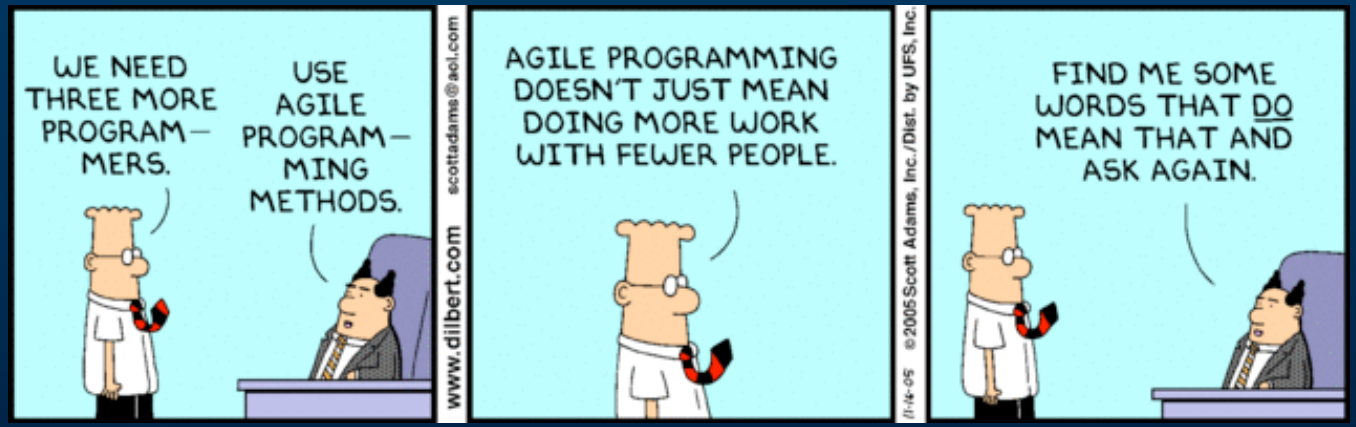
D0020E PROJECT I COMPUTER SCIENCE 2023/2024 LECTURE 3.2: TEST (PART 2)



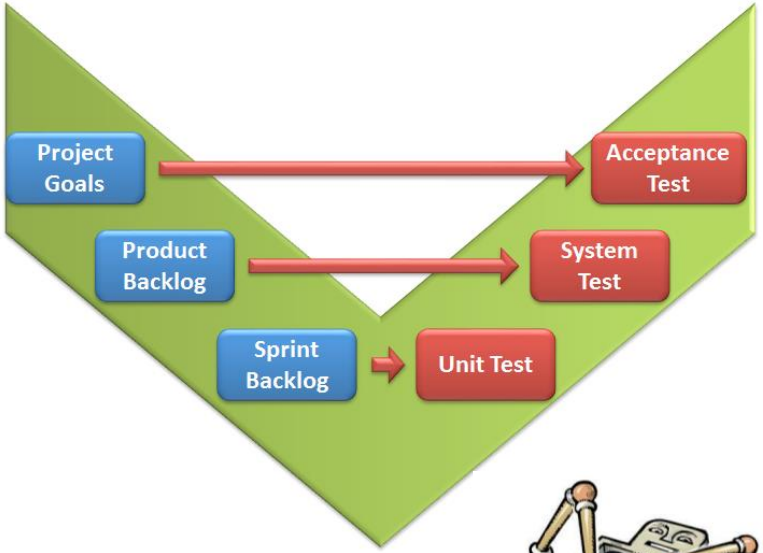
Ulf Bodin

LTU 02.11.2022

LULEÅ UNIVERSITY OF TECHNOLOGY

HOW ARE WE GOING TO WORK?

- Make sure defining tests to stories
  - Unit tests
    - For sprint backlog **tasks**
    - Means often to test/verify **tasks**
  - System tests
    - For product backlog **stories**
    - Means to test all finished **stories**
- Acceptance test at sprint review
  - Not right, but how we'll do it in practice
  - Should be done as a testing event with customer or as a dedicated activity that is preferably done outside sprints
- Automated testing
  - Quite often the same thing as regression testing
  - Throughout the project, consider test automation when doing unit testing and when preparing demos and deliverables
  - Define and document your strategy for test automation at unit and system level

