MIS 5203
Systems & Infrastructure Lifecycle Management 1

Week 8
October 21, 2013
Study Objectives

• Software Development contd.
• Quality Assurance
• Testing
3GL Programming Language

Phase 1: Feasibility
Phase 2: Requirements
Phase 3: Design/Selection
Phase 4: Development/Configuration
Phase 5: Implementation
Phase 6: Post Implementation

How Does the Program Work?

1. Write the Code
   - One or more files

2. Compile the Code
   - Compiled code

3. Link the code
   - Linked code with other libraries

4. Create Executable
   - This is the code we deploy

5. Run the Code
   Executable code can run
Debugging Example Using Visual Studio IDE /Class Demo
Debugging

- What’s Debugging?
- Allows “bugs” or program defects to be easily found and removed
- Concept of “break point” “watch point” “step in” “step over”
- Can see the flow of control during “run time”
- Can see memory dumps, which provides picture of actual memory at run time
What’s an IDE (Integrated Development Environment)?

IDE allows writing code, compiling, building, debugging, etc. using one tool.

Examples: Visual Studio, Eclipse, NetBean.

Can support multiple languages.

Can pre-generate Skelton code for many templates.

Adds to productivity by showing programming errors and providing prompts while writing code.

Creates both debug and deployable code and solution.
Program Design

• Designing of the Program so that they are structured well
• Multiple routines, procedures, classes need to be written. Many times a procedure calls another to do it’s work
• Would like to avoid hotchpotch, where multiple programs call each other, which makes understanding of the control and process flow difficult
• Better written program follow the principle of “more cohesion” and “less coupling”
Logging

• Allows different level of transaction and program execution logs

• The level of the logging could be one of the following with increased level of logging
  – Fatal
  – Error
  – Informational
  – Debug

• Logging can be turned on and off in production by configuration changes, which helps in troubleshooting
Unit Testing

• Developers use unit testing to test the code that they build
• This testing focuses on the each code component that is being written or changed
• Unit testing may not cover all the functional scenarios
Code Profiling

• Code profiling is used to check the efficiencies of the code
• Memory leak or performance of the code can be found by profiling
• Commercial tools such as Jetbrain, Fortify, Checkstyle
What advantages of the integrated development environment (IDE) would have compared to standalone development?

A. It prevents errors during Design (SDLC phase 3)
B. It eliminates most of the work needed during the requirement phase (SDLC phase 2)
C. It makes programming simple and helps debug program code
D. It eliminates the testing needs (SDLC phase 4)
Quality Assurance (QA)

- What’s QA role in Software Development?
  
A. To review the deliverables and results from each SDLC phase to ensure that the particular SDLC phase conforms to the Requirements

B. To ensure participation from all the stakeholders on the development of the application development standards

C. To ensure compliance on the agreed SDLC methodology

D. To review and evaluate the project at the regular milestones

E. To make sure the development and Test (non-production) environments exist and maintained

F. Regular reporting and communication on the QA activities
What’s true for Testing?

A. Testing ensures that code meets the Requirements expectation.
B. Testing ensures that the incremental functionality is provided by the software (Progression).
C. Testing ensures that the existing functionality continue to work (regression).
D. Testing ensures that the system meets performance expectation.
E. Testing ensures that when different components of the system are put together, the system as a whole continue to function.
F. Testing ensures when a software is released to production, it behaves the same way it had behaved in the lower environments.
G. Testing ensures that Project deliverables are on time.
H. Testing ensures that the budget planned for the project is met.
Types of Testing at Different Phases of SDLC (Week 4)

V-Model is a variant of water-fall model

Adapted from “CISA Review Manual 2013” ISACA, page 156
Software Testing Strategy

Constitutes

- Goal of the testing
- Types of testing to be conducted (for example, progression, regression, performance, etc.)
- Roles and responsibilities of the test teams and teams supporting testing
- Entrance and Exit criteria (for example, no Sev 1 or 2 open defects)
- Tools used to track Test Plan, Execution, and Defects (for example, Quality Center)
- Communication
- Artifacts (Test results, testing metrics etc.)
Elements of Testing

1. Test Plan
   - Functional Description of the Test Cases
   - Traceability helps keep track of the requirements are being tested
   - Typically one to many relationship between Requirements and Test Cases
   - “Bottom Up” (more common on large projects) or “Top Down” approach

