Test Case

Software Testing

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

What is a Test Case?

Test Case

- A test that (ideally) executes a single well defined test objective (Testing Computer Software – Kaner, Faulk, Nguyen)
- A specific set of test data and associated procedures developed for a particular objectives (IEEE 729-1983)

Why write test cases?

Why write test case?

- Accountability
- Reproducibility
- Tracking
- Automation
- To find bugs
- To verify that tests are being executed correctly
- To measure test coverage

Test case Essentials?

Test case Essentials

- Tracking information
- Test case ID
- Test case description
  - Objective/Title
  - Steps
  - Test data: input/output/ default
  - Expected results
  - Observed results
  - Status: Pass/Fail/ Blocked/Skipped
  - Test environment
  - Script
  - Bug ID
  - Comments
  - …

Test case Essentials

- Test case Objective/Title
  - The most important essential
  - Gives reader a description and idea of the test
  - A good test name makes review easier
  - Easier to pass to another person, automation team
  - In many cases, may be the only part of the test case documented

Exercise 1

- Testing the scenario of moving a file from folder A to folder B

Exercise 1

- Testing the scenario of moving a file from folder A to folder B
  - Trying to move the file when it is Open
  - You do not have the security rights to paste the file in Folder B
  - Folder B is on a shared drive and storage capacity is full
  - Folder B already has a file with the same name
  - ...

Exercise 2

- Testing the inserting of a record to a table

Exercise 2

- Testing the inserting of a record to a table
  - “Validate that you can insert an entry”
  - “Validate that insertion fails if entry already present”
  - “Validate that insertion fails if table already full”
  - “Validate that you can insert an entry to an empty table (initial)”

Exercise 3: ATM Example

Business Requirements:

- - “ATM must do withdrawals”
- - “Withdrawals are between $20-$300”
- - “Withdrawals are in $20 multiples”

Group Exercise!

- 1. Limit the scope to these 3 requirements.
- 2. What will you validate (test for)?
- 3. Are there any implied requirements that may not be written out?

Test Requirements

- “Validate that a withdrawal option is available”
- "Validate that a withdrawal of a multiple of $20, between $20-$300 can be done"
- "Validate that <$20 is not allowed"
- "Validate that >$300 is not allowed"
- "Validate that $20 multiples >$300 is not allowed"
- "Validate that non-$20 multiples between $20-$300 not allowed"
- "Validate strange numeric amounts/combinations not allowed (all zero's, all 9's, 20.0000)"
- “Validate that the withdrawal received is equal to the amount requested”
- "Validate that a valid withdrawal amount must be below the account balance”

Test Scenarios/Cases for - "Validate that a withdrawal of a multiple of $20, between $20-$300 can be done"

Case #	P/F	$ entered	Expected Results	Actual Results
WD01	Pass	20	$20 withdrawn	
WD02	Pass	40	$40 withdrawn	
WD03	Pass	60	$60 withdrawn	
WD04	Pass	80	$80 withdrawn	
WD05	Pass	100	$100 withdrawn	
:	:	:	:	
WD13	Pass	260	$260 withdrawn	
WD14	Pass	280	$280 withdrawn	
WD15	Pass	300	$300 withdrawn	

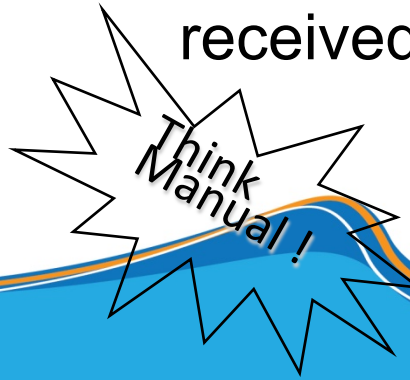
Test Procedure & Script for previous example

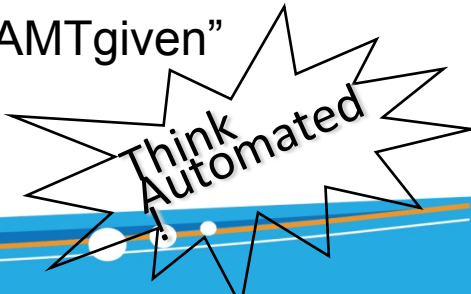
Procedure:

