

COURSE DESCRIPTION – ACADEMIC YEAR 2024/2025

Course title	Agile Software Development
Course code	76055
Scientific sector	INF/01
Degree	Master in Software Engineering (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 have 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	MS Teams
Specific educational objectives	<p>The course belongs to the type “caratterizzanti – discipline informatiche”.</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	Ser I 1.07, xiaofeng.wang@unibz.it, +39 0471 016181
Scientific sector of lecturer	INF/01
Teaching language	English
Office hours	During the lecture time span, Fridays 16:00 - 18:00, arrange beforehand by email.
Lecturing Assistant (if any)	Dron Khanna
Contact LA	Ser I 1.07, dron.khanna@unibz.it
Office hours LA	During the lecture time span, Fridays 16:00 - 18:00, arrange beforehand by email.
List of topics	<ul style="list-style-type: none"> • Origin and evolution of agile software development and modern agile • Major agile frameworks and hybrid approaches • Key agile engineering and project management practices

	<ul style="list-style-type: none"> • People-centric and teamwork in agile software development • Continuous experimentation using agile approaches • Scaling agile: distributed and/or large agile software development projects
Teaching format	

Learning outcomes	<p>Knowledge and understanding:</p> <ul style="list-style-type: none"> • 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; • D1.6 To understand the elements of corporate and professional culture. <p>Applying knowledge and understanding:</p> <ul style="list-style-type: none"> • 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>Making judgments:</p> <ul style="list-style-type: none"> • 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; • D3.3 To be able to define work objectives compatible with the time and resources available; • 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; • D3.5 To be able to work with large autonomy, also assuming responsibility for projects and structures. <p>Communication skills:</p> <ul style="list-style-type: none"> • D4.4 To be able to prepare and conduct technical presentations in English; • D4.5 To be able to interact and collaborate during the implementation of a project or research with peers and experts. <p>Learning skills:</p> <ul style="list-style-type: none"> • 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.
--------------------------	--

Assessment	<p>Exam type for regularly attending students:</p> <ul style="list-style-type: none"> • Project work (50% of the final mark): applying agile approaches in a software development project (team score);
-------------------	---

	<ul style="list-style-type: none"> • Oral exam (50% of the final mark): to test the understanding of theories and knowledge application skills, and verification of project results (individual score). <p>Note: Positive project result is necessary to attend the oral exam. Both parts of the 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"> • Project work (70% of the final mark): applying the learnt agile knowledge to analyse an existing software development project; • Oral exam (30% of the final mark): to test the understanding of theories and knowledge application skills, and verification of project results. <p>Note: Positive project result is necessary to attend the oral exam. Both parts of the 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>
Assessment language	English
Assessment typology	Monocratic
Evaluation criteria and criteria for awarding marks	<p>For attending students:</p> <ul style="list-style-type: none"> • Evaluation criteria for project work: effective application of agile knowledge good teamwork quality of developed solution • Evaluation criteria for oral exam: ability to summarize, evaluate, and make connections between various topics clarity of answers <p>For non-attending students:</p> <ul style="list-style-type: none"> • Evaluation criteria for project work: effective application of agile knowledge quality of the analysis • Evaluation criteria for oral exam: ability to summarize, evaluate, and make connections between various topics clarity of answers
Required readings	<p>Agile Manifesto: http://agilemanifesto.org/ Agile Essentials on Agile Alliance website: https://www.agilealliance.org/agile-essentials/ Modern Agile: https://modernagile.org/ Subject Librarian: David Gebhardi, David.Gebhardi@unibz.it</p>

Supplementary readings	Highsmith, Jim. Agile Software Development Ecosystems. Boston, 2002. Research papers on agile software development, which will be distributed during the lectures
Software used	Based on types of projects, decided by project teams