Verification and validation

Verification&validation

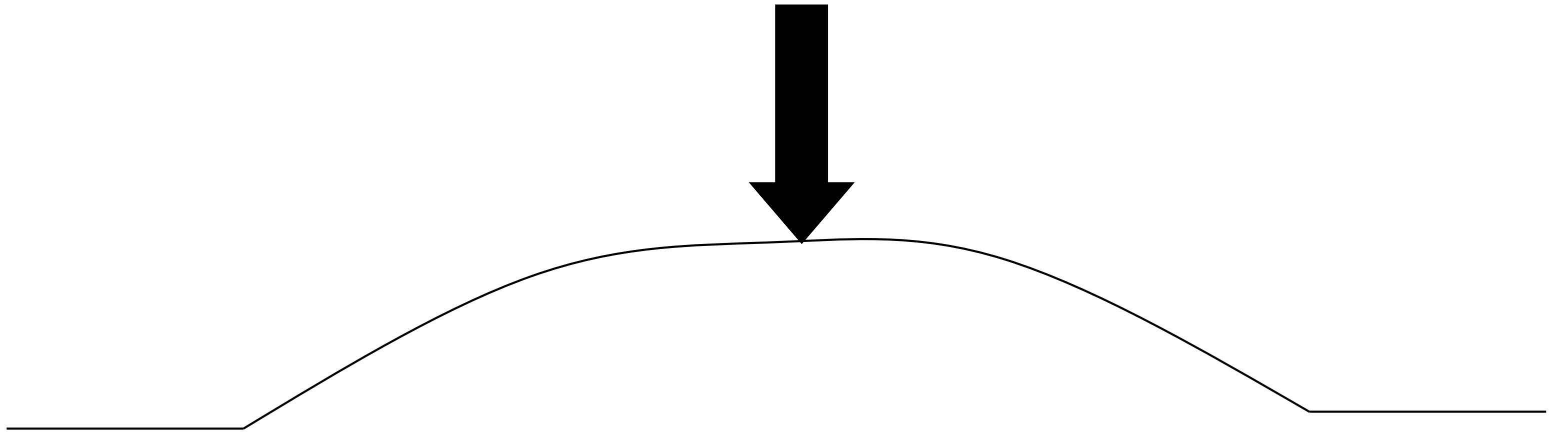
- Verification
  - *did we build the program right?*
- Validation
  - *did we build the right program?*

Why, what, where?

- Zero defect software practically impossible to achieve
- Careful and continuous verification needed
- Everything must be verified (spec. documents, design documents, test data, …)
  - even the verification must be verified!
- Verification along the entire development process, not just at the end

Verification in engineering

- Example of bridge design
- One test assures infinite correct situations



Verification in software engineering

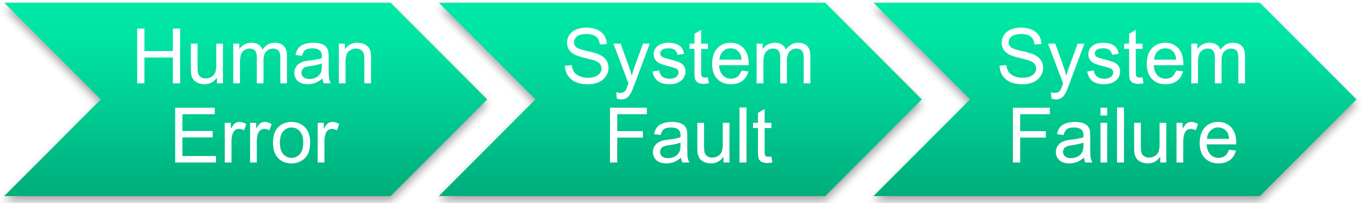
- Programs do not display a “continuous” behavior
- Verifying the function in one point does not tell us anything about other points
  - Example 1

```
. . .
a =  … / (x  +20) ...
. . .
```

Any value of x is ok, except for x = -20!