- Step 1: Insert Card
- Step 2: Enter PIN
- Step 3: Select Withdraw option
- Step 4: Enter dollar amount
- Step 5: Validate amount received

Script: (in pseudo-code)

- Do until EOF ‘until end of data file
- Input data record
- Senddata CARDINFO to “Cardfield”
- Senddata “Enter”
- Senddata PIN to “PINFfield”
- Senddata “Enter”
- Senddata “W” to “SelectionField”
- Senddata AMOUNT to “DollarField”
- Senddata “Enter”
- If ErrorMsg > 0 then print ErrorMsg
- Print “DollarAMTgiven”
- Loop





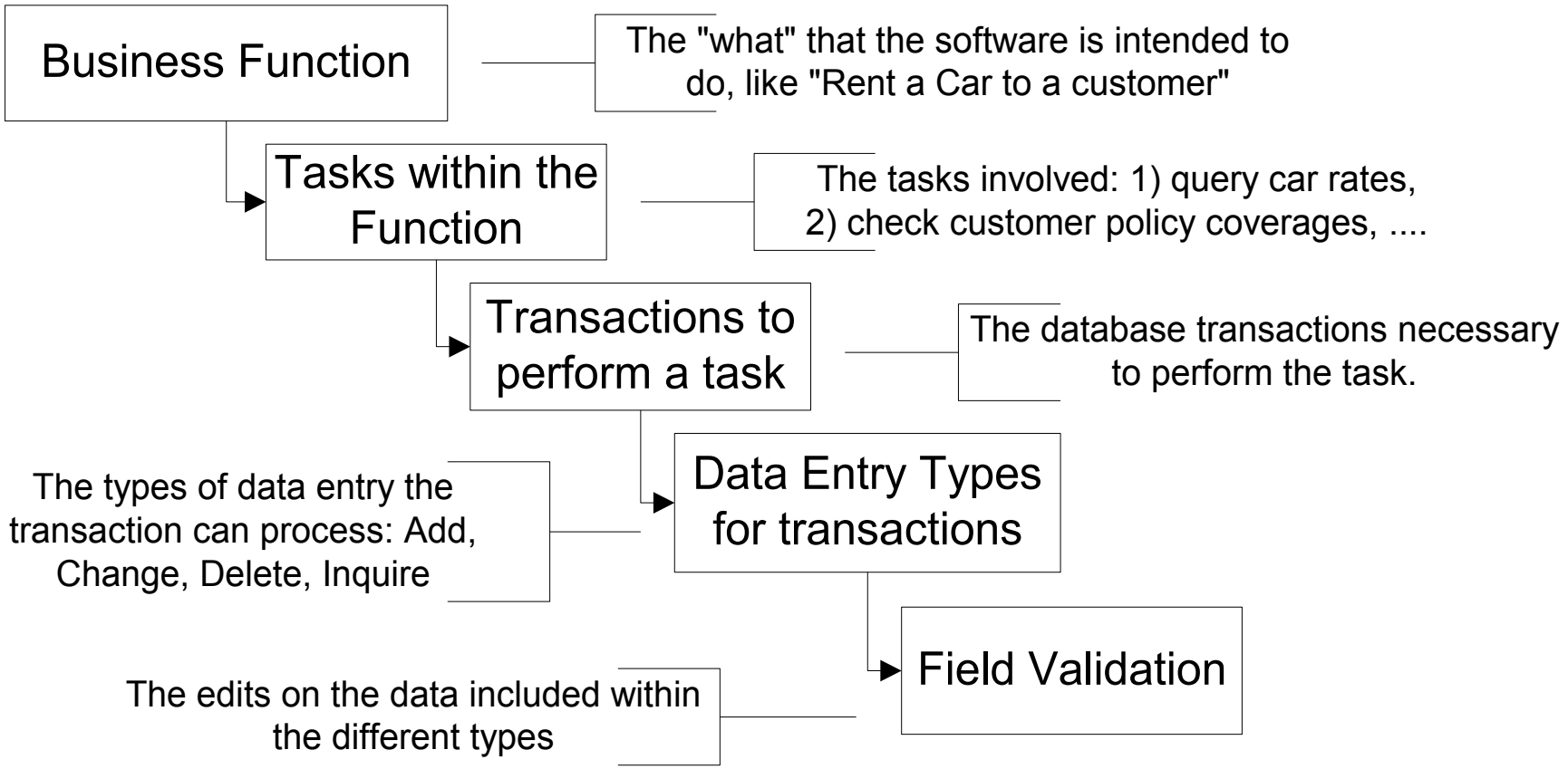
Distinguishing the types of testing….

