Semester	IV		
Course Code:	IT4406		
Course Name:	Agile Software Development		
Credit Value:	4 (3L + 1P)		
Core/Optional	Core		
Hourly Breakdown	Theory	Practical	Independent Learning
	45 hrs.	30 hrs.	125 hrs.

Introduction:

- This course focuses on the software development process using the Agile approach. Students will master Agile approaches primarily Scrum and an introduction other approaches such as Extreme Programming, Lean, and Kanban. The lifecycle and practices of Scrum will be covered in detail.

Intended Learning Outcomes:

After successful completion of this course students will be able to:

- Obtain a firm foundation on Agile concepts and methodologies.
- Acquire understanding of the practices and application of Agile practices Scrum.
- Learn how to apply the Agile framework in software Development Projects.
- Learn fundament concepts of Extreme Programming.

Minor Modifications:

When minor modifications are made to this syllabus, those will be reflected in the Virtual Learning Environment (VLE) and the latest version can be downloaded from the relevant course page of VLE. Please inform your suggestions and comments through the VLE. http://vle.bit.lk

Course Content: (Main Topics, Sub topics)

Topic	**Theory (Hours)**	**Practical (Hours.)**
1. Introduction to Agile Software Development	1	-
2. Agile Principles	3	-
3. Introduction to Scrum	2	2
4. Core Concepts in Scrum	6	4
5. Scrum Roles	6	4
6. Scrum Planning	10	10
7. Sprints	8	5
8. Introduction to Testing in SCRUM	6	5
9. Extreme Programming (XP)	3	-
Total	**45**	**30**

