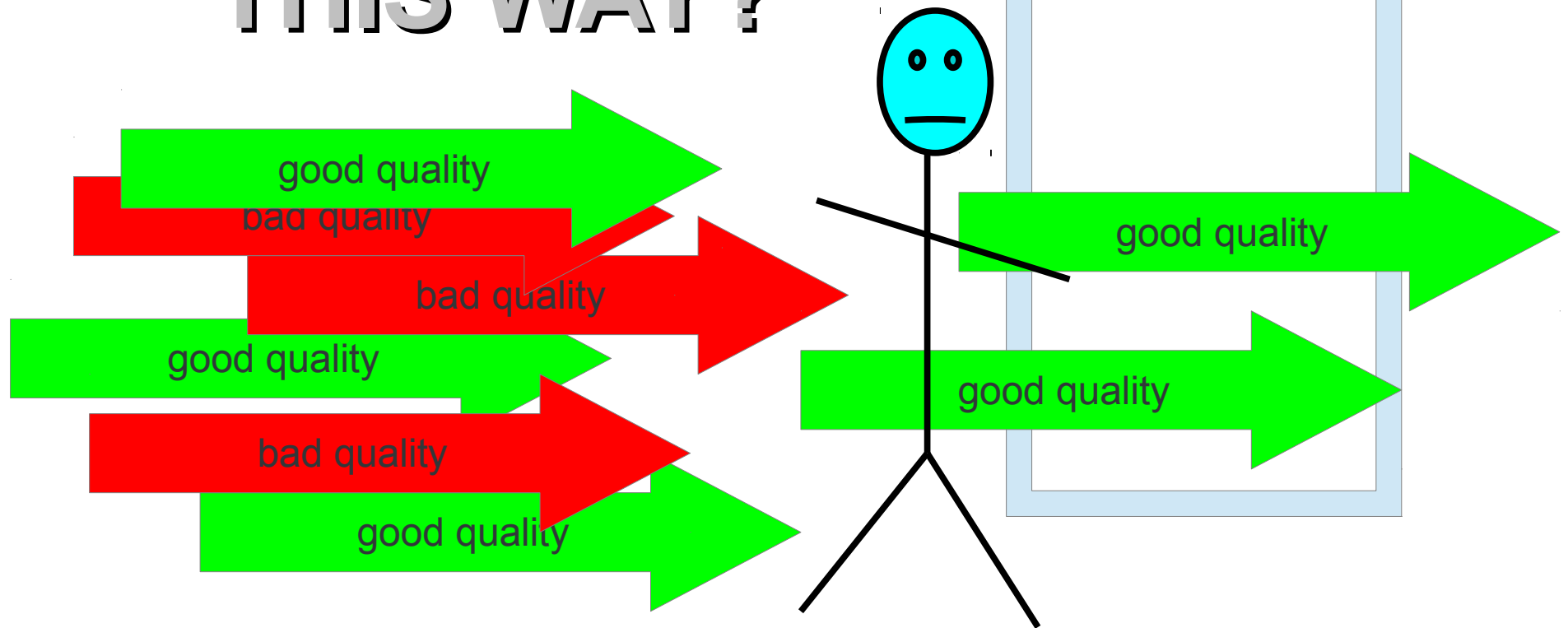


Tester and quality

- Do you think that tester's role is related to software quality? How?

Testers as the gate keeper

CAN YOU BUILD QUALITY SOFTWARE THIS WAY?



Quality

- You can't improve software quality by testing at the end.

Quality must be baked in.

How?

How you build software...

- usually determines the quality of the software you build.

