Agile software development methods

Academic Year: (2019 / 2020) **Review date: 06/05/2020 13:24:06**

Department assigned to the subject:

Coordinating teacher: GARCIA GUZMAN, JAVIER

Type: Compulsory ECTS Credits : 6.0

Year : 3 Semester : 2

REQUIREMENTS (SUBJECTS THAT ARE ASSUMED TO BE KNOWN)

Principles of Software Development
Software Engineering

OBJECTIVES

The goal of this course is to allow the student knowing the basic concepts and principles of agile software project management methods and techniques. To achieve this aim, the students have to acquire specific knowledge and a set of skills and competencies.

Regarding the students knowledge at the end of this course, the student has to know:
1) Basic principles of the agile software development approach. (PO: a, c)
2) Most relevant methods for agile software development. (PO: a, c)
3) Know the business processes related to the creation of innovative and disruptive software services or products (PO: a, c)
   a) Lean Startup Overview
   b) Minimum Viable Product
   c) Measurement
   d) Learning
   e) Business Model Canvas
4) Know the processes and techniques for an agile requirements management. (PO: a, c)
   a) Techiques to discover and elicit new products or services: Product Visión Box, Story Mapping y Story Telling
   b) Techniques to manage a product backlog: Estimation, Priorization Technical Debt
5) The foundations of the management of software projects based on an incremental and iterative lifecycle with time-boxed iterations. (PO: a, c)
   a) Porfolio, Product, release and iteration planning.
   b) Quantitative agile projects control and tracking.
   c) Iterations control and tracking.
   d) Showcases and retrospectives.
6) Understand the difference between traditional and agile software development methods, being able to determine the most appropriate approach depending on a specific Project restrictions. (CECRI16) (PO: a, c)

Regarding the skills and competencies, they can be classified in two groups: specific and generic competences.

Regarding the specific competences, at the end of the course, an student has to be able to (CG1):
1) Organize the strategic project for creating a disruptive service or product (PO: a, c, i, k)
2) Identify and manage the user needs regarding an information system using the user stories technique. (PO: a, c, i, k)
3) Organize the different iterations of a software development project applying the adaptative planning technique (CESI5). (PO: a, c, i, k)
4) Know agile project control techniques: daily meetings, burn down/up charts, showcases and retrospectives. (PO: a, c, i, k)

Regarding the generic competences, at the end of the course, an student has to be able to:
1) Organize, plan and manage a team to develop a project applying the agile software development principles. (PO: d)
2) Follow a plan to obtain a product with the required level of quality. (PO: e)
3) Apply the principles learnt to the management of a software project.
4) Problems resolution including the identification and validation of an hypothesis to solve the problem in a collaborative and participative way (PO: d, e)
5) To control and communicate effectively the current state, results and benefits of a software project using objective evidences. (PO: g)

DESCRIPTION OF CONTENTS: PROGRAMME

1) Agile Development Fundamentals
1.1.- Agile Manifest
1.2.- Agile Principles
1.3.- Agile Software Project Lifecycle
2) Agile Requirements Management
2.1.- User Stories: Product Vision Box, Story Telling and Story Mapping
2.2.- Product Backlog Management
3) Lean Startup
3.1.- Introduction to Lean Startup
3.2.- Minimum Viable Product
3.3.- Measurement
3.4.- Learning
3.5.- Business Model Canvas
4) Agile Project Planning
4.1.- Agile Planning Levels
4.2.- Product roadmap
4.3.- Release Planning
4.4.- Iteration Planning
4.5.- System Tests as mechanisms to manage software projects
5) Agile Projects Management
5.1.- Tasks management using Kanban charts
5.2.- Burn down and burn up charts.
6) Retrospectives and Showcases
6.1.- Showcases and demonstrations
6.2.- Retrospectives

LEARNING ACTIVITIES AND METHODOLOGY

The learning activities considered for this subject are:

1) Theory Lectures (1,5 ECTS): These classes have as main goal to provide to the students the theoretical knowledge considered for this subject. (PO: a, c, i, k)
2) Practice Lectures (1,5 ECTS): These classes have as main goal to learn the students how to use in a practical way the agile software development techniques and tools. (PO: a, c, i, k)
3) Exercises and Practices (3 ECTS):
3.a) Guided Exercises: The students will complete individually (or in pairs) a set of exercises to enhance the learning of the theoretical concepts of this subjects. These guided exercises will include a script to complete them and they will be started during the practice lectures and finalized as homework. (PO: c, e)
3.b) Final Project: The objective of this final Project is to apply in a practical way all the concepts, techniques and tools learnt during the subject. This final project consists of the improvement of an already existing software system in teams having its members a predefined role. (PO: d, e, g)

ASSESSMENT SYSTEM

% end-of-term-examination/test:	40
% of continuous assessment (assigments, laboratory, practicals…):	60

The assessment of this subject is fully based on the application of the continuous evaluation principles. The students final mark will be calculated on the basis of the results obtained as result of the evaluation of the solution provided by the students for the guided exercises and final project. The mark will be calculated in the following way:
1) Theory evaluation - Test and class activities (PO: a, c, e, k)
- 30% of the final mark

% end-of-term-examination/test:	40
% of continuous assessment (assigments, laboratory, practicals…):	60

2) Guided Exercises (PO: a, c, e, k)
- 30% of the final mark

3) Final Project (PO: a, c, d, e, g, i, k)
- 40% of the final mark

BASIC BIBLIOGRAPHY

- Jeff Patton User Story Mapping, O¿Reilly Media, 2014

- Kniberg, Henrik. Lean from the Trenches. Managing Large-Scale Projects with Kanban. , The Pragmatic Bookself, 2011

- Moreira M, Lester M, Holzner S. Agile for Dummies, Wiley Publishing , 2010

- Richard Levy Agile Foundations: Principles, practices and frameworks, BCS, 2015

- Rubin, Kenneth S., Essential Scrum: a practical guide to the most popular agile process, Pearson Education, Inc., 2013

- Venkat Subramaniam and Andy Hunt Practices of an Agile Developer, The Pragmatic Bookshelf. Pracmatic Programmmers, 2013

ADDITIONAL BIBLIOGRAPHY

- Cockburn, Alistair Agile Software Development, Addison-Wesley, 2001.

- Highsmith, Jim Agile Software Development Ecosystems, Addison-Wesley, 2002.

- Schwaber, Ken et al. Agile Software Development With SCRUM, Prentice Hall, 2001

BASIC ELECTRONIC RESOURCES

- Agile Alliance . Agile Alliance: http://www.agilealliance.org