2. Test Environment and Data Preparation
   - Ensure the test data is planned available to meet testing needs
   - Data often need to be stubbed for privacy reasons
   - Test environment is isolated from production

3. Test Execution
   - Running Test Cases and Reporting the status and test defects

4. Defects Management
   - Tracking of the defects
   - Fix of the defects
## Develop Test Plan for The Web Ordering Example (week 5) – Class Group Exercise 15 minutes

<table>
<thead>
<tr>
<th>No.</th>
<th>Requirement</th>
<th>Type</th>
<th>Priority</th>
<th>Must Have?</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Provide ability to collect name and address from the end users</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1</td>
<td>`fields</td>
<td>Functional</td>
<td>High</td>
<td>Must-have</td>
<td></td>
</tr>
<tr>
<td>1.2</td>
<td>Provide ability to enter address, which would have Street number, Street name, city, and zip code in separate input fields</td>
<td>Functional</td>
<td>High</td>
<td>Must-have</td>
<td></td>
</tr>
<tr>
<td>1.2.1</td>
<td>An address must have street name and city</td>
<td>Functional</td>
<td>Medium</td>
<td>Must-have</td>
<td></td>
</tr>
<tr>
<td>1.2.2</td>
<td>Zip code must be 5 character numbers</td>
<td>Functional</td>
<td>Low</td>
<td>Nice-to-have</td>
<td></td>
</tr>
<tr>
<td>1.3</td>
<td>If user clicks on “submit” button without entering name and address fields, prompt user with “you must enter your name and address to continue”</td>
<td>Functional</td>
<td>Medium</td>
<td>Must-have</td>
<td></td>
</tr>
<tr>
<td>1.4</td>
<td>If a user is a returning user, pre-populate the existing user name and address</td>
<td>Functional</td>
<td>Medium</td>
<td>Nice-to-have</td>
<td></td>
</tr>
<tr>
<td>1.5</td>
<td>Provide ability to store the name and address fields as part of the customer records</td>
<td>Functional</td>
<td>High</td>
<td>Must-have</td>
<td></td>
</tr>
<tr>
<td>1.6</td>
<td>Provide ability to submit the name and address page in less than 1 seconds</td>
<td>Performance</td>
<td>Medium</td>
<td>Nice-to-have</td>
<td></td>
</tr>
<tr>
<td>1.7</td>
<td>Provide ability to report on for manual re-entry by Support team, if the customer submits the name and address field and data is not successfully written in customer records</td>
<td>Operational</td>
<td>High</td>
<td>Must-have</td>
<td>Address part of reporting</td>
</tr>
<tr>
<td>1.8</td>
<td>The systems should be able to handle 2 million concurrent transaction with no degradation in performance</td>
<td>Performance</td>
<td>Medium</td>
<td>Must-have</td>
<td>At least 1 Million</td>
</tr>
</tbody>
</table>

---

**Vasant Kumar**

Master of IT Auditing & Cyber Security
Develop Test Plan for The Web Ordering Example