Discussion

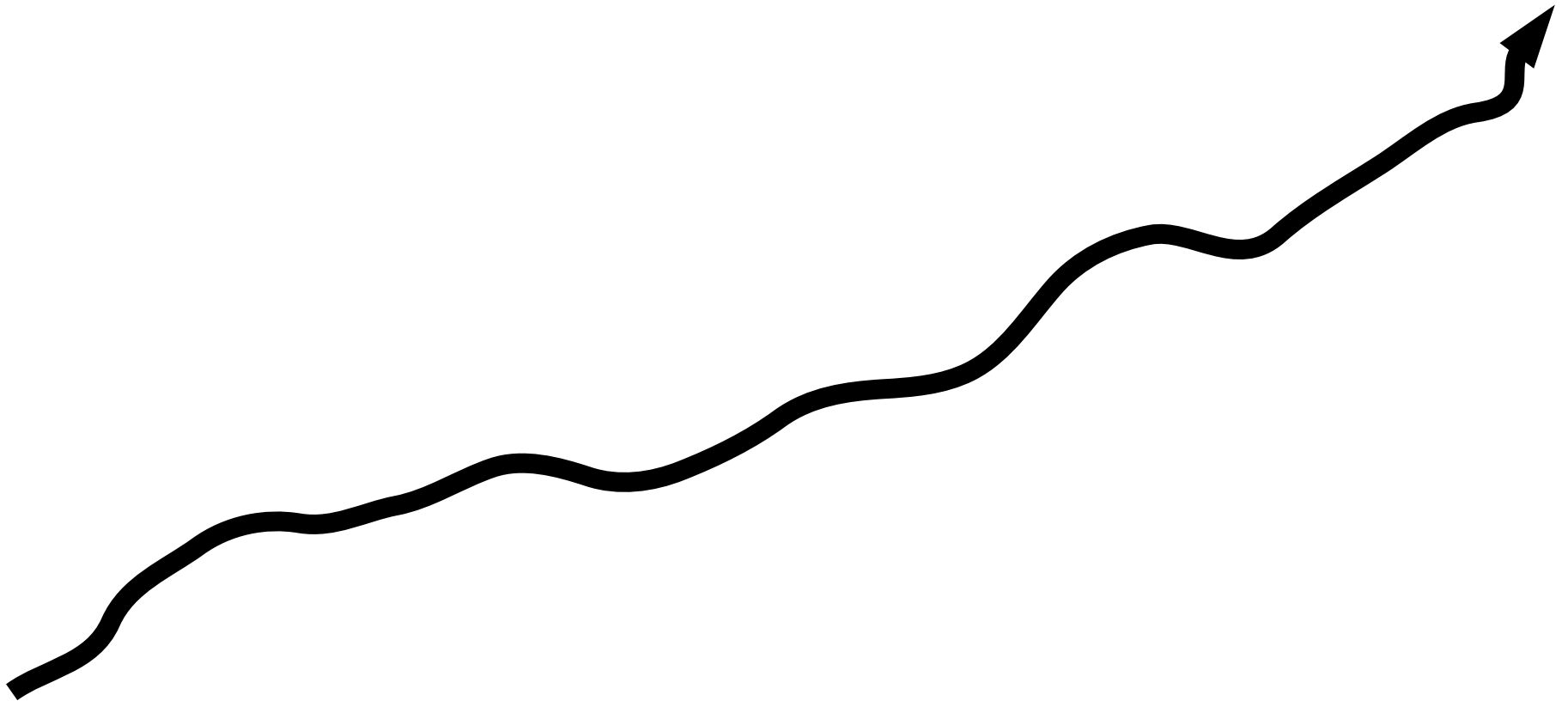
- Explain what you know about how to build software.

Discussion : Testing Experience

- Have you tested your software?
- When? (through out? / at the end? / never?)
- Do you think that you did enough testing?

Software Development as a Journey

- You have (unclear) goals, with limited resource.