I. Function-Based Tests
II. User Interface Tests
III. Security Tests
IV. Installation Tests
V. Configuration Tests
VI. Performance Tests (Response)
VII. Load Tests (simultaneous users, lots of small transactions)
VIII. Volume Tests (Big transactions)
IX. Stress Tests (breaking point: memory, resources)
X. Resource Usage Tests
XI. Documentation Tests
XII. Compatibility Tests
XIII. Recovery Tests
XIV. Serviceability Tests

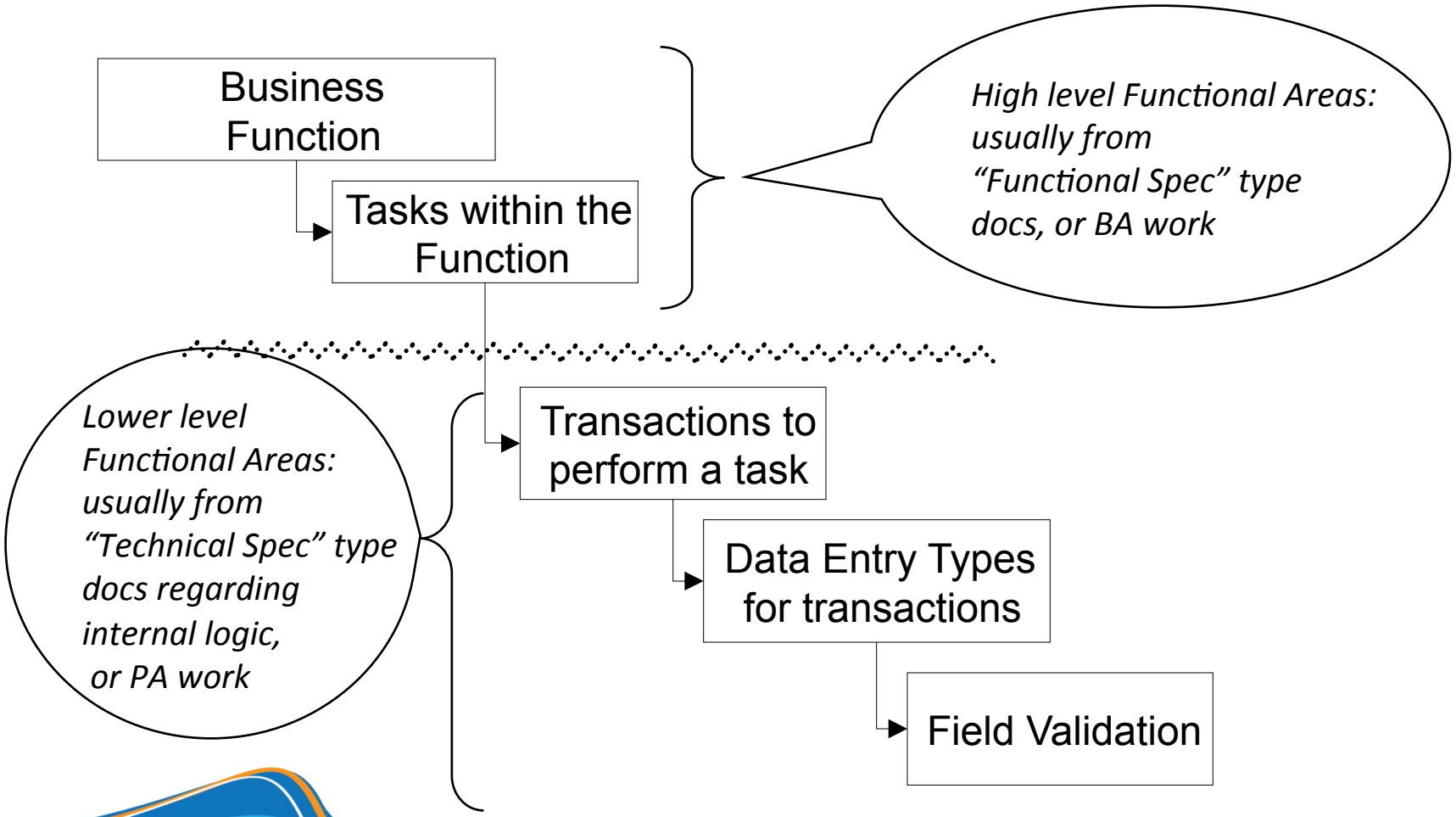
and others…

**III - XIV are all "Systems-based tests"*

Test Requirement Decomposition



Test Requirement Decomposition



Example: Rental Car Application

1. Validate that a Rental can occur.
    1.1 Check Customer policy coverage
    1.2 Query Car availability
    1.3 Query Car rates
    1.4 Open a Rental ticket
        1.4.1 Validate that a customer record can be entered
        1.4.2 Validate that credit card approval is obtained
        1.4.3 Validate that status on the car record is changed from “waiting” to “rented”
2. Billing Function
3. Reservation Function

Let’s look at the lower levels for this one

Then we’ll try it on this one

Example: Rental Car Application

1. Validate that a Rental can occur.
    1.4 Open a Rental ticket
        1.4.1 Validate that a customer record can be entered
            1.4.1.1 Validate that a new customer can be added to the customer table
                1.4.1.1.1 Validate that the first name is all alpha
                1.4.1.1.2 Validate that the age is > 21.
                1.4.1.1.3 Validate that the phone number is numeric
                1.4.1.1.4 Validate area code is an existing area code number.
            1.4.1.2 Validate changing an existing customer

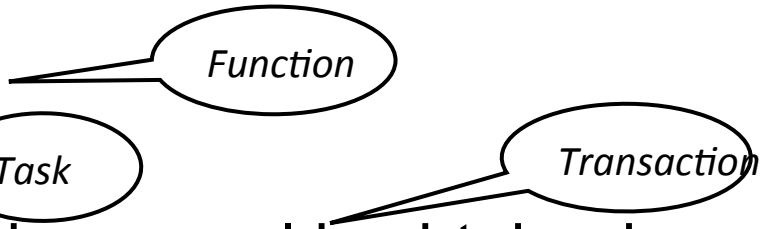
Example: Rental Car Application

1. Validate that a Rental can occur.
