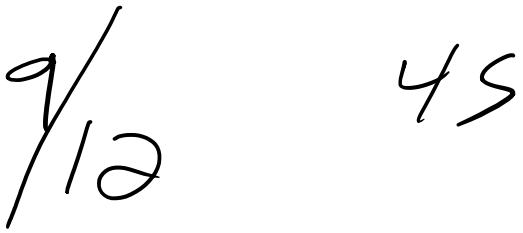
Some Classic Testing Mistakes

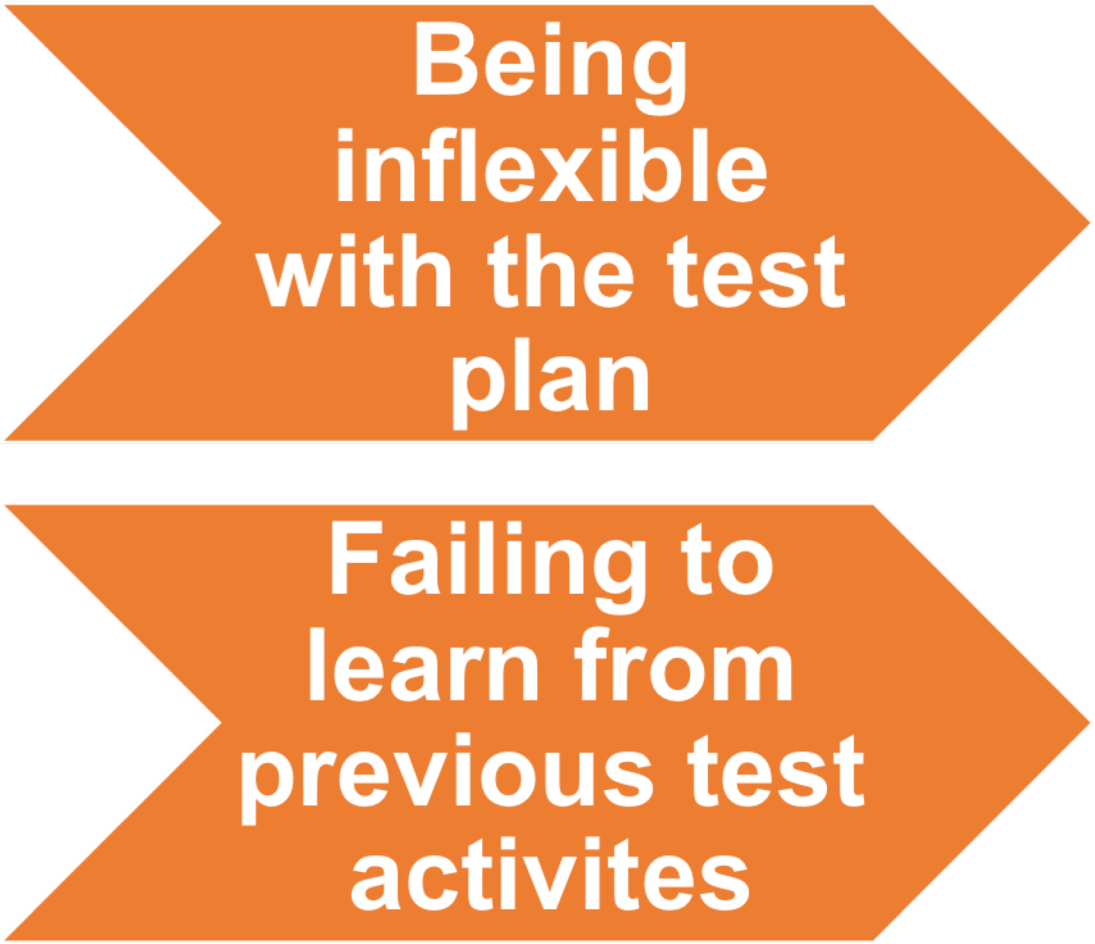
- **Believing the primary objective of system testing is to find bugs**
  - Test must concentrate on finding important problems
  - Test must provide an estimate of system quality
- **Not focusing on usability issues**
- **Starting too late**
  - Test must help development avoid problems



