|  |  |
| --- | --- |
| **Course Title/ Code** | **Agile Technologies (CSW317B)** |
| **Course Type** | Core (Departmental) |
| **Course Nature** | Workshop |
| **L-T-P-O Structure** | (0-0-3-0) |
| **Objectives** | To introduce the practical applications of agile software development tools. |

|  |  |  |
| --- | --- | --- |
| **Syllabus** | **Sections** | **Weightage** |
| A | 25% |
| B | 25% |
| C | 25% |
| D | 25% |
| **TOTAL** | **100%** |

**Section-A**

**Agile Programming Tools**: UNIX, Eclipse, Git, jUnit,Processes: Stories, End-to-end Testing, Unit Testing, TDD, Refactoring. Reading: Scrum, Extreme Programming, Features Driven Development, Lean Software Development

**Section-B**

**Agile DesignTools**: Use Cases, PowerPoint Design, Requirements/Story Extraction, Test Case Management. Processes: Use cases to Design, Design to Backlog, Backlog to Tasks, End to End Testing, Estimation

**Section-C**

**Agile Architecture/Design and Continuous IntegrationTools**: Jenkins, Maven, Cucumber. Processes: Scrum, Architecture, Iterative Refinement, Agile Design.

# 

**Section-D**

**Agile Process ManagementTools**: Agilefant. Processes: Agile Process Management, Estimation, Burn-down, Release Planning, Multi-team coordination, Distributed teams

**List of Experiments**

1. Test Driven Development on Eclipse using jUnit
2. Software Configuration Management using Git
3. Backlog development from use cases and user interface designs using Specification by Example
4. End-to-end/Acceptance tests using Cucumber
5. Continuous Integration using Jenkins
6. Agile Process Management using Agilefant.

**Text Books:**

1. Robert C. Martin, Clean Code: A Handbook of Agile Software Craftsmanship, available at http://www.it-ebooks.info/book/1441/.

**Reference Links:**

1. Agile software development, http://en.wikipedia.org/wiki/Agile\_software\_development
2. Scrum, http://en.wikipedia.org/wiki/Scrum\_%28software\_development%29
3. Extreme Programming, http://en.wikipedia.org/wiki/Extreme\_programming
4. Feature-drive development, http://en.wikipedia.org/wiki/Feature-driven\_development
5. Lean Software development, http://en.wikipedia.org/wiki/Lean\_software\_development
6. Test-driven development, http://en.wikipedia.org/wiki/Test-driven\_development
7. Unit testing, http://en.wikipedia.org/wiki/Unit\_testing
8. Specification by example, http://en.wikipedia.org/wiki/Specification\_by\_example
9. Behavior-driven development, http://en.wikipedia.org/wiki/Behavior-driven\_development
10. Code refactoring, http://en.wikipedia.org/wiki/Code\_refactoring
11. User Experience, http://en.wikipedia.org/wiki/User\_experience

**Tool Web Sites:**

1. Ubuntu, http://www.ubuntu.com/desktop
2. Eclipse, https://eclipse.org/users/
3. jUnit, http://junit.org/
4. Git, http://git-scm.com/
5. Jenkins, https://jenkins-ci.org/
6. Ant, http://ant.apache.org/
7. Maven, https://maven.apache.org/
8. Cucumber, https://cukes.info/
9. Fitnesse, http://www.fitnesse.org/
10. Agilefant, http://agilefant.com/