1. **Introduction to Agile Software Development (01 hour)**

   1.1. Rational of Agile [Ref 2: pages 3-7]

   1.2. How to use Agile [Ref 2: pages 9-12]

   1.3. Agile Manifesto [ref URL1, Ref 3 pages 369-373]

   1.4. Scrum, Lean, Kanban, Extreme Programming [Ref 1: page 1, Ref URL2, Ref 2: page 15]

2. **Agile Principles (03 hours)**

   2.1. Introduction [Ref 1: page 29]

   2.2. Variability and Uncertainty [Ref 1: pages 32-36]

   2.3. Prediction and Adaptation [Ref 1: pages 37-43]

   2.4. Validated Learning [Ref 1: pages 44-46]

   2.5. Work in Process (WIP) [Ref 1: pages 48-52]

   2.6. Progress [Ref 1: pages 54-55]

   2.7. Performance [Ref 1: pages 56-57]

   2.8. Comparison Summary of Plan-Driven and Agile Principles [Ref 1: page 59]

3. Introduction to Scrum (02 hour)

3.1. What is Scrum? [Ref 1: page 1]

3.2. History of Scrum [Ref 1: page 3]

3.3. Advantages of Scrum [Ref 1: page 4]

3.4. Genomica Results [Ref 1: page 4]

3.5. Use of Scrum [Ref 1: pages 5-9]

4. Core Concepts in Scrum (06 hours)

4.1. Scrum Framework

4.1.1. Introduction [Ref 1: page 13]

4.1.2. Scrum Roles [Ref 1: page 14]

4.1.2.1. Product Owner [Ref 1: page 15]

4.1.2.2. Scrum Master [Ref 1: page 16]

4.1.2.3. Development Team [Ref 1: page 16]

4.1.3. Scrum Activities and Artifacts [Ref 1: page 16]

4.1.3.1. Product Backlog [Ref 1: page 18]

4.1.3.2. Sprints [Ref 1: page 20]

4.1.3.3. Sprint Planning [Ref 1: page 21]

4.1.3.4. Sprint Execution [Ref 1: page 23]

4.1.3.5. Daily Scrum [Ref 1: page 23]

4.1.3.6. Done [Ref 1: page 25]

4.1.3.7. Sprint Review [Ref 1: page 26]

4.1.3.8. Sprint Retrospective [Ref 1: page 27]

4.2. Sprints

4.2.1. Introduction [Ref 1: page 61]

4.2.2. Timeboxed [Ref 1: pages 62-64]

4.2.3. Short Duration [Ref 1: pages 64-66]

4.2.4. Consistent Duration [Ref 1: pages 67-68]

4.2.5. No Goal-Altering Changes [Ref 1: pages 69-72]

4.2.6. Definition of Done [Ref 1: pages 74-77]

4.3. Requirements and User Stories

4.3.1. Introduction [Ref 1: page 79]

4.3.2. Using Conversations [Ref 1: page 81]

4.3.3. Progressive Refinement [Ref 1: page 82]

4.3.4. What Are User Stories? [Ref 1: pages 83-85]

4.3.5. Level of Detail [Ref 1: page 86]

4.3.6. INVEST in Good Stories [Ref 1: pages 88-92]

4.3.7. Nonfunctional Requirements [Ref 1: page 93]

4.3.8. Knowledge-Acquisition Stories [Ref 1: page 93]

4.3.9. Gathering Stories [Ref 1: pages 95-96]

4.4. Product Backlog

4.4.1. Introduction [Ref 1: page 99]

4.4.2. Product Backlog Items [Ref 1: page 100]

4.4.3. Good Product Backlog Characteristics [Ref 1: pages 101-103]

4.4.4. Grooming [Ref 1: pages 104-106]

4.4.5. Definition of Ready [Ref 1: page 108]

4.4.6. Flow Management [Ref 1: pages 110-111]

4.4.7. Which and How Many Product Backlogs? [Ref 1: pages 112-117]

4.5. Estimation and Velocity

4.5.1. Introduction [Ref 1: page 119]

4.5.2. What and When We Estimate [Ref 1: pages 120-122]

4.5.3. PBI Estimation Concepts [Ref 1: pages 123-125]

4.5.4. PBI Estimation Units [Ref 1: page 128]

4.5.5. Planning Poker [Ref 1: pages 129-133]

4.5.6. What Is Velocity? [Ref 1: page 133]

4.5.7. Calculate a Velocity Range [Ref 1: page 134]

4.5.8. Forecasting Velocity [Ref 1: page 135]

4.5.9. Affecting Velocity [Ref 1: page 135-137]

4.5.10. Misusing Velocity [Ref 1: page 137-138]

5. Scrum Roles (06 hours)

5.1. Product Owner

5.1.1. Introduction [Ref 1: page 165]

5.1.2. Principal Responsibilities [Ref 1: pages 166-171]

5.1.3. Characteristics/Skills [Ref 1: pages 171-173]

5.1.4. A Day in the Life [Ref 1: page 174-175]

5.1.5. Who Should Be a Product Owner? [Ref 1: pages 176-180]

5.1.6. Product Owner Combined with Other Roles [Ref 1: page 181]

5.1.7. Product Owner Team [Ref 1: pages 182-183]

5.2. Scrum Master

5.2.1. Introduction [Ref 1: page 185]

5.2.2. Principal Responsibilities [Ref 1: pages 185-187]

5.2.3. Characteristics/Skills [Ref 1: pages 188-189]

5.2.4. A Day in the Life [Ref 1: page 190]

5.2.5. Fulfilling the Role [Ref 1: pages 191-192]

5.3. Development Team in Scrum

5.3.1. Introduction [Ref 1: page 195]

5.3.2. Role-Specific Teams [Ref 1: page 195]

5.3.3. Principal Responsibilities [Ref 1: pages 196-197]

5.3.4. Characteristics/Skills [Ref 1: pages 198-209]

5.4. Scrum Team Structures

5.4.1. Introduction [Ref 1: page 213]

5.4.2. Feature Teams versus Component Teams [Ref 1: page 213]

5.4.3. Multiple-Team Coordination [Ref 1: pages 218-220]

5.5. Managers

5.5.1. Introduction [Ref 1: page 225]

5.5.2. Fashioning Teams [Ref 1: pages 227-230]

5.5.3. Nurturing Teams [Ref 1: pages 231-233]

5.5.4. Aligning and Adapting the Environment [Ref 1: pages 233-234]

5.5.5. Managing Value-Creation Flow [Ref 1: pages 235-236]

5.5.6. Project Managers [Ref 1: pages 237-239]

6. Scrum Planning (10 hours)

6.1. Scrum Planning Principles [Ref 1: page 245]

6.1.1 Introduction [Ref 1: page 247]

6.1.2 “Don’t Assume We Can Get the Plans Right Up Front” [Ref 1: page 248]

