Validation & Verification

Agenda

- Quality Management
- Software Quality
- Validation & Verification
- Reviews & Inspections

Validation & Verification

- Validation – "Are we building the right product?"
  - demonstrate software meets its requirements
- Verification – "Are we building the product right?"
  - discover defects in the software
    - behaviour is incorrect
  - also called defect testing

V&V Confidence

- Establish confidence that system is "fit for purpose"
  - level of confidence depends on how critical the software is to an organisation
  - users may have low expectations
  - time to market may be more important than finding defects

Quality and Testing

- Testing does not *ensure* quality
  - hence quality management
- Testing provides an *indication* of product quality

Why Do We Test?

- To see if something works?
- To see if something doesn't work?

"Errors are more common, more pervasive, and more troublesome in software than with other technologies."
– David Parnas

Stages of Testing

- Development testing
  - to discover defects
- Release testing
  - complete system is tested before it is released to users
    - typically by independent test team
- User testing
  - users, or potential users, test the system in their own environment
    - alpha, beta, acceptance

Development Testing

- Unit testing
- Integration / Component testing
  - several units are integrated to create composite components
  - test interfaces
- System testing
  - components are integrated and system is tested as a whole
  - test component interactions
    - particularly emergent behaviour of system

Testing Policies

- Exhaustive testing is impossible
- Policies define required test coverage
- Examples:
  - all functions that are accessed through menus should be tested
  - where user input is provided, all functions must be tested with both correct and incorrect input

Release Testing

- Primary goal is to convince the *supplier* that system is "fit for purpose"
  - demonstrate system delivers its specified
    - functionality
    - performance
    - dependability
    - does not fail during normal use
- Usually black-box testing
  - tests derived from system specification

Release Testing

- Form of system testing
- Release testing should be performed by an independent team (IV&V)
  - determine if system meets its requirements and is good enough for external use
    - validation
- System testing performed by development team
  - discover bugs
    - verification (defect testing)

Requirements Testing

- Develop tests for each requirement
  - functional & non-functional
- Test to specification

Mentcare Example

- If a patient is known to be allergic to any medication, then prescription of that medication shall result in a warning message being issued to the system user.
- If a prescriber chooses to ignore an allergy warning, they shall provide a reason why this has been ignored.

Test Cases

- Patient record with no known allergies
  - prescribe medication with common allergies
    - check warning message is not displayed
- Patient record with a known allergy
  - prescribe medication to which the patient is allergic
    - check warning is displayed

Test Cases (cont.)

- Patient record with multiple allergies
  - prescribe each drug separately
    - check correct warning for each drug is displayed
  - prescribe two drugs to which the patient is allergic
    - check two warnings are correctly displayed
- Prescribe a drug that displays a warning and overrule the warning
  - check user is required to provide information explaining why the warning was overruled

Scenario Testing

- Describe typical usage scenario
- Test emergent behaviour

Mentcare Scenario

George is a nurse who specialises in mental healthcare. One of his responsibilities is to visit patients at home to check that their treatment is effective and that they are not suffering from medication side effects.

On a day for home visits, George logs into the Mentcare system and uses it to print his schedule of home visits for that day, along with summary information about the patients to be visited. He requests that the records for these patients be downloaded to his iPad. He is prompted for his key phrase to encrypt the records on the iPad.

Mentcare Scenario

One of the patients that he visits is Jim, who is being treated with medication for depression. Jim feels that the medication is helping him but believes that it has the side effect of keeping him awake at night. George looks up Jim's record and is prompted for his key phrase to decrypt the record. He checks the drug prescribed and queries its side effects. Sleeplessness is a known side effect so he notes the problem in Jim's record and suggests that he visits the clinic to have his medication changed. Jim agrees so George looks up who is Jim's treating doctor. George opens the doctor's schedule and arranges an appointment for Jim. George makes a note to call and check on Jim two weeks after the appointment. George ends the consultation and the system re-encrypts Jim's record.

Patient records are synced back to Mentcare automatically. The system generates a daily call list for George of those patients who he has to contact for follow-up information.

Features Tested

- Authentication
- Downloading and uploading of specified patient records
- Home visit scheduling
- Encryption and decryption of patient records on a mobile device
- Record retrieval and modification
- Appointment scheduling
- Links with drug database that maintains side-effect information
- Call prompting

Use Case Testing

- Use cases describe functional requirements
  - need to be validated during release testing
- Generate test scripts and scenarios from use case descriptions
  - each sequence of events should generate *at least* one test script
    - typical and alternatives
  - pre & post conditions must be tested
- Test scripts need to be traceable to their originating use case sequence of events

ATM Withdraw Test Script

Sequence of Events	Typical
Success Criteria	Nominated amount of cash is provided and withdrawn from selected account.
Pre-Requisites	Customer has entered their bank card and PIN to be identified.
Known Issues	Attempting to overdraw an account will not be caught in current version of system.