Terminology

Term	Description
Human error	Human action that results in software containing a defect or fault
System error, aka fault or bug	Discrepancy between an observed, computed, measured value and the true, specified, or theoretically correct value
System Failure	Inability of a system or component to perform a required function according to its specification


Human Error

System Fault

System Failure

Faults, errors and failures

- Failures are usually a result of faults introduced by a human error
- Faults do not necessarily lead to system failures
  - The error can be corrected by built-in error detection and recovery
  - The faulty system state may be transient and ‘corrected’ before a failure occurs

Difficulties in V&V (1)

- Checking of some qualities does not have a binary (yes/no) outcome
- Many properties are subjective
- Some are even implicitly stated

Difficulties in V&V (2)

- Qualities are not clearly stated or
- Are not reasonable (is 100% availability a reasonable goal?)
- The relative importance of qualities and their relationships with other project objectives needs to be identified

Difficulties in V&V (3)

- It is almost impossible to develop error free software
- New approaches and technologies may introduce new errors and problems
  - E.g., transition to new language or development environment
- Challenge
  - find the right blend of verification and validation approaches for each specific software

When do V&V start?

- As soon as we decide to develop a product
- During feasibility study we consider
  - functionality, required qualities and their impact on costs

- Quality manager participates in the feasibility study
  - focuses on how to assess and control quality during development
  - influences the definition of the preliminary architecture of the system in order to ensure that it can be tested and analyzed more easily

An example

- The development of a web application
  - If the application is decomposed in three layers (UI, business and data layers) the quality assurance team can be structured accordingly
    - The human interface group is responsible for usability
    - The key quality people can be involved in checking the kernel of critical functions within the business and data layers
    - Less experienced persons can take care of the other parts
  - Some preliminary decisions about the quality assurance approach can be taken. For instance:
    - A first prototype will not go through a complete acceptance test but will be used to validate requirements and design
    - The acceptance test for the first release will be focused on usability feedback from a subset of users and will check typical security problems
    - The acceptance test for the second release will include a check of all functionalities and reliability measures

What V&V technique should be applied?

- The choice depends on quality, cost, schedule, resource constraints
- Combination of different techniques because
  - Each technique may be effective for different classes of faults
  - May be applicable at different points in a project
  - May have different purposes
  - May have different tradeoffs in cost and assurance

An example

- While developing our web application
  - A semi-formal notation is used for requirement description and system design
    - The quality manager decides to use *inspection* to check these documents
      - performed by single persons or small groups for design documents
      - performed by a larger group according to well formalized procedures for req. descriptions and specs.
  - For unit test each developer is required to produce functional test cases together with the code
    - If less than 80% code statements are executed by these test cases, other tests are identified by using a structural approach (the company has a tool to evaluate test coverage)
  - Integration and system test cases are generated by the quality team. Scaffolding and oracles are part of the system architecture

we will see these

How can we assess the readiness of a product?

- Finding all faults is nearly impossible
- Analysis and testing cannot go on forever
- … but the product should be delivered when it meets the functionality and the quality required by the market (e.g., dependability)

- Examples of important measures for dependability
  - Availability: QoS in terms of running versus down time
  - Mean Time Between Failure (MTBF): QoS in terms of the length of time interval during which the service is available
  - Reliability: a fraction of all attempted system operations that completed successfully

How can we control the quality of successive releases?

