

# Web Application Development

## Bugs

# What is a Bug?

- In software a bug refers to an error in the application which is created.
- A programmer while designing and building the software can make mistakes or an error.
- These mistakes or errors mean that there are flaws in the software.

# Other Definitions for Bugs

- When the actual result deviates from the expected result while testing a software application or product then it results in what is known as a bug.
- Any deviation from the specification mentioned in the product functional specification document is considered a bug.
- When the result of the software application or product does not meet with the end user expectations or the software requirements then it is considered a Bug.

# Other terms used for Bugs

- Mistake
- Anomaly
- Fault
- Failure
- Error
- Exception
- Defect
- Incident
- Side effect

# An Example of a Bug

```
var break = 1;  
alert("Value of break is: " +break);
```

Note: Have used a keyword (see Javascript Basics lecture for list of keywords) as a variable name. This would result in an error which indicates that a bug is present.

See BugSample.html in Internet Explorer Development Tool, or use some other browser development tool.

# Bug Detection

- Previous slide showed a bug which is easily detectable and fixed. This type of bug is usually fixed by the developer while they are still in the process of writing the code.
- Programmers usually will detect this kind of bug fairly quickly as it will be reported by the development tool they are using, debugger, or by running a quick test to determine the code does what is expected of it. This is known as unit testing.
- Other types of bugs are not as obvious and can be missed by the developer. Once the programmer is satisfied that no defects are present or that the software is in a stable state, they will release it so that testers can perform functional testing.

# Bug Fix Iteration

- Functional testing is performed and if any bugs are detected these are reported.
- They are then prioritized and/or given a severity classification.
- The bug is then assigned to a programmer, usually the original programmer if they are available.
- The bug is fixed and the fix is released for further functional testing.
- If the tester(s) are satisfied the bug is fixed they will then close the bug. If not, then the bug is assigned back to the programmer(iteration).

# Managing Software Testing

- There are various tools available for managing the testing of software.
- Some examples of well known tools are:
  - Jira
  - Bugzilla
  - Test Director
- Various other development tools include a task manager which can be used to manage testing.
- The typical process for managing testing is described on the following slide.



# Managing Software Testing

- From software requirements, test cases are defined.
- A simple example. A project exists to create a web page with the following requirements:
  - The page contains two fields which allow a user to input numbers
  - The page contains a button, which when clicked will check if the two fields are populated with numbers.
  - If the two fields are populated with valid numbers then these numbers will be added together with the result shown in a message written to the web page.
  - If the two fields are not populated with valid numbers an alert will appear informing the user that only valid numbers are accepted as inputs to the fields.

# Managing Software Testing

- From these requirements a number of test cases exist.
- One test case is to ensure the fields contain valid numbers when the button is clicked.
- This test case would be logged on jira for example, and assigned to a tester. It would probably be logged as a task and include some steps for the tester to follow to confirm an expected result or expected results. The expected result(s) are also included in the task.
- The tester would then follow the steps and confirm if expected result(s) achieved.

# Managing Software Testing

- If the expected result(s) obtained, then the tester will update the status of the task (also commonly referred to as tickets), to a value indicating test was a success, and assign it back to a manager. Sometimes just updating the status will update who the task is assigned to automatically.
- If the expected result(s) not obtained, then the tester will update the status of the task to a value indicating the test fail. The tester would include some comments on their observations and provide steps to recreate the bug if necessary. The task is then assigned back to the developer to fix.
- Sometimes the test tools are configured to automate an email or sms or im to the person the task is newly assigned to in order to prompt them to have a look at it.

# Managing Software Testing

- Note: From project to project and company to company the steps outlined on the previous slides relating to creating, assigning, updating test cases may vary slightly. For example the test cases maybe in a document which the tester follows and they may only create a task on a tool like jira if a bug is encountered. The statuses used, priorities, automating assignment of tasks and email etc ... could differ. It depends on the tools, processes and people involved.

# Bug Priority / Severity

- When a tester is updating the status of a test case to fail, they will usually put in a value for the severity and priority of the bug.
- This helps the developer to determine in what order the bugs need to be addressed.
- One bug maybe more severe than another bug so should be looked at before any lesser severe ones. Priority also plays a role, and sometimes, depending on the software being developed, or the project, maybe the only value considered when determining the order in which the developer should address the defects.
- A diagram with a sample of the severity and priority levels is shown on the next slide.

# Bug Priority / Severity

<b>P1</b>	• Critical	<b>S1</b>	• Critical
<b>P2</b>	• High	<b>S2</b>	• Major
<b>P3</b>	• Medium	<b>S3</b>	• Moderate
<b>P4</b>	• Low	<b>S4</b>	• Low

# Bug Priority / Severity

- Obviously a bug which is classified with severity level of S1 and a priority level of P1 is the most critical bug and must be addressed before all others.
- The levels shown in the diagram on the previous slide are what is typical. However, different projects or companies may have different levels defined.
- Note: Lecturer will give some examples in the class of what determines what level.

# Bug Priority / Severity

- The priority and severity levels can also determine how fit software is for release, in terms of the volume of bugs.
- Releases will typically go ahead with some known bugs due to time and cost restraints.
- For example, a sign off to release a project might be agreed if there are:
  - No S1 or S2 bugs
  - No S3 with priority of P1 or P2.
  - No more than five S3 bugs with a priority of P3.
  - Any number of S3 bugs with a priority of P4.
  - Any number of S4 bugs.



# Consequences of Undetected Bugs

- Bug could lead to a security breach which may result in loss of customer data, or money. The kind of publicity this generates can have a negative impact on a company.
- Peoples lives being endangered.
- Poor performance of product which leads to a lot of negative feelings about the product which is bad for companies image. May result in the loss of future sales.

# Bug Fixing Cost

Bug fixing cost