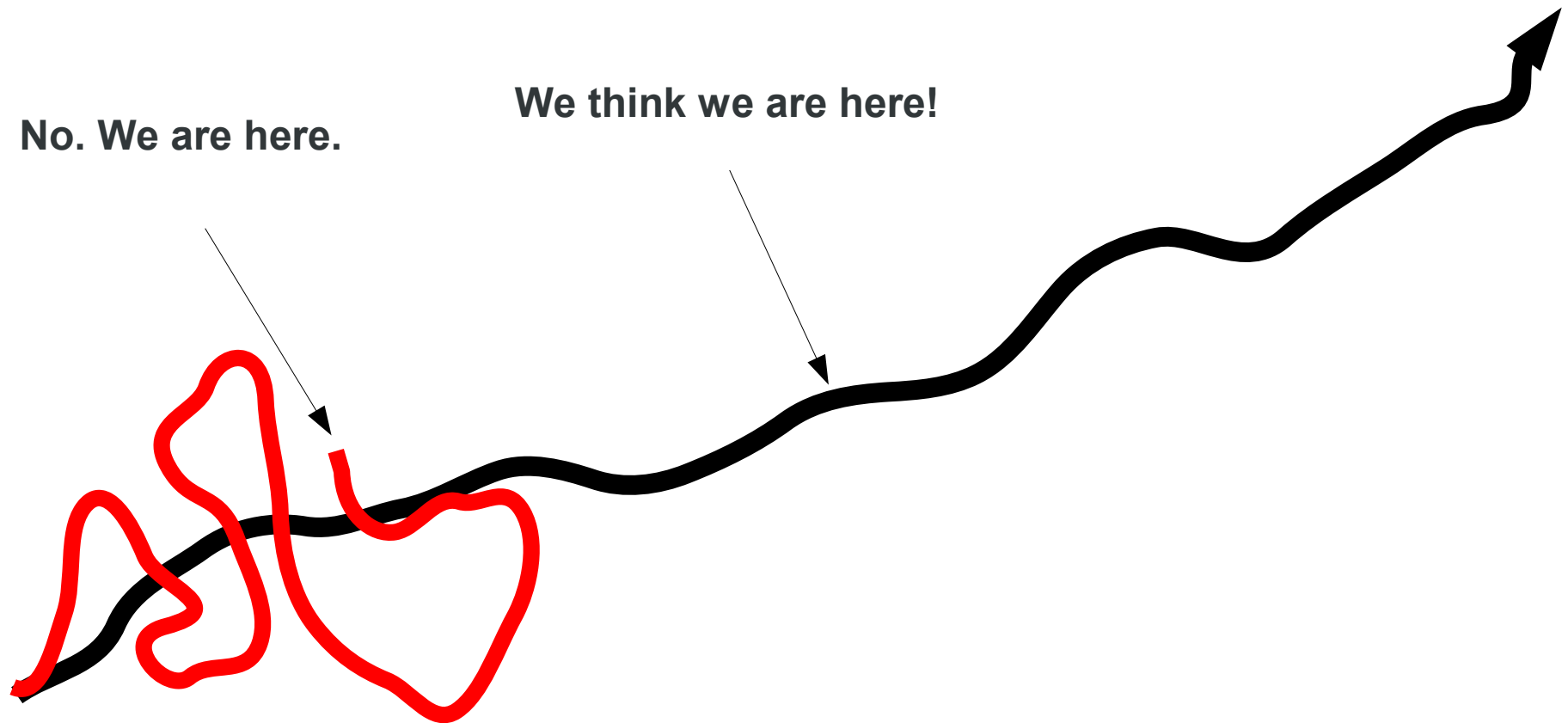


Lesson 1

You are the
headlights of
the project.