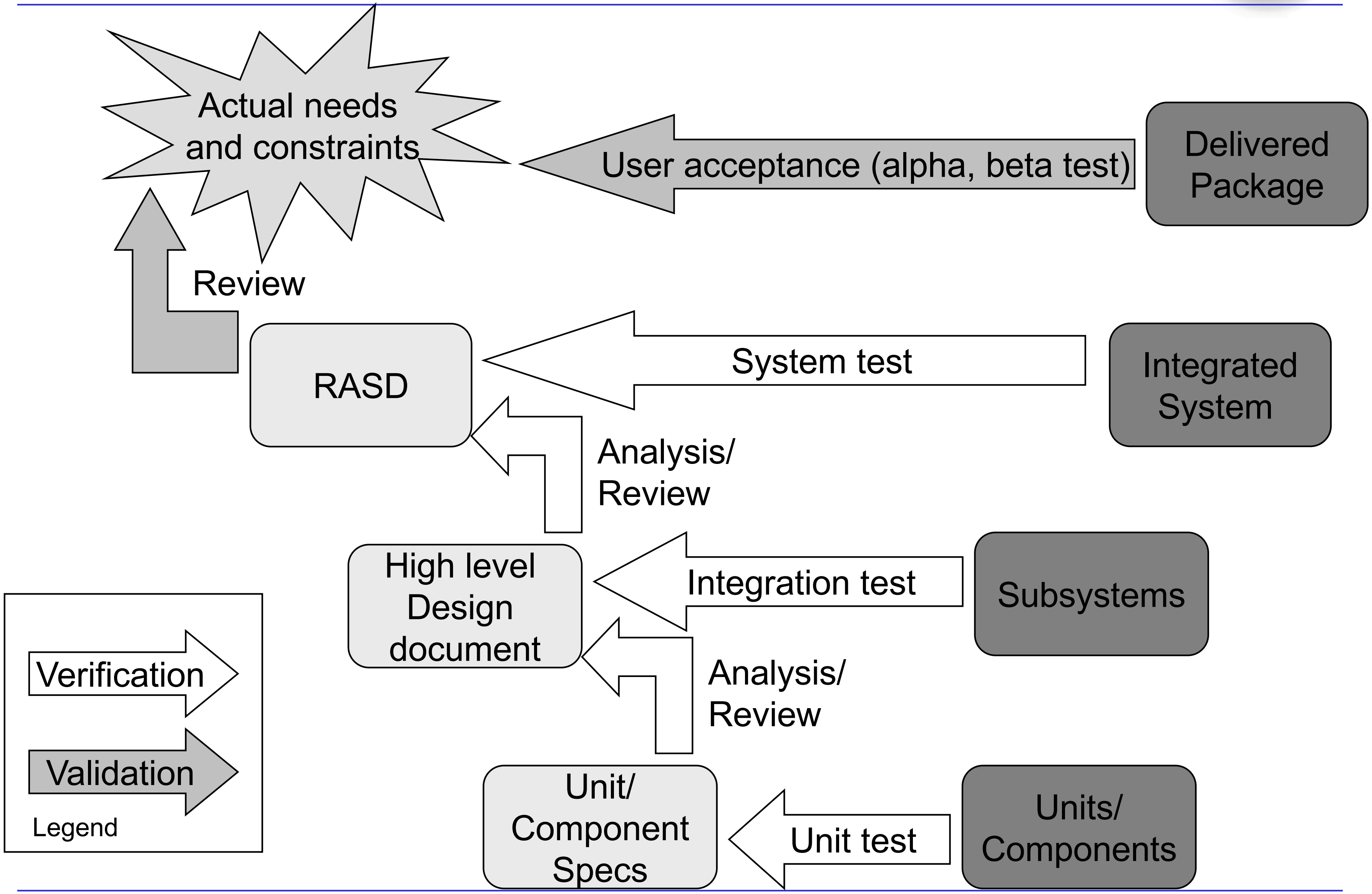
- Various new versions of a software can be produced during its life cycle
  - Patches
  - Major releases
- Tests already executed on the first release need to be executed again on the new versions (**regression testing**)
- Automatic test execution is desirable for speeding up the process
- New test cases are added to the regression test suite as a new version is developed

A short note on quality of tests

- Also tests need to be of good quality!

WE ADDED A NEW PERFORMANCE TEST, BUT LEARNED THAT THE TEST ITSELF IS FLAWED.
Dilbert.com DilbertCartoonist@gmail.com
NOW OUR PRODUCT FAILS OUR OWN TESTS AND OUR CUSTOMERS ARE ASKING TO SEE THE TEST RESULTS.
8-11-10 ©2010 Scott Adams, Inc./Dist. by UFS, Inc.
DO I HAVE PERMISSION TO FAKE THE TEST DATA?
I DIDN'T EVEN KNOW DATA CAN BE REAL.