6.1.3 Up-Front Planning Should Be Helpful without Being Excessive [Ref 1: page 248]

6.1.4 Keep Planning Options Open Until the Last Responsible Moment [Ref 1: page 249]

6.1.5 Focus More on Adapting and Re-planning Than on Conforming to a Plan [Ref 1: page 249]

6.1.6 Correctly Manage the Planning Inventory [Ref 1: page 251]

6.1.7 Favor Smaller and More Frequent Releases [Ref 1: page 252]

6.1.8 Plan to Learn Fast and Pivot When Necessary [Ref 1: page 254]

6.2. Multilevel Planning [Ref 1: page 257]

6.2.1 Introduction [Ref 1: page 257]

6.2.2 Portfolio Planning [Ref 1: page 259]

6.2.3 Product Planning (Envisioning) [Ref 1: pages 259-260]

6.2.4 Release Planning [Ref 1: page 261]

6.2.5 Sprint Planning [Ref 1: page 264]

6.2.6 Daily Planning [Ref 1: page 264]

6.3. Portfolio Planning [Ref 1: page 267]

6.3.1 Introduction [Ref 1: pages 267-268]

6.3.2 Scheduling Strategies [Ref 1: pages 270-274]

6.4. Envisioning (Product Planning) [Ref 1: page 287]

6.4.1 Introduction [Ref 1: pages 287-290]

6.4.2 SR4U Example [Ref 1: page 290]

6.4.3 Visioning [Ref 1: page 291]

6.4.4 High-Level Product Backlog Creation [Ref 1: page 294]

6.4.5 Product Roadmap Definition [Ref 1: page 295]

6.4.6 Other Activities [Ref 1: page 298]

6.4.7 Economically Sensible Envisioning [Ref 1: pages 299-305]

6.4.8 Closing [Ref 1: page 306]

6.5. Release Planning (Longer-Term Planning) [Ref 1: page307]

6.5.1 Introduction [Ref 1: pages 307-309]

6.5.2 Release Constraints [Ref 1: pages 311-314]

6.5.3 Grooming the Product Backlog [Ref 1: page 315]

6.5.4 Refine Minimum Releasable Features (MRFs) [Ref 1: page 316]

6.5.5 Sprint Mapping (PBI Slotting) [Ref 1: page 316]

6.5.6 Fixed-Date Release Planning [Ref 1: page 318]

6.5.7 Fixed-Scope Release Planning [Ref 1: page 323]

6.5.8 Calculating Cost [Ref 1: page 325]

6.5.9 Communicating [Ref 1: pages 326-329]

7. Sprints (08 hours) [Ref 1: page 333]

7.1. Sprint Planning [Ref 1: page 335]

7.1.1 Introduction [Ref 1: pages 335-336]

7.1.2 Approaches to Sprint Planning [Ref 1: pages 338-339]

7.1.3 Determining Capacity [Ref 1: pages 340-342]

7.1.4 Selecting Product Backlog Items [Ref 1: page 343]

7.1.5 Acquiring Confidence [Ref 1: page 344]

7.1.6 Refine the Sprint Goal [Ref 1: page 346]

7.1.7 Finalize the Commitment [Ref 1: page 346]

7.2. Sprint Execution [Ref 1: page 347]

7.2.1 Introduction [Ref 1: pages 347-348]

7.2.2 Sprint Execution Planning [Ref 1: page 349]

7.2.3 Flow Management [Ref 1: pages 349-354]

7.2.4 Daily Scrum [Ref 1: page 354]

7.2.5 Task Performance-Technical Practices [Ref 1: page 355]

7.2.6 Communicating [Ref 1: pages 356-360]

7.3. Sprint Review [Ref 1: page 363]

7.3.1 Introduction [Ref 1: page 363]

7.3.2 Participants [Ref 1: page 364]

7.3.3 Sprint Review Prework [Ref 1: pages 365-368]

7.3.4 Sprint Review Approach [Ref 1: pages 368-371]

7.3.5 Sprint Review Issues [Ref 1: pages372-373]

7.4. Sprint Retrospective [Ref 1: page 375]

7.4.1 Introduction [Ref 1: page 375]

7.4.2 Sprint Retrospective Participants [Ref 1: page 377]

7.4.3 Sprint Retrospective Prework [Ref 1: pages 378-380]

7.4.4 Sprint Retrospective Approach [Ref 1: pages 380-390]

7.4.5 Sprint Retrospective Follow Through [Ref 1: page 391]

7.4.6 Sprint Retrospective Issues [Ref 1: page 392]

****Guided Practical 01**

Tutorials on agile collaboration tools.