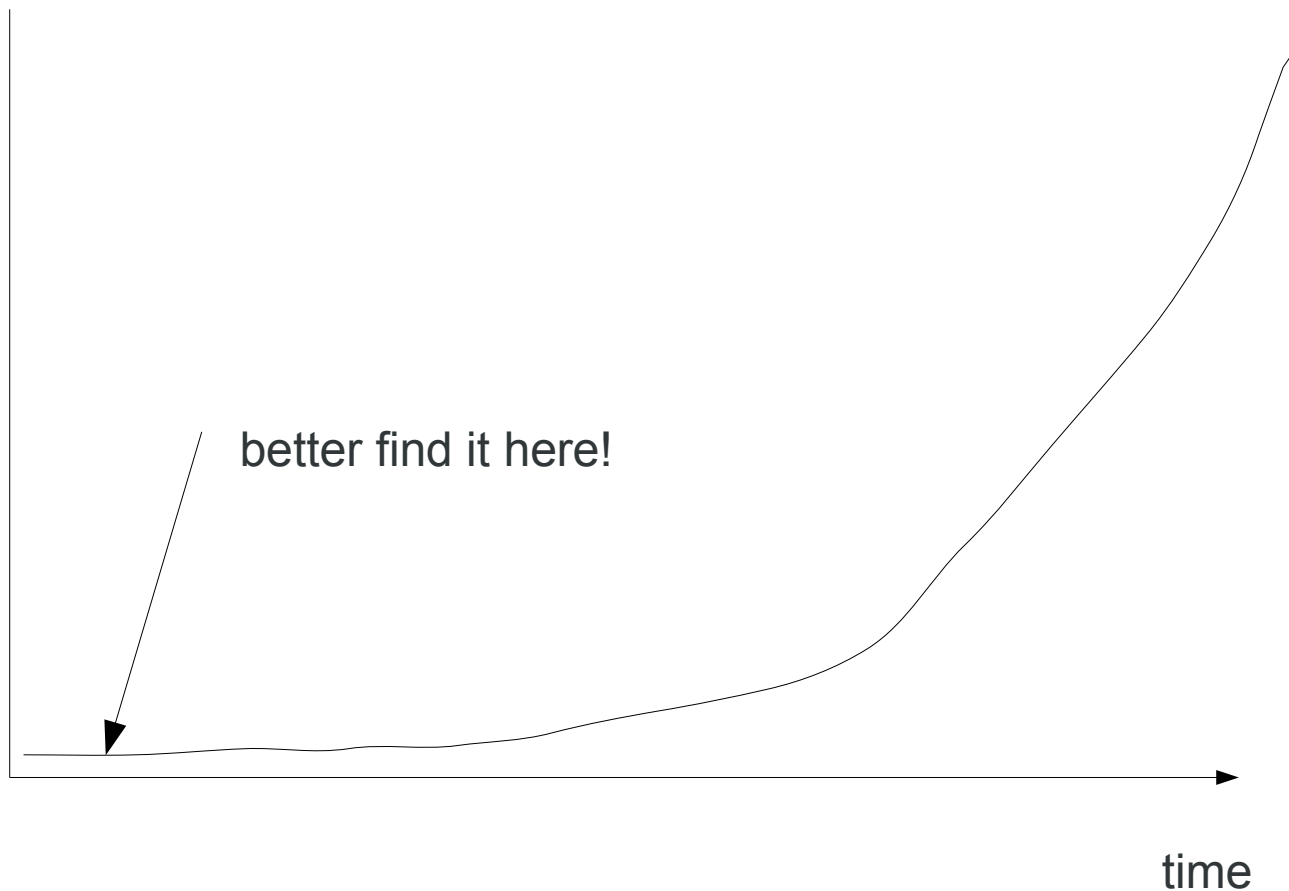
Software Development as a Journey

- You have (unclear) goals, with limited resource.