Some Classic Testing Mistakes

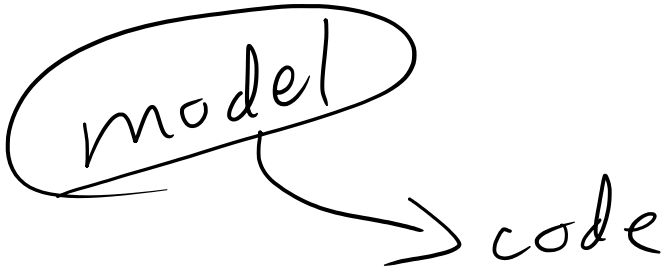
- Delaying stress and performance testing until the end
- Not testing the documentation
- Not staffing the test team with domain experts
- Not communicating well with developers
- Failing to adequately document and review test designs

Some Classic Testing Mistakes

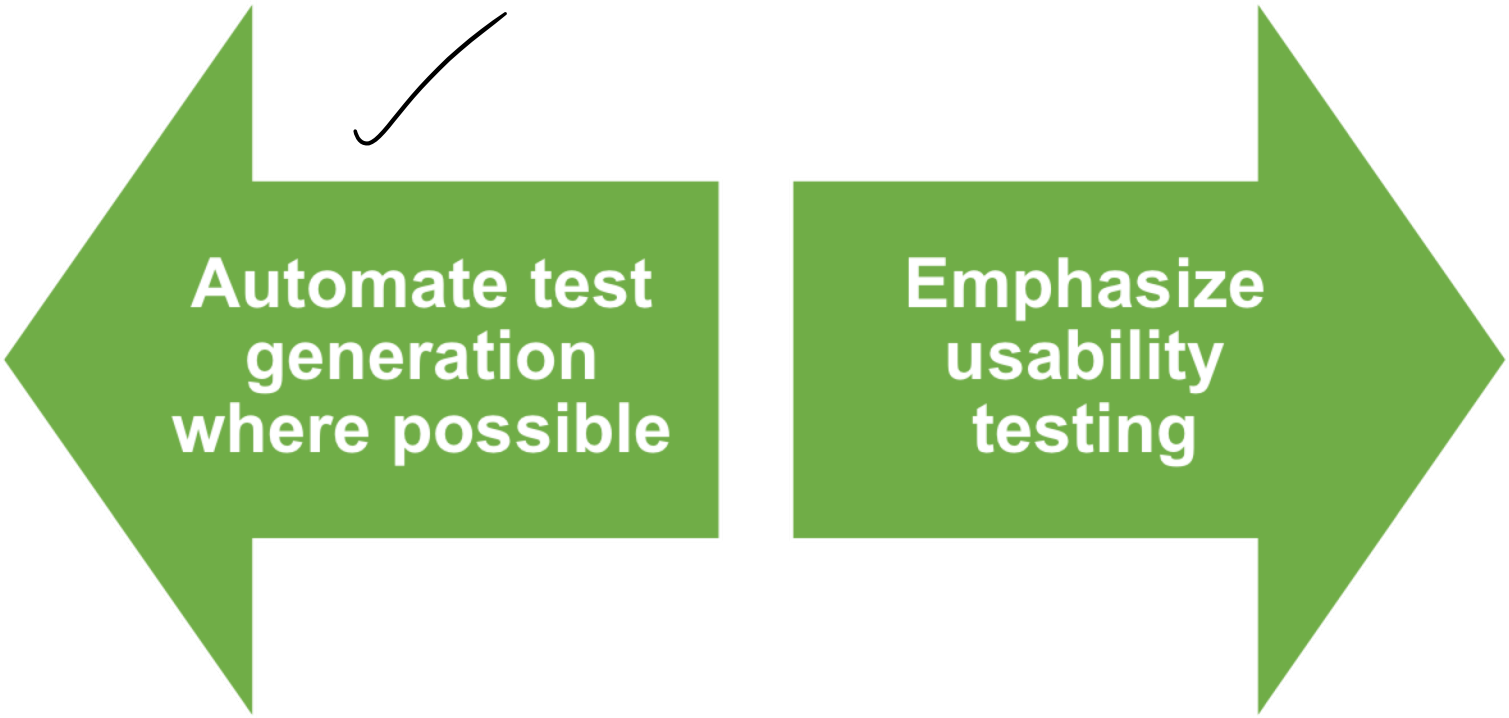


Best Testing Practices

- **Assess software reliability via statistical testing**
- **Utilize model-based testing techniques**
  - State diagrams
- **Develop an agile test design**
  - Accommodate late changes
  - Emphasis on regression testing
- **Develop cross-functional development and test teams**



Best Testing Practices



Definitions

Validation:

- Are we building the right product?