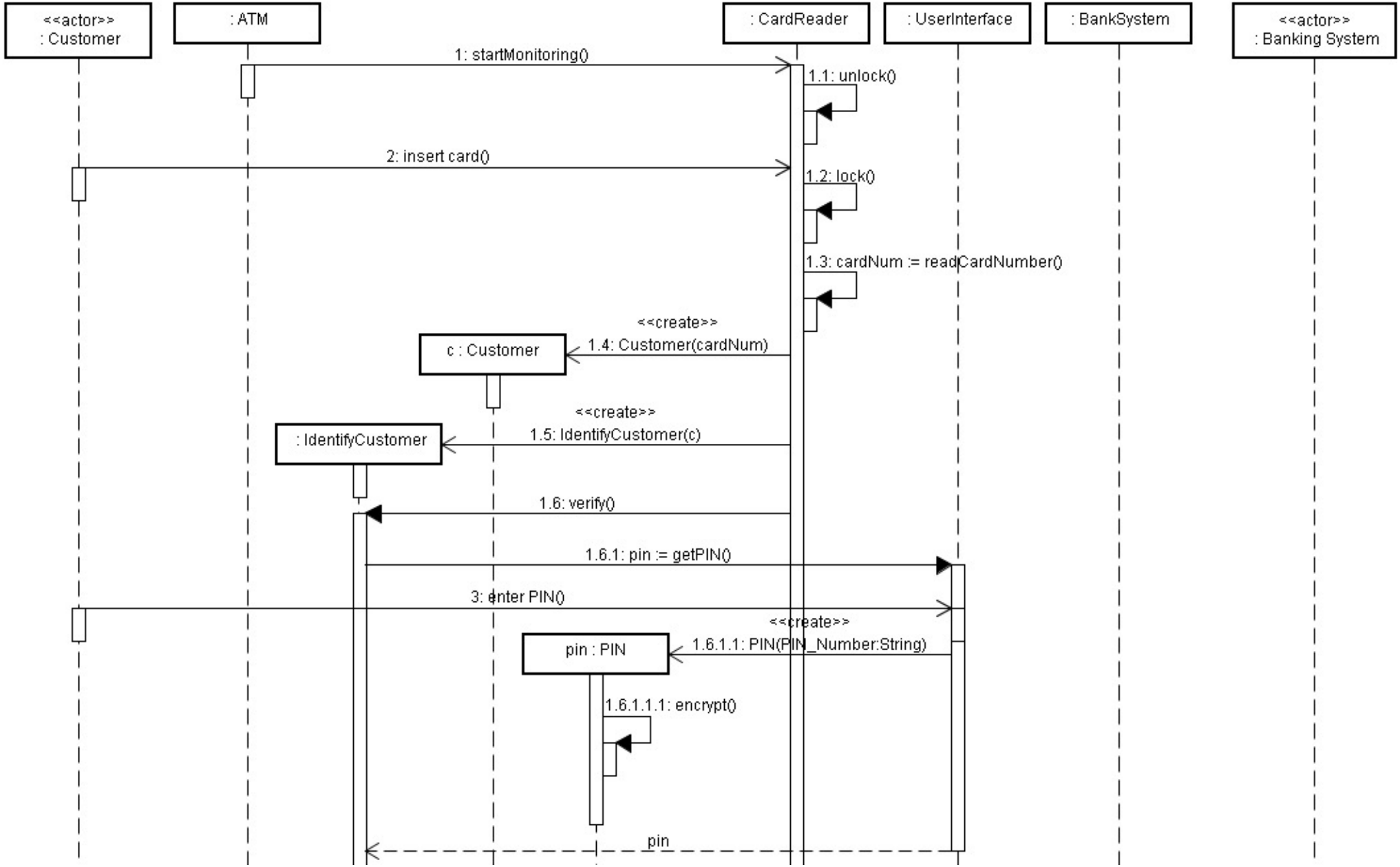
Test Script Steps

Step #	Execution Steps	Expected Result	Test Data	Comments	Pass / Fail
1	Select withdraw option	ATM requests account from which to withdraw			Pass
2	Select savings account	ATM requests amount to be withdrawn		Layout of accounts makes it difficult to determine how to select an account.	Pass
3	Enter amount to withdraw	ATM ejects card, dispenses $50, prints receipt	Amount = $50		Pass

Use Case Testing & Sequence Diagrams

- Sequence diagrams are usually associated with use cases
  - describe how the design model provides the functionality
- Can be used to drive integration and system testing
  - provides view of what software components to test
- Provides traceability between requirements, design, code and system tests

Identify Customer



Test Cases – Identify Cust.

- Insert card
  - accepted
- Insert valid card
  - locked
  - Customer created with card number
  - PIN request displayed
- Insert invalid card
  - error message
  - card ejected

Test Cases – Identify Cust.

- Enter PIN
  - PIN object created & data is encrypted
  - PIN attached to Customer
- …

Performance Testing

- Part of release testing
- Should reflect expected use
- Series of tests – load is steadily increased
  - until performance becomes unacceptable
- Stress testing is a form of performance testing
  - system deliberately overloaded to test failure behavior
- Use cloud / VMs to provide load
  - much easier to do now

Test Automation

- Testing should be automated
  - as much as possible
- Test scripts can be automated
  - run by a continuous integration environment
- Variety of tools available
  - GUI drivers
  - Web drivers
  - Mobile drivers
  - DB drivers
  - …

User Testing

- Essential, even with comprehensive system and release testing
- Self-hosted system
  - user's working environment affects reliability, performance, usability and robustness
    - cannot be replicated in test environment
    - reason for *vendor* popularity of SaaS
- Any system
  - human factors

Types of User Testing

- Alpha Testing
  - users test software in developer's environment
- Beta Testing
  - users test software in their environment
    - performing normal tasks
  - raise problems with developers
- Acceptance Testing
  - client tests system to decide whether it is ready to be deployed
    - primarily for bespoke systems

When is Testing Done?

- When there are no more errors?
  - How can you tell?
- Never!
  - "… burden simply shifts from you (the software engineer) to your customer."
- "… when you run out of time or money."
- When number failures per execution of the test suite falls below a threshold
  - statistically determine rate of reduction of errors and decide on a release point

Reading

- Sommerville
  - chapter 8
  - section 13.5
- Wiegers
  - sections 17.1, 17.4 & 17.5