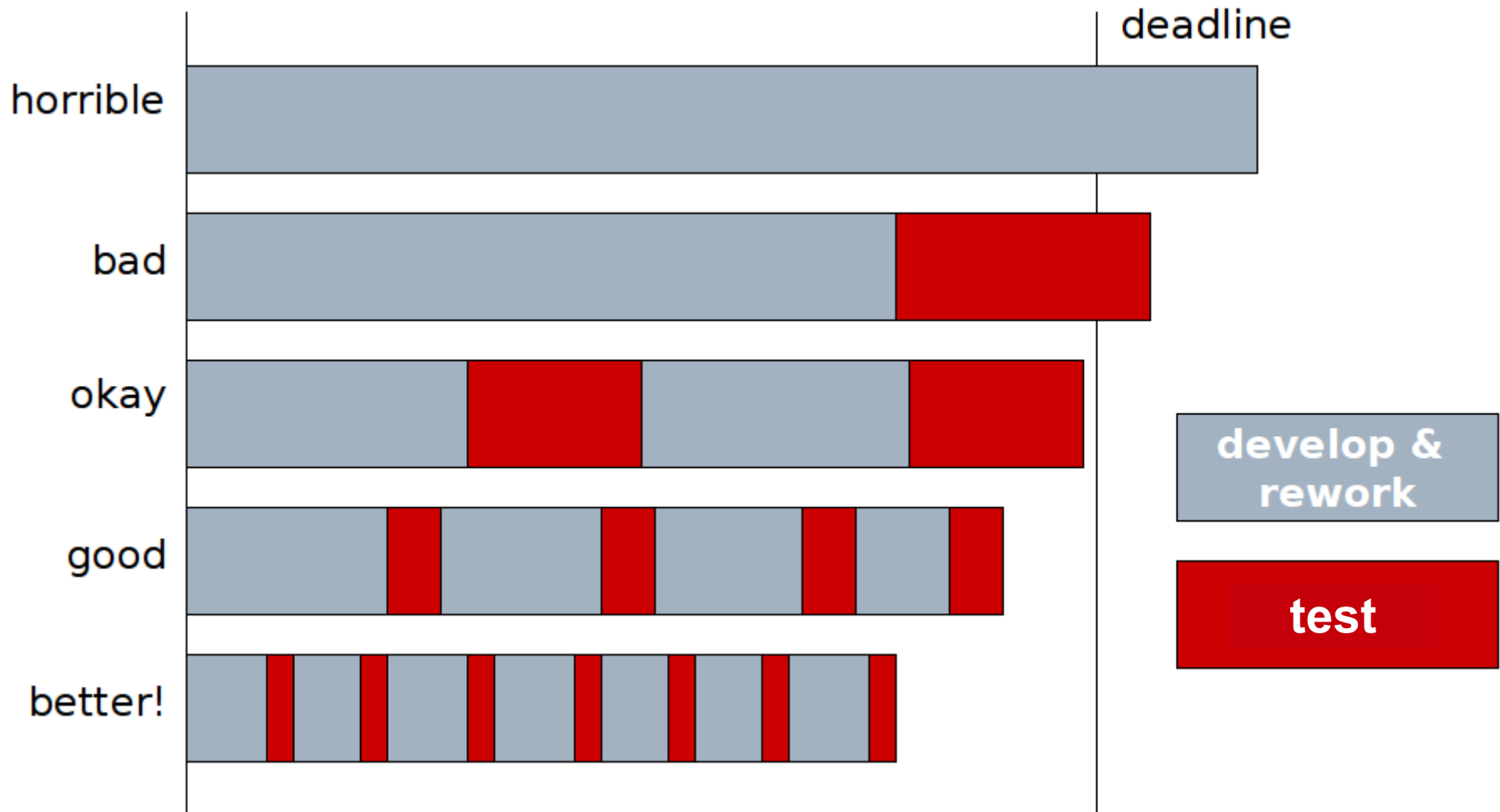


Fact 1

- The cost of fixing a defect increases (badly) over time.