<table>
<thead>
<tr>
<th>TC</th>
<th>Traceability</th>
<th>TC Name</th>
<th>TC Description</th>
<th>Step Name</th>
<th>Step Description</th>
<th>Expected Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.1 Func_First_Last Name</td>
<td>User can enter first and last name successfully in the Personal Information screen</td>
<td>Step 1</td>
<td>Precondition - User must have clicked on &quot;order&quot; button on the home page and user is not a pre-registered user</td>
<td>Personal Information screen pops up</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Step 2</td>
<td>Check &quot;editability&quot; and initial data in first and last names fields</td>
<td>Both First and Last Name fields are editable, Both fields are empty</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Step 3</td>
<td>Enter first name - &quot;John&quot;</td>
<td>First name displays &quot;John&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Enter last name - &quot;Doe&quot;</td>
<td>Last name displays &quot;Doe&quot;</td>
</tr>
<tr>
<td>2</td>
<td>Func_First_Last</td>
<td>User should not be able to enter special characters and numerals first and last name successfully in the Personal Information screen (negative test)</td>
<td>Step 1</td>
<td>Precondition - User must have clicked on &quot;order&quot; button on the home page and user is not a pre-registered user</td>
<td>Personal Information screen pops up</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.1 Name_No_Splchar_or_numbers</td>
<td></td>
<td></td>
<td>Step 2</td>
<td>Check &quot;editability&quot; and initial data in first and last names fields</td>
<td>Both First and Last Name fields are editable, Both fields are empty</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Step 3</td>
<td>Enter first name - &quot;John123&quot;</td>
<td>First name displays &quot;John&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Enter last name - &quot;$$@Doe&amp;&quot;</td>
<td>Last name displays &quot;Doe&quot;</td>
</tr>
<tr>
<td>3</td>
<td>1.2 Func_custaddress_after_name</td>
<td>User should be able to enter address, which would have Street number, Street name, city, and zip code in separate input fields. This is after entering the name.</td>
<td>Step 1</td>
<td>Precondition - User must have clicked on &quot;order&quot; button on the home page and user is not a pre-registered user</td>
<td>Personal Information screen pops up</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Step 2</td>
<td>Check &quot;editability&quot; and initial data in Street number, Street name, city, and zip code in separate input fields</td>
<td>Street number, Street name, city, and zip code in separate input fields are editable, Street number, Street name, city, and zip code in separate input fields are empty</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Step 3</td>
<td>Enter first name - &quot;John&quot;</td>
<td>First name displays &quot;John&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Enter last name - &quot;Doe&quot;</td>
<td>Last name displays &quot;Doe&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Step 4</td>
<td>Enter Street Number - 100</td>
<td>Street Number displays - &quot;100&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Enter Street Name - &quot;Main St&quot;</td>
<td>Street Name displays - &quot;Main St&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Enter City - &quot;Anytown&quot;</td>
<td>City displays - &quot;Anytown&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Enter Zip Code - 12345</td>
<td>Zip Code displays - 12345</td>
</tr>
</tbody>
</table>

Testing Classification

1. Unit Testing – Testing of Individual module or unit

2. Functional Testing – Testing functionality
   - Progression Testing: Test incremental functionality
   - Regression Testing: Test Existing functionality to ensure they continue to work, and a new problem has not been introduced
   - Sometimes can be part of Integration, Unit, or Systems Testing

3. Integration Testing – Interface between the systems is tested
Testing Classification

4. Systems Testing – Complete System is tested. It also includes:
   • Recovery Testing: system’s ability to recover from a software or hardware failure
   • Security Testing: Access control, security risks within the program
   • Performance Testing:
     ▪ Load Testing: Test the system at peak production simulated load
     ▪ Volume Testing: Test systems performance at incremental volume
     ▪ Stress Testing: Testing “breaking point”

5. Acceptance Testing
   • UAT (User Acceptance Testing): User Acceptance
   • QAT (Quality Assurance Testing): Final Regression
Other Testing Types

White Box and Black Box Testing

• **White Box Testing**
  – Testing program logic
  – Typically by developers during unit-testing phase

• **Black Box Testing**
  – Functional testing without going at the low level (program level)
  – Part of progression or regression testing
  – Can be during Integration or Systems Test phase

Alpha and Beta Testing

• **Alpha Testing**
  – Alpha version is early version of the application tested by “internal” users

• **Beta Testing**
  – Beta version of the application used by selected users

• Used to get users feedback before releasing the application widely
Other Testing Types

Pilot Testing
  • Testing certain features of the application
  • Proof of concept

Parallel Testing
  • Feeding test data to two systems, typically one existing and other new
  • Validates the data input and output through two systems

Sociability Testing
  • Ensure the new or modified system can operate in the target environment without affecting other systems in the ecosystem
Example of Testing Tools

• Quality Center (HP) – For Test Case Management, Execution, Defects Management
• ServiceTest, QuickTestPro (HP) – For Service and UI Testing Automation
• LoadRunner (HP) – For Performance Testing
Question

Which of the following applies to regression testing?

A. Its purpose is to test internal program logic
B. It’s a means to test white-box functions
C. It ensures that the change did not introduce a new problem
D. It’s a means to test black-box functions
Upcoming Assignments/Tests

1. Individual Case Study -2 (Requirements and Use Case): Mon 10/21 before the class
3. Group Case Study -2 (Requirements): Mon 11/4 before the class

Questions?
Summary of Today’s Class

- Unit Testing
- IDE, Debugging, Programming Techniques, Logging, Profiling
- QA
- Testing
- Focus of the Next Class and Reading
- Questions