

COURSE DESCRIPTION – ACADEMIC YEAR 2019/2020

Course title	Agile Software Development
Course code	76055
Scientific sector	INF/01
Degree	Master in Software Engineering for Information Systems (LM-18)
Semester	1
Year	1
Credits	6
Modular	No
Total lecturing hours	40
Total exercise hours	20
Attendance	Attendance is not compulsory, but non-attending students are suggested to contact the lecturer at the start of the course to agree on the modalities of the independent study.
Prerequisites	Basic knowledge of software engineering activities and processes, open mindset and willingness to work under uncertainty.
Course page	https://ole.unibz.it/
Specific educational objectives	<p>The course belongs to the type caratterizzanti – discipline informatiche and is part of the Specialization Topics.</p> <p>The Agile Software Development course intends to instill into future software engineers an agile mentality, and to improve their capabilities of working on software development projects in an agile manner. The main educational objectives are:</p> <ul style="list-style-type: none"> ● Understanding the root and essence of agile software development and different agile approaches ● Applying key agile engineering and project management practices in software development projects ● Improving teamwork using agile approaches ● Scaling agile software development beyond agile home ground, including distributed and large software development projects.
Lecturer	Xiaofeng Wang
Contact LA	Office POS 3.15, xiaofeng.wang@unibz.it , +39 0471 016181
Scientific sector of lecturer	INF/01
Teaching language	English
Office hours	During the lecture time span, Fridays 15:00 - 17:00, arrange beforehand by email.
Lecturing Assistant (if any)	--
Contact LA	--
Office hours LA	--
List of topics	<ul style="list-style-type: none"> ● Software crisis and the origin of agile software movement ● Different agile software development approaches and key agile practices ● From time-boxed agile methods to continuous flow: lean software development ● Continuous experimentation and continuous software engineering

	<ul style="list-style-type: none"> • Teamwork in agile software development • Scaling agile: distributed and/or large software development projects using agile methods
Teaching format	Frontal lectures and team projects
Learning outcomes	<p>Knowledge and understanding:</p> <p>D1.5 To know the fundamentals, techniques and methods of design, customization and implementation of software to support the automation of new generation information systems for industrial production and business;</p> <p>D1.6 To understand the elements of corporate and professional culture.</p> <p>Applying knowledge and understanding:</p> <p>D2.4 To be able to define an innovative technical solution to an application problem that meets technical, functional and organisational constraints and requirements.</p> <p>Making judgments:</p> <p>D3.2 To be able to plan and re-plan a technical project activity and to carry it out in accordance with defined deadlines and objectives;</p> <p>D3.3 To be able to define work objectives compatible with the time and resources available;</p> <p>D3.4 To be able to reconcile the objectives of the project that are in conflict, to trade-off cost, resources, time, knowledge or risk;</p> <p>D3.5 To be able to work with large autonomy, also assuming responsibility for projects and structures.</p> <p>Communication skills:</p> <p>D4.5 To be able to prepare and conduct technical presentations in English;</p> <p>D4.6 To be able to interact and collaborate during the implementation of a project or research with peers and experts.</p> <p>Learning skills:</p> <p>D5.3 In the context of a problem solving activity, to be able to extend knowledge, even if incomplete, taking into account the final objective of the project.</p>
Assessment	<p>Exam type for regularly attending students:</p> <ul style="list-style-type: none"> • Project work and oral exam: <ul style="list-style-type: none"> ○ Project work to apply agile practices in a software development project (70% of the mark, team); ○ Oral exam to test the understanding of theories and knowledge application skills, and verification of project results (30% of the mark, individual). <p>Note: Positive project result is necessary to attend the oral exam. Both parts of results must be positive to pass the exam. In case of a positive mark, the project will count for all 3 regular exam sessions.</p> <p>Exam type for non-attending students:</p> <ul style="list-style-type: none"> • Written report and oral exam:

	<ul style="list-style-type: none"> o Written report on a piece of research related to agile software development (agreed upon with the lecturer at the beginning of the course) (70% of the mark); o Oral exam to test the understanding of theories and verification of written report (30% of the mark). <p>Note: Positive written result is necessary to attend the oral exam. Both parts of results must be positive to pass the exam. In case of a positive mark, the written result will count for all 3 regular exam sessions.</p>
Assessment language	English
Assessment typology	Monocratic
Evaluation criteria and criteria for awarding marks	<p>For regularly attending students:</p> <ul style="list-style-type: none"> • Evaluation criteria for project work: <ul style="list-style-type: none"> o effective application of agile practices o good teamwork o quality of developed solution • Evaluation criteria for oral exam: <ul style="list-style-type: none"> o ability to summarize, evaluate, and make connections between various topics o clarity of answers <p>For non-attending students:</p> <ul style="list-style-type: none"> • Evaluation criteria for written report: <ul style="list-style-type: none"> o good understanding of the literature o clarity of the research method o convincing research results • Evaluation criteria for oral exam: <ul style="list-style-type: none"> o ability to summarize, evaluate, and make connections between various topics o clarity of answers
Required readings	<ul style="list-style-type: none"> • Agile Manifesto: http://agilemanifesto.org/ • Highsmith, Jim. Agile Software Development Ecosystems. Boston, 2002. <p>Subject Librarian: David Gebhardi, David.Gebhardi@unibz.it</p>
Supplementary readings	<ul style="list-style-type: none"> • Rubin, Kenneth. Essential Scrum: A Practical Guide to the Most Popular Agile Process. Safari, an O'Reilly Media Company, 2012. • Beck, Kent, and Andres, Cynthia. Extreme Programming Explained: Embrace Change. 2.nd ed. Boston: Addison-Wesley, 2005. • Poppendieck, Mary, and Poppendieck, Tom. Lean Software Development: An Agile Toolkit for Software Development Managers. Harlow: Addison-Wesley, 2003.
Software used	Based on types of projects, decided by project teams