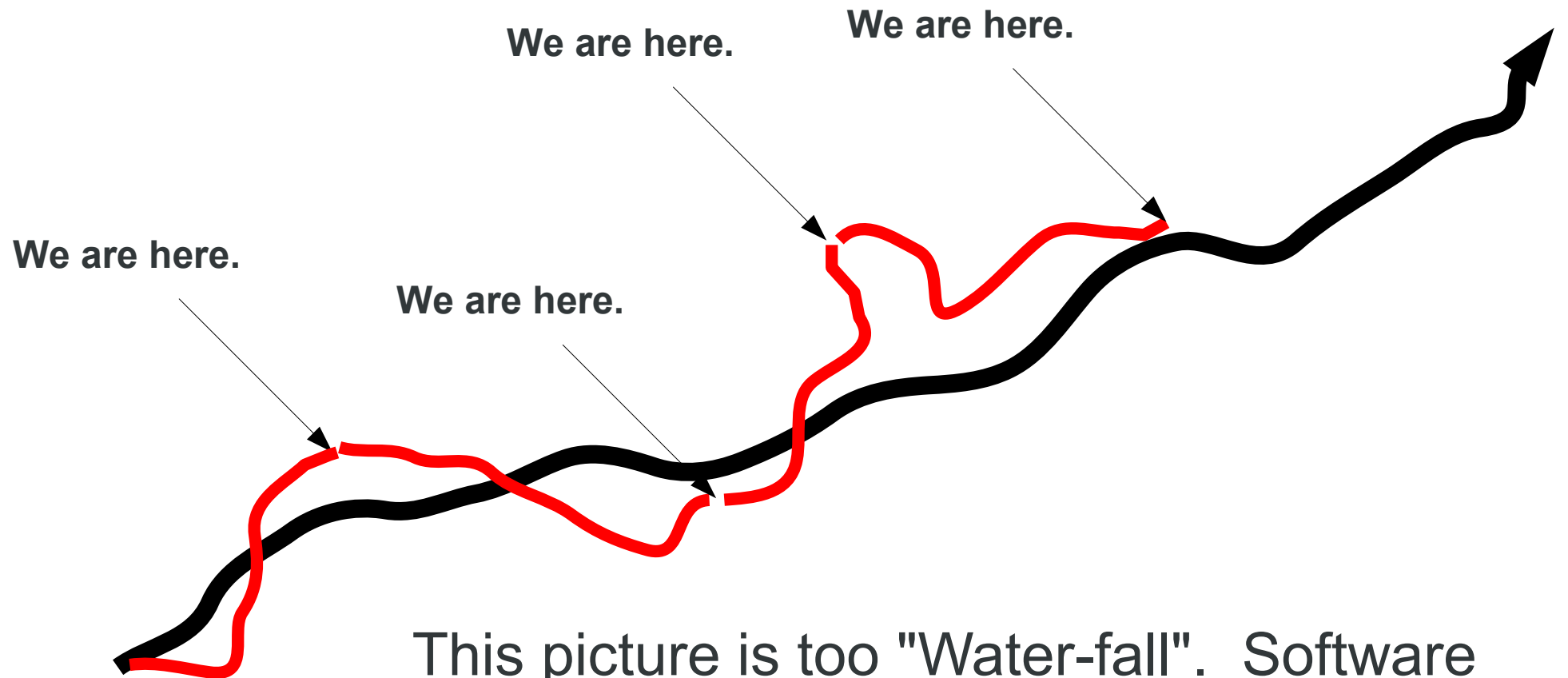


Rule 1 : Test Early and Often



Software Development as a Journey

- Test early, test often



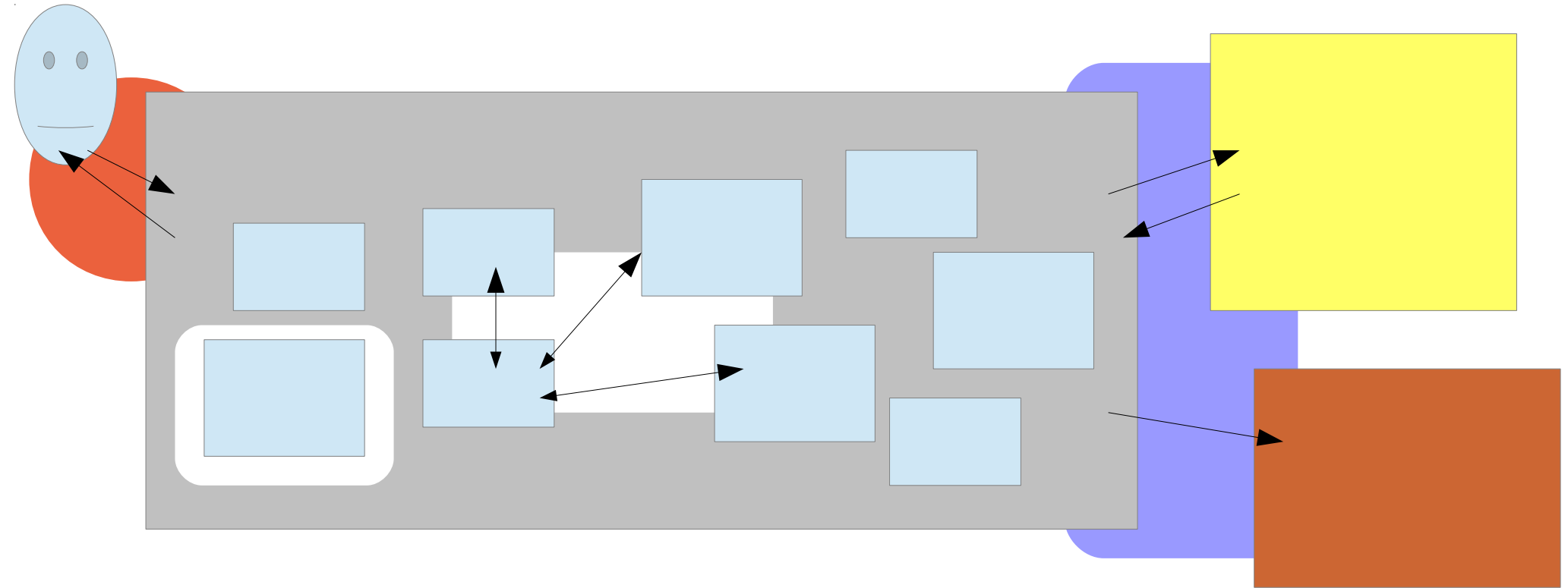
This picture is too "Water-fall". Software development rarely has clear, defined goals.

Software Development as a Journey (Agile view)

- Test early, test often



The BIG Picture : Levels of Testing



- There are many levels of software components; there are many level of testing with different techniques and focuses.

Levels of Testing

- Component testing (unit testing)
- Integration testing
- System testing
- User acceptance testing

The BIG Picture : Test Types

- There are many test types, for verifying the system under on various reasons and goals.
- Sample test types
 - Functional testing
 - Non-functional testing (-ility):
 - Usability testing
 - Scalability testing
 - Load testing
 - Performance testing
 - Regression testing

Regression Testing

- When defects are found, the developer fix the problems. The testers re-test to make sure the problems are actually fixed.
- Then, developers add more features. How can you be sure that these new features do not break the old ones?
- You perform **regression test** to make sure that the tests that are already passing still pass after the modification.

Changes

- When new code is added to the code-base, regression test should be done.
- For manual tests, it is very difficult to re-test every test cases, so regression testing has to perform on selected set of test cases (and less frequently).

Automated Testing

- Automated tests are usually scripted tests.
- You can run it as often as you like because it is usually does not take much time and resource.
- Automated test suite can be run when new code is checked in into the code repository.