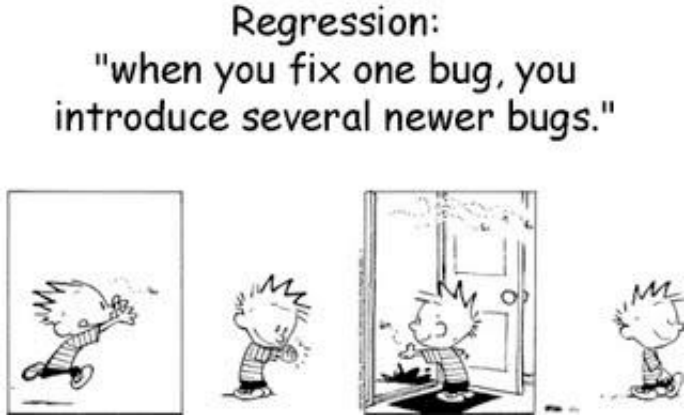


REGRESSION RISKS

- Maintainability
  - Requires that regression can be found
  - AND efficiently analyzed so that errors can be fixed
  - A testing framework can help with this task
- Regression
  - “When a feature stops functioning as intended after some kind of change to the system is integrated”
- Different types of regressions
  - Local regression
  - Remote regression
  - Unmasked regression
  - Performance regression

SOFTWARE REGRESSION

- Different types of regressions
  - Local regression
    a change introduces a new bug in the changed module or component
  - Remote regression
    a change in one part of the software breaks functionality in another part
  - Unmasked regression
    a change unmasks an already existing bug that had no effect before the change
  - Performance regression
    a change degrades the performance of the system



REGRESSION RISKS

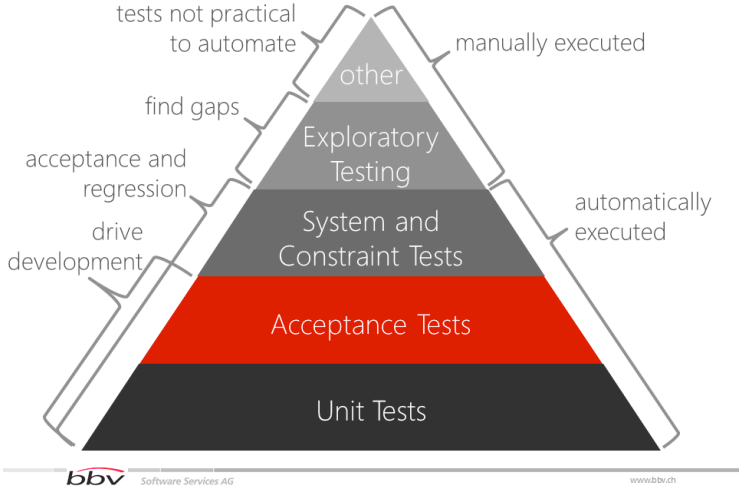
- Different types of regressions
  - Local regression
    - Change affects the changed component directly (component already exists)
    - For example, if a component is modified to allocate and free memory but stale pointers are used, the component itself fails
  - Remote regression
    - Change to one component **breaks** functionality in another component.
    - This reflects violation of an implied but possibly unrecognized contract between components
  - Unmasked regression
    - Creates a situation where a **previously existing** bug that had no affect is suddenly exposed in the system
    - May be as simple as exercising a code path for the first time
  - Performance regression
    - Causes the performance requirements of the system to be violated.
    - For example, changing a low-level function that is repeated frequently may reduce performance and render the system unusable or provide less satisfaction

HOW TO FIND REGRESSION

- There are most often not possible to test everything although extensive automation
  - Test automation and manual testing consumes time
  - Hence, it is essential to chose wisely
- Strategies to chose what to test
  - Addressing technical risk
    - Traceability
      - Select tests based on **behavioral descriptions** of the system
      - Requirement and design specifications, and documented quality risks
      - Activity and sequence charts may come useful
    - Change analysis
      - Look at **structural descriptions** of the system
      - Search for how change effects could ripple through the system
      - Class and module charts may be useful (knowing the code structure)
  - Addressing business risk
    - Quality risk analysis
      - Search for areas that have high business risk
      - **Requirements documentation** used in dialogue with the customer is useful

CROSS-FUNCTIONAL TESTING

- Although wisely selected tests, errors in unanticipated areas can be missed as not all functions are tested
- Cross-functional testing
  - Run a test suite with functions that **should not be affected** by a given update
  - Exploratory, system, acceptance and unit testing
  - Possible strategy
    - Cover large areas of the system to make an initial wide search
    - Focus on areas were bugs have appeared before
    - Intensify testing in areas were new bugs are found



LULEÅ
UNIVERSITY
OF TECHNOLOGY