Table of Contents

1. Software Development Methodologies

2. Quality Assurance
   - Goals
   - Process

3. Challenges of Testing
Waterfall Model
Waterfall Model

- Simple
- One of the 1st models
- Sequential
- Slow
- Feedback and Evaluation at the end.
- Testing is final step
- Initial faults are costly

- Maintenance
  - Is it broken?
  - Out of Date?
- Upgrading
- New Versions
- Changing Environment
  - Laws
  - Technologies
  - etc.
Spiral Model

Determine Objectives

Identify Risks

Prototype 1
Prototype 2
Prototype 3

Development and Testing

Plan next iteration
Spiral Model

- Multiple Waterfall Models.
- Iterations
- Repetitive
  - Requirement Specification
  - Feasibility Study
  - Design
  - Implementation
  - Testing
  - Delivery

+ Smaller Iterations
+ Short Deadlines
+ ”Quick”
+ ”Safety”
+ Fewer mistakes
+ Less risk
  - Iterations are closed
  - Strict sequence
  - Testing is a final step
Table of Contents

1 Software Development Methodologies

2 Quality Assurance
   - Goals
   - Process

3 Challenges of Testing
Goals of Quality Assurance

1. Flawless (current) product. (Short term)
   - Focus on current project.
   - Flawless does not exists. So as perfect as
     - possible (theory, in classroom)
     - necessary (better companies)
     - required in order to someone pay for it (other companies)

2. Determine the quality criteria in general. (Medium term)
   - Test coverage.
   - \# of \{minor, major, critical\} bugs in delivered product.
   - Support, bug fixes and patches.

3. Improve development processes and efficiency. (Long term)
   - Training (technology, soft skills, communication)
   - Monitoring employees performance.
   - Let’s fire Joe because he creates a tons of bugs.
Improve Current Product

- Fault Types
  - Syntax
  - Semantic
  - Unexpected Value
  - Wrong Value
  - Imprecise Definition
  - Wrong Decision

- Costs increase with time.
Process Improvement

- Development process
  - Inter Team Communication
  - Work-flow
- Project independent
- Frequent faults
  - Data collection
  - Categorize
  - Analysis
- Big Data
Criteria Definition

"Well-known oval red object with white marks."

(a) Customer’s expectation  
(b) Specification  
(c) Implementation

Figure: Importance of Specification
Validation and Verification

"Are we building the right product?"
- Will the product satisfy the customer's needs.
- Prototypes
- Requirement Specification
- Alpha Testing
- Acceptance Test

"Are we building the product right?"
- Does the product meet the specification.
- Dynamic Testing
- Automated Tests
- Unit Test
- Component Test
- Integration Test
- System Test
Validation and Verification
Lack of Validation

- Product meets the specification.
- Product does not satisfy the customer’s needs.

(a) Customer’s Expectations
(b) Specification
(c) Implementation

**Figure:** Lack of Validation
Lack of Verification

- Specification represents the uses needs.
- Implementation differs from specification.

(a) Customer’s Expectations
(b) Specification
(c) Implementation

Figure: Lack of Verification
1 Software Development Methodologies

2 Quality Assurance
   - Goals
   - Process

3 Challenges of Testing
Quality Assurance Process

- Overlap multiple development processes.
- Asynchronous with development process.
- Support
  - developers
  - management
- Various intensity
Specification Analysis

- Feasibility Study
- Risk Analysis
- Main Functionalities
- Look for existing components.

- Setting Quality Gates
  - Reliability
  - Security
  - Robustness

- Design of Acceptance Test
- Outline System Tests
System Design

- Design System Architecture
- Definition of Major Components
- Components based Communications
- Assign Functionalities to Components

- Design of Unit and Component Tests
- Analysis of System Architecture
- Determination of Expected Test Results
Implementation

- Implementation of classes and packages
- Bug fixing

- Test running
  1. Unit
  2. Component
  3. Integration
  4. System
  5. Acceptance

- Code coverage
- Data collection
- Monitoring
- Fault Analysis
Maintenance

- Bug fixing
- Changing Environment
  - Laws
  - Business policies
  - etc.
- Data collection
- Fault Analysis
Table of Contents

1. Software Development Methodologies

2. Quality Assurance
   - Goals
   - Process

3. Challenges of Testing
State–Space

- Method
  - Access Modifier
  - Return type
  - Method name
  - Parameters
  - Exceptions

- Types
  - Domain
  - Precision

- Arbitrary Parameters
- Unexpected values
  - Negative age
  - Empty String
  - ...

Zsolt Tóth (UM)
class Trivial{
    static int sum(int a, int b) {
        return a + b;
    }
}

- State–Space: \((\text{Integer.MAX} - \text{Integer.MIN})^2 \approx 10^{21}\)
- Total enumeration is impossible.
- Testing is inaccurate.
Inaccuracy

- Pessimistic
  - Specification Analysis and proofs
  - Determines sufficient properties.
  - Costly

- Optimistic
  - Testing
  - Faults may be undetected if the program satisfies necessary conditions.
  - Why?
Köszönöm a figyelmet!