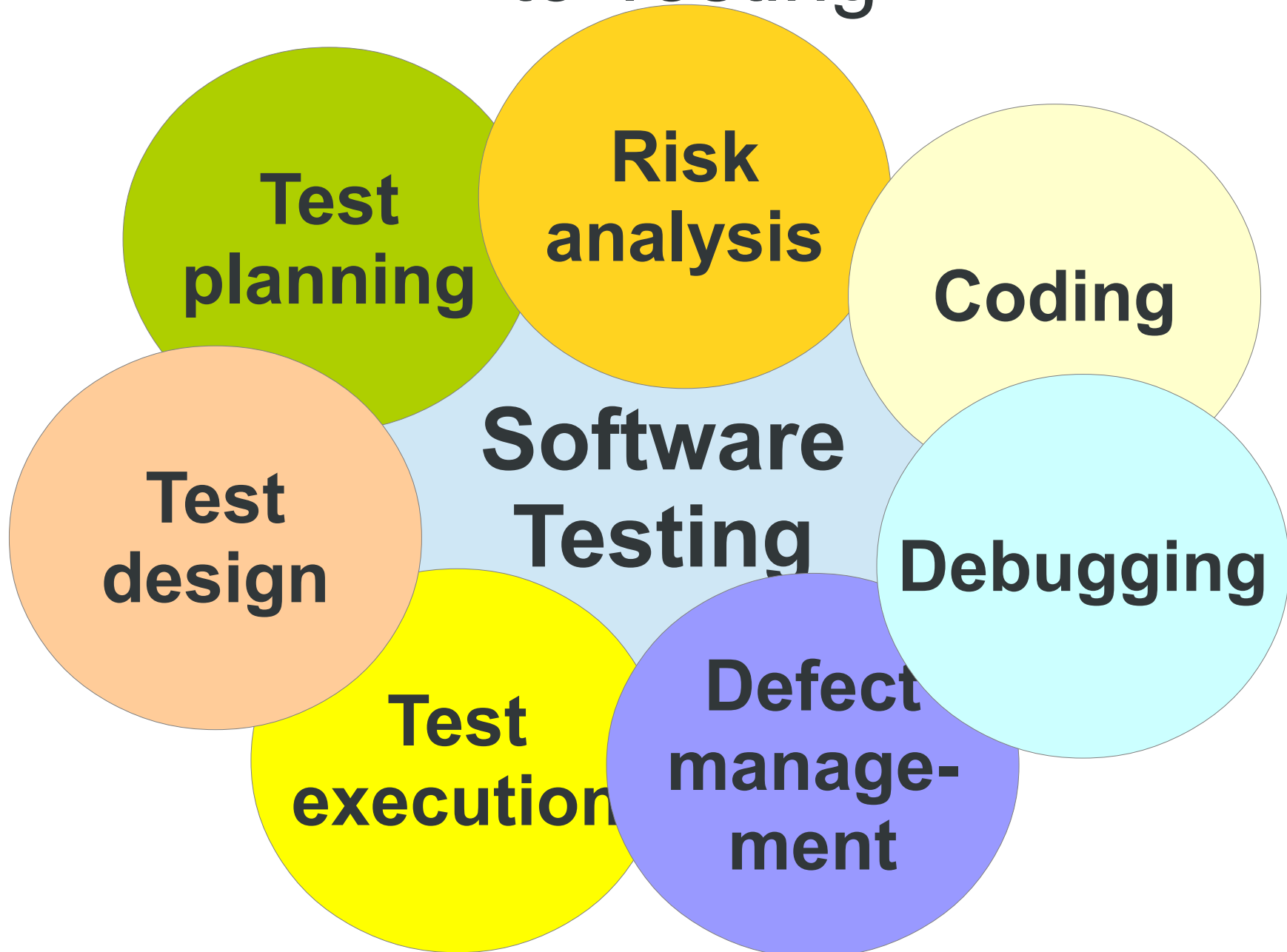
Automated Testing Heaven

- If you can automate all tests, then you can be confident about the code quality after changing the code.
- In this course, we will learn techniques to automate as many testing activities as possible.
- We will see that for some type of modules, automated testing is very easy to do; but for other types of modules (e.g., UI) automated testing might be too hard to do.

The BIG Picture : Roles Related to Testing



The BIG Picture : Activities Related to Testing



Contents of This Course

- **One sentence:** Software testing techniques and various activities related to software testing

Contents of This Course

- Developer testing
 - Unit testing, TDD, Automated testing
- Testing techniques
 - Test case design
 - Load testing / stress testing
- Test management
 - Test processes
 - Test planning

Answer: why do you test?

- We test to find defects.
- We test to be confident about the quality of the software.
- We test to prevent defects.

Practice

- We want to test Facebook for features related to privacy.
- Form a group of 3 people, play with Facebook, and develop test cases. Each group should write at least 5 test cases.

Write you test cases in this format

- **Purpose:** (describe what this test case verifies)
- **Steps:**
 - (describe how to perform the testing)
 -
 -
- **Expected result:** (explain the expected result)

Discussion

- How long does it take to perform one test case?
- What do we need to automate these test cases?

Next week

- Bring your laptops.
- We will program in Java, so if you need IDE, please have it ready to use.
- Get your IDE integrated with Junit.
- Prepare to have some fun!
 - [Google/youtube: TDD](#)