The practicals focus on the theoretical agile aspects in the previous sessions in the course module. The practical includes agile collaboration tools such as products by Atlassian (e.g.: Jira, Confluence, and other products) that have a free account option where students can interact with, or open-source collaboration tools such as MyCollab, OpenProject etc. However, it is advisable to select a tool that is freely accessible and widely used by the software industry.

Guide

7.5.1. Pre-requisites:

7.5.1.1. Select appropriate agile collaboration tools. (i.e., Jira [URL 3, URL 4], Confluence [URL 5])

7.5.2. Introduction to the agile collaboration tool [URL 3, URL 4, URL 5].

7.5.3. Setting up the tool: account registration, sign in, configurations etc. [URL 10]

7.5.4. Scrum using the collaboration tool. [URL 7, URL 8]

7.5.4.1. Create a scrum project

7.5.4.2. Create user stories or tasks in the backlog

7.5.4.3. Using epics in your backlog

7.5.4.4. Customizing your workflow

7.5.4.5. Create a sprint

7.5.4.6. Hold the sprint planning meeting

7.5.4.7. Start the sprint

7.5.4.8. Hold the daily standup meetings

7.5.4.9. View the Burndown Chart

7.5.4.10. View the sprint report

7.5.4.11. Hold the sprint review meeting

7.5.4.12. Hold the sprint retrospective meeting

7.5.4.13. Complete the sprint

7.5.5. Practicing Kanban using the collaboration tool [URL 6]

7.5.5.1. Creating a Kanban project

7.5.5.2. Configuring workflows

7.5.5.3. Add tasks, bugs, or user stories to the backlog

7.5.5.4. Prioritize the backlog

7.5.5.5. Select work from the backlog

7.5.5.6. Column constraints

7.5.6. Version control [URL 9]

7.5.6.1. Create a version in Jira Software

7.5.6.2. Add issues to version

7.5.6.3. Monitoring the progress of a version

7.5.6.4. Complete a version

7.5.7. Working with a collaborative knowledge sharing tool. [URL 5]

7.5.7.1. Learn about spaces

7.5.7.2. Creating a space

7.5.7.3. Customize space

7.5.7.4. Organize content

7.5.7.5. Manage users and permissions

7.5.7.6. Create a new page

7.5.7.7. Edit an existing page

7.5.7.8. Collaborate on work

7.5.8. Linking and interactions between the selected collaborative tools [URL 11]

8. Introduction to Testing in SCRUM (06 hours)

8.1. Testing in waterfall method vs Agile Testing

8.2. Agile Quality Assurance vs Traditional Quality Assurance

8.2.1 Traditional Quality Assurance Tools [Ref 2: page 172]

8.2.2 Agile Quality Assurance Tools [Ref 2: page 174]

8.2.3 Test Planning in Scrum [Ref 2: page 181]

8.3. Unit testing [Ref 2: page 56]

8.3.1 The Test First approach [Ref 2: page 74]

8.3.2 Unit Testing Frameworks [Ref 2 page 84]

8.3.3 Unit Test Management [Ref 2: page 87]

8.4. Integration Testing [Ref 2: page 94]

8.4.1 Designing Integration Test Cases [Ref 2: page 97]

8.4.2 Dependencies and Interfaces [Ref 2: page 104]

8.4.3 Integration Levels [Ref 2: page 106]

8.4.4 Continuous Integration [Ref 2: page 111]

8.4.5 Implementing CI [Ref 2: page 114]

8.4.6 Integration Test Management [Ref 2: page 119]

8.5. System Testing and Testing Nonstop [Ref 2: page 125]

8.5.1 Manual System Testing [Ref 2: page 131]

8.5.2 Automated System Testing [Ref 2: page 134]

8.5.3 Using Test First for System Testing [Ref 2: page 143]

8.5.4 Non-functional Testing [Ref 2: page 145]

8.5.5 Automated Acceptance Testing [Ref 2: page 148]

8.5.6 System Test Management [Ref 2: page 155]

****Guided Practical 02 [for 8.1.5]**

Introduction to selenium IDE [Ref 5: page 5]

- Download and install selenium IDE [Ref 5: page 5]
- Explain the basic features of Selenium IDE [Ref 5: page 7]
- Creating Selenium IDE tests [Ref 5: page 8]
    - Recording and adding command in a test

        - Saving the recorded test
        - Saving the test suite
        - Executing the recorded test
    - Script Debugging [Ref 5: page 13]
        - Finding and fixing errors in the script
        - Install Power Debugger
    - Inserting Verification Points [Ref 5: page 17]
    - Different Browser Execution [Ref 5: page 25]
    - Locators Usage [Ref 5: page 76]
        - Java Syntax and different methods (ID, Name, Tag, CSS, Xpath etc.)