V&V activities and software artifacts (the V model)



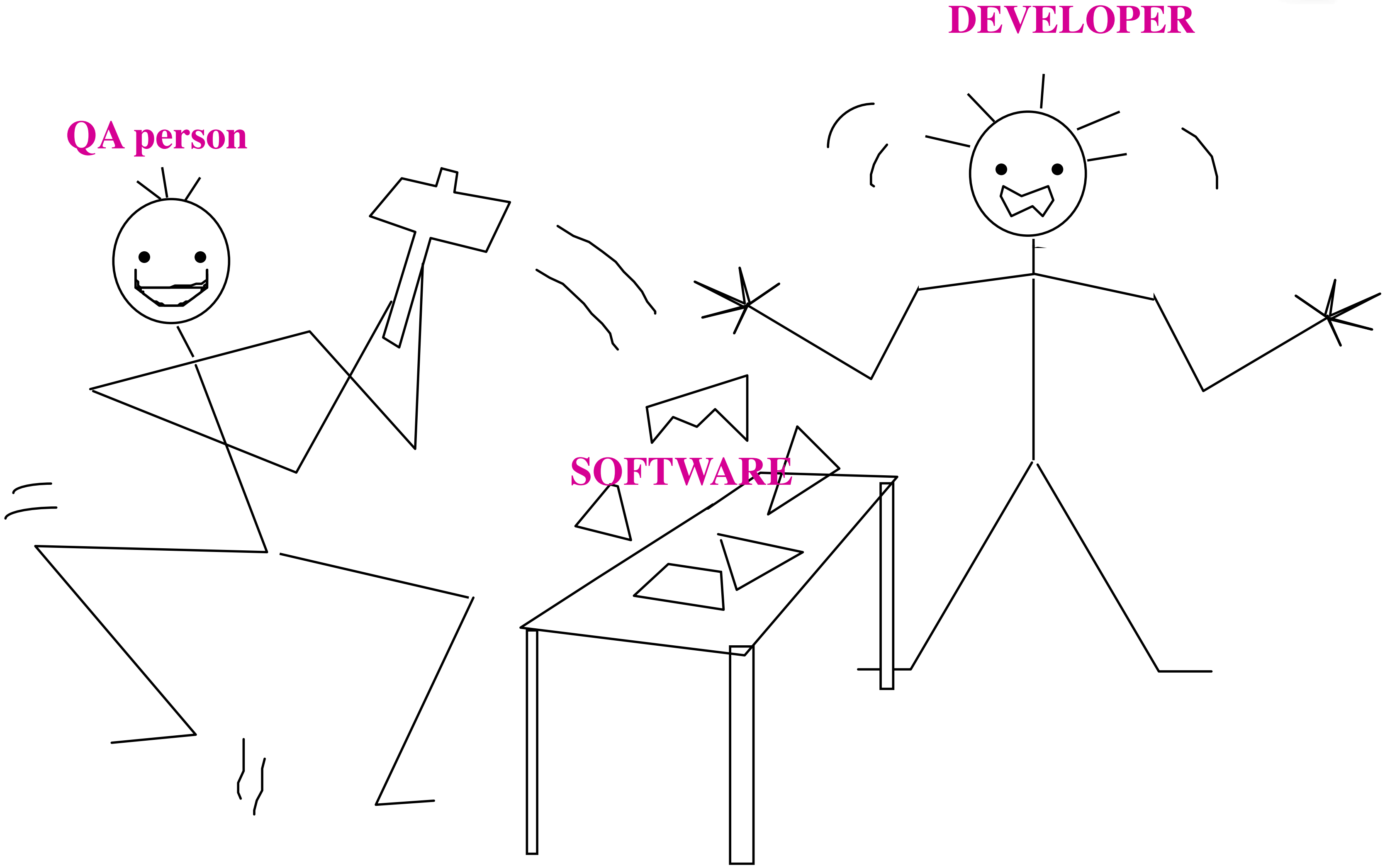
Planning and monitoring

- An *analysis and test plan* identifies
  - Objectives (quality goals) and scope
  - Documents and items that need to be available to perform the various quality assurance activities
  - Items to be tested (and features to be tested)
  - The analysis and test activities to be performed
  - The staff to be involved
- It includes
  - Constraints, pass/fail criteria, schedule, deliverables, hw and sw requirements, risks and contingencies
- Process monitoring and visibility is very important
  - Visibility on the schedule (are we on time with respect to the plan?)
  - Visibility on the achievement of the quality goals