Verification:

- Are we building the product right?

Testing:

- The examination of the behavior of the program by executing it on sample data sets

Definitions

Error:

- Mistake made by a human

Defect/Fault:

- Result of error manifested in the code

Failure:

- Software doesn’t do what it is supposed to do

Poor Reliability

- Software defects rates are around 1 delivered defect per thousand lines of code
- With applications spanning millions of lines of code, customers experience many defects

Definitions

Reliability:

- The probability that a software program operates for some given time period without software error

Testers vs Pollsters Analogy

Introduction to Software Verification, Validation and Testing

Testing Background

Ira A. Fulton Schools of Engineering
Arizona State University

Objective

Objective

Define common testing terminology

History

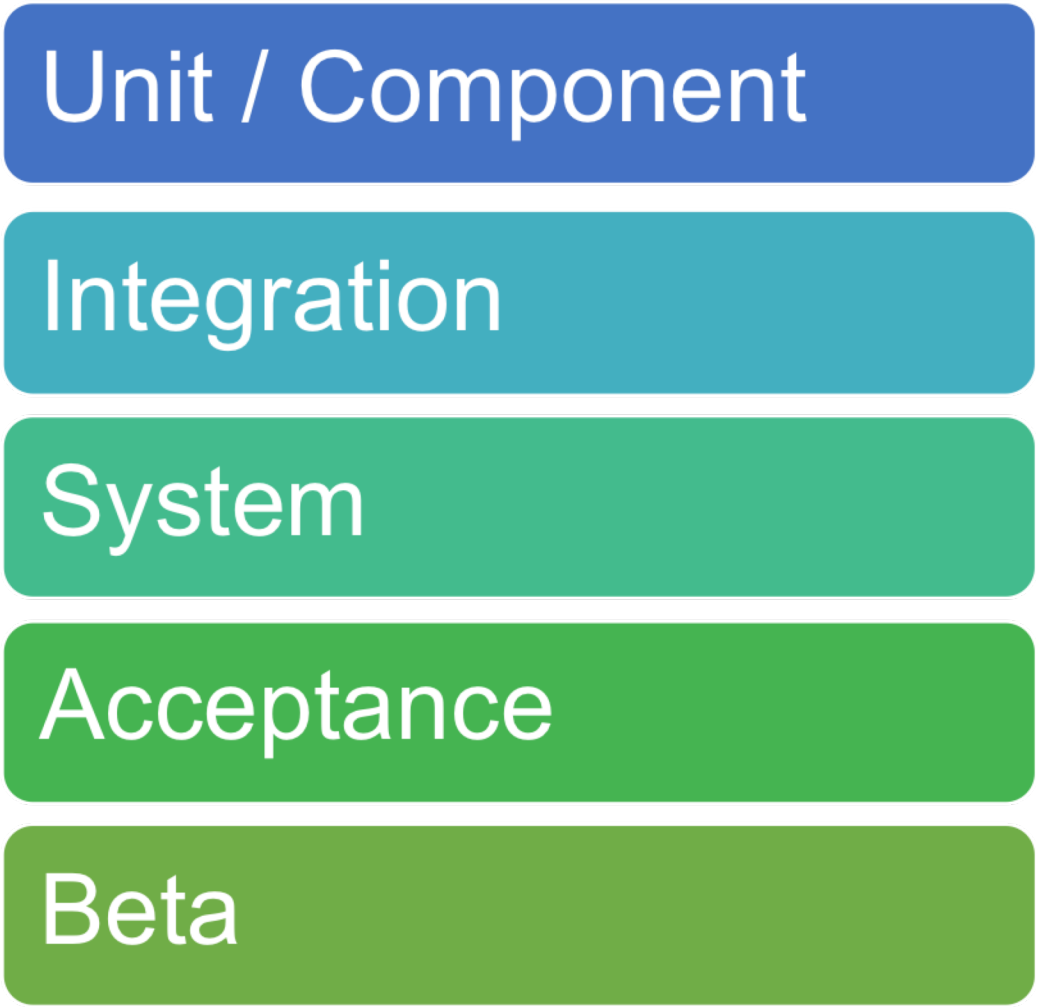
The term software engineering was first used at a workshop in West Germany in 1968 considering the growing problems of software development