9. Extreme Programming XP (03 hours) [Ref 3: page 15]

9.1 Introduction to XP [Ref 2: page 15]

9.2 XP Lifecycle [Ref 2: pages 17-22]

9.3 The XP Team [Ref 2: page 28]

9.3.1 The Whole Team [Ref 2: page 28]

9.3.2 On-Site Customers [Ref 2: page 29]

9.3.3 Programmers [Ref 2: page 33]

9.3.4 Testers [Ref 2: page 34]

9.3.5 Coaches [Ref 2: page 35]

9.3.6 Other Team Members [Ref 2: page 36]

9.3.7 The Project Community [Ref 2: page 36]

9.3.8 Filling Roles [Ref 2: page 39]

9.3.9 Team Size [Ref 2: page 39]

Teaching /Learning Methods:

You can access all learning materials and this syllabus in the VLE: http://vle.bit.lk , if you are a registered student of BIT degree program. It is especially important to participate in learning activities given in the VLE to learn this subject.

Assessment Strategy:

Continuous Assessments/Assignments:

The assignments consist of two quizzes, assignment quiz 1 (It covers the first half of the syllabus) and assignment quiz 2 (It covers the second half of the syllabus). The maximum mark for a question is 10 and the minimum mark for a question is 0 (irrespective of negative scores). The final assignment mark is calculated considering both assignments. To pass the online assignment component, students will have to obtain at least 40% for each assignment. Students are advised to complete online assignments before the given deadline. It is compulsory to pass the online assignment component to qualify to obtain the Level II Higher Diploma in IT (HDIT) certificate.

In the course, case studies/Lab sheets will be introduced, and students have to participate in the learning activities.

Final Exam:

Final examination of the course will be held at the end of the semester. The course is evaluated using a two-hour question paper which consists of 25 MCQ (1 hour) and 2 Structured Questions (1 hour).

References/ Reading Materials:

Main references

- **Ref 1:** Essential Scrum Practical Guide to the Most Popular Agile Process, Kenneth S. Rubin (First edition, 2012)
- **Ref 2:** Testing in Scrum: A Guide for Software Quality Assurance in the Agile World (First edition, 2014)
- **Ref 3:** The Art of Agile Development, James Shore and Shane Warden (First edition, 2007)

Supplementary references

- **Ref 4:** Agile and Iterative Development: A Manager's Guide by Craig Larman, Agile Software development series, Alistair Cockburn and Jim Highsmith, Series Editors
- **Ref 5**: https://www.tutorialspoint.com/selenium/selenium_tutorial.pdf
- **URL 1:** Manifesto for Agile Software Development [http://agilemanifesto.org/]
- **URL 2:** Lean Software Development [https://msdn.microsoft.com/enus/library/hh533841.aspx]
- **URL 3:** Atlassian Agile Coach: https://www.atlassian.com/agile/tutorials
- **URL 4:** Tutorials point: https://www.tutorialspoint.com/jira/index.htm

- **URL 5:** Confluence guide: https://www.atlassian.com/software/confluence/guides/get-started/set-up
- **URL 6:** Kanban Guide: https://www.atlassian.com/agile/tutorials/how-to-do-kanban-with-jira-software
- **URL 7:** Scrum with Jira: https://www.atlassian.com/agile/tutorials/how-to-do-scrum-with-jira-software
- **URL 8:** Advanced Scrum with Jira: https://www.atlassian.com/agile/tutorials/how-to-do-advanced-scrum-practices-with-jira-software
- **URL 9:** Versions in Jira: https://www.atlassian.com/agile/tutorials/versions
- **URL 10:** Getting started with Jira: https://www.atlassian.com/software/confluence/guides/expand-confluence/confluence-and-jira#integrate-confluence-and-jira-software
- **URL 11:** Using Confluence Cloud and Jira Software Cloud together: https://www.atlassian.com/software/confluence/guides/expand-confluence/confluence-and-jira#integrate-confluence-and-jira-software