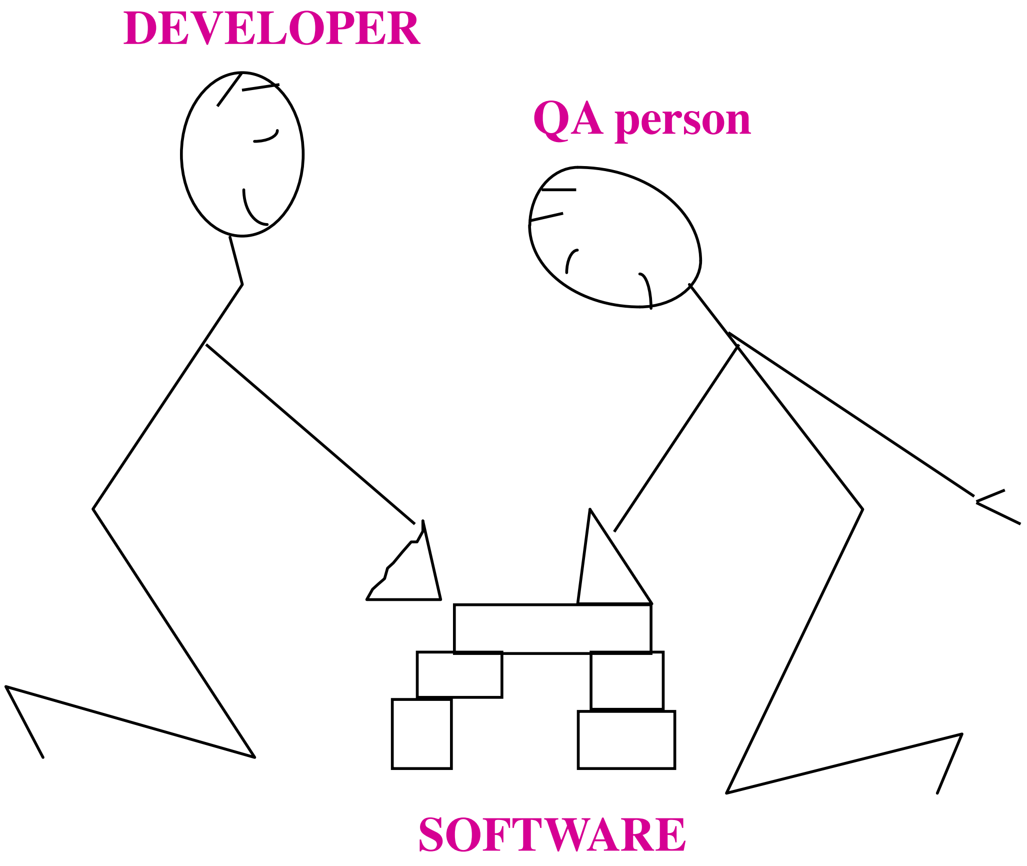
The V&V process improvement

- Should be part of the overall process improvement process
  - Team members should be properly motivated
- Based on analysis of faults detected in previous projects and on the identification of the errors that caused them
- Four phases:
  - Defining the data to be collected about faults
  - Analyzing collected data to identify fault classes
  - Analyzing selected fault classes to identify weaknesses in the development and quality measures
  - Adjusting the quality and development process

Software Verification: from this ...



... to this



Different attitudes:

DEVELOPER	QA Person
•Optimistic	•Pessimistic
•How to better design	•How to better observe
•Interpret and repair bugs	•Discover and report bugs
•Focus on how it could work	•Focus on how it could break

Complementary

The quality improvement group should involve both developers and quality assurance people

Main approaches to V&V

- ANALYSIS (usually, static technique)
  - analytic study of properties
- TESTING (dynamic technique)
  - experimenting with behavior of the products
  - sampling behaviors
    - GOAL: find “counterexamples”