   1.4 Open a Rental ticket
      1.4.2 Validate that credit card approval is obtained
         …fill in the lower level test requirements!
         First, Identify any sub-areas (further tasks, or even
            separate transactions within this)
         Then, Identify the lowest level field validation test
            requirements (think about what is typically
            involved with credit card authorizations)

What did you come up with?

1. Validate that a Rental can occur.
    1.4 Open a Rental ticket
        1.4.2 Validate that credit card approval is obtained

Possible Test Requirements...



1. Validate that a Rental can occur.
   1.4 Open a Rental ticket
      1.4.2 Validate that credit card approval is obtained
         1.4.2.1 Validate the expiration date is a valid future date
         1.4.2.2 Validate the expiration date is not within 1 month of
            expiring.
         1.4.2.3 Validate that the CC# is 12 digits
         1.4.2.4 Validate that the $ amount is <= credit balance available
         1.4.2.5 Validate that an authorization # is received.

What is a Good Test case?

- Accurate – tests what it is designed to test
- Economical – no unnecessary steps
- Repeatable, reusable – keep going on
- Traceable – to a requirement
- Appropriate – for test environment
- Self standing – independent of the writer
- Self cleaning – picks up after itself

NOT so-good test cases

- Leaves it up to the user to find test data
- Gives very high level instructions that leave too much room for “artistic interpretation”
- Does not consider the Tester’s experience
- Leaves out follow-up verification steps which make it difficult to determine Pass or Fail criteria
- Too complex (test multiple conditions)
- Redundant with other test cases

Question & Homework