- High Cost
- Difficult to Manage
- Poor Reliability
- Lack of User Acceptance
- Difficult to Maintain

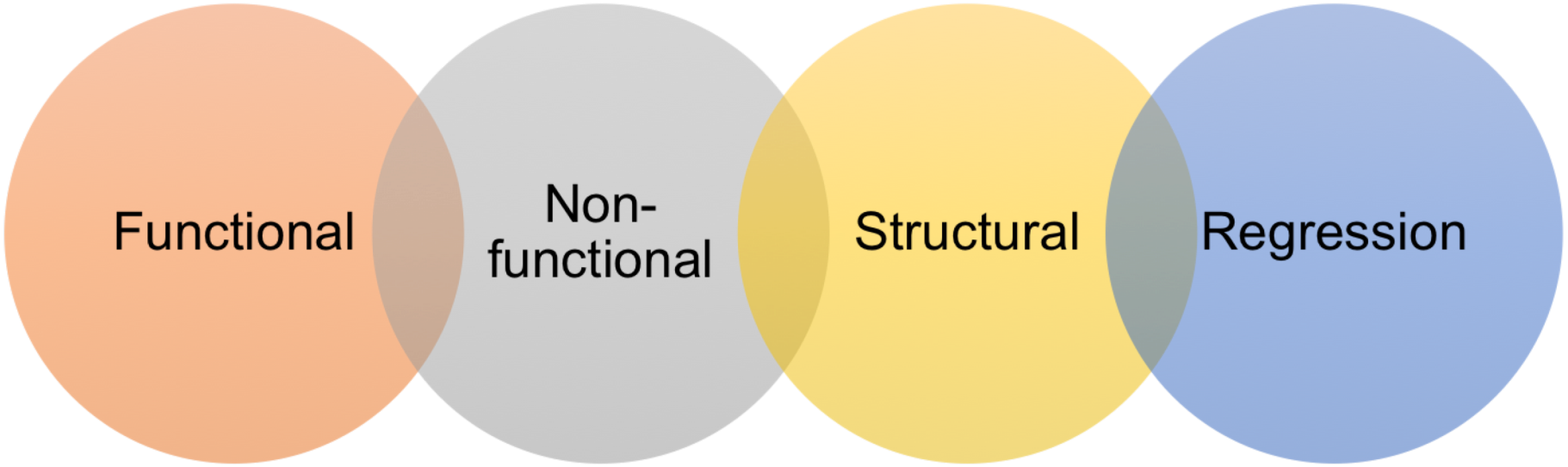
Current State of Software Development

- High Cost
- Difficult to manage
- Poor Reliability
- Lack of User Acceptance
- Difficult to Maintain

Testing Levels



Test Types



Summary

Introduction to Software Verification, Validation and Testing

Testing Principles and Best Practices

Ira A. Fulton Schools of Engineering
Arizona State University

Objective

Objective

Explain best practices for software testing.

Testing Principles

Principle 1:

- Testing only shows the presence of defects – not proof of correctness

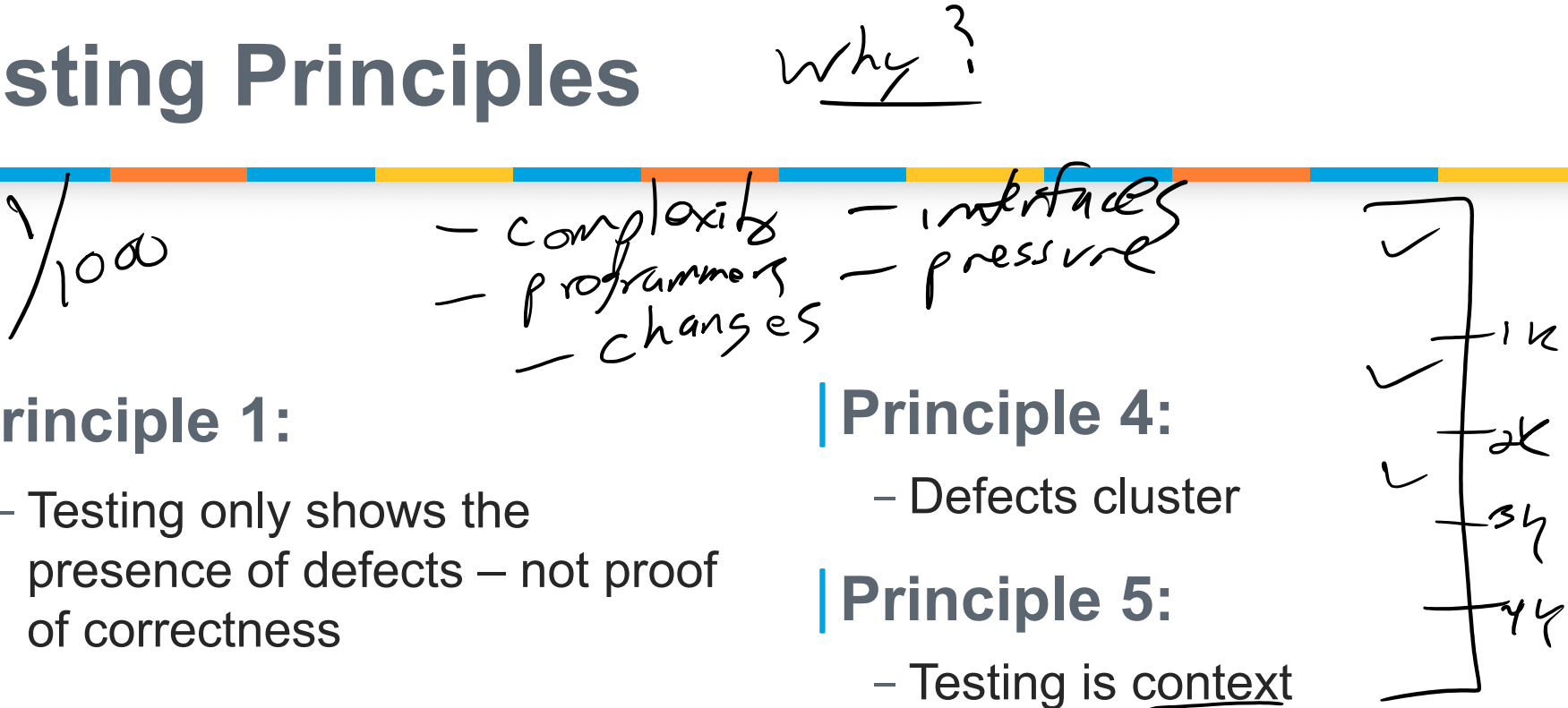
Principle 2:

- Exhaustive testing is impossible

Principle 3:

- Start testing early

Testing Principles



Principle 1:

- Testing only shows the presence of defects – not proof of correctness

Principle 2:

- Exhaustive testing is impossible

Principle 3:

- Start testing early

Principle 4:

- Defects cluster

Principle 5:

- Testing is context dependent

Principle 6:

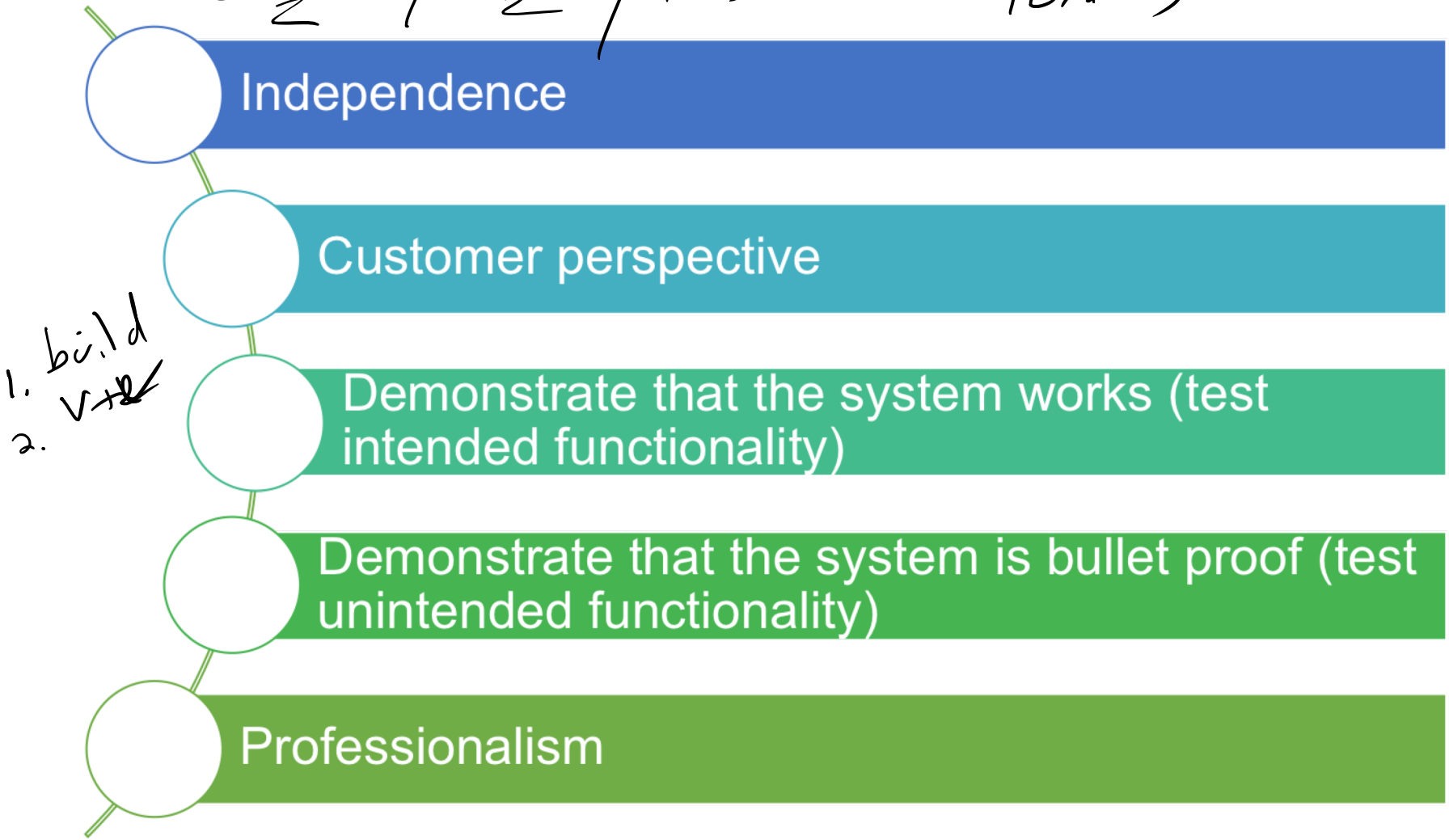
- Absence-of-errors fallacy

Testing Attitude

SE Licensing

Education / Experience / Tests

TEXAS



1. build
2. V+V

Introduction to Software Verification, Validation and Testing

Testing Throughout Life Cycle

Ira A. Fulton Schools of Engineering
Arizona State University

Objectives

Objective

Describe how testing is integrated into software development phases

Objective

Define the objectives of the different levels of testing

Testing Throughout Life Cycle

- Waterfall
- Agile
- TDD

- Agile Methodology: The Complete Guide to Understanding Agile Testing

Agile Testing

Continuous Integration (at least daily)

- Static Code Analysis
- Compile
- Unit Test
- Deploy into Test Environment
- Integration / Regression Test

Test Driven Development (Red, Green, Refactor Cycle)

Red Phase:

- Write a minimal test on the behavior needed

Green Phase:

- Write only enough code to make the failing test pass

Refactor Phase:

- Improve code while keeping tests green

Software Development Process vs Test Development Process