

## COURSE DESCRIPTION – ACADEMIC YEAR 2020/2021

<b>Course title</b>	<b>Agile Software Development</b>
<b>Course code</b>	76055
<b>Scientific sector</b>	INF/01
<b>Degree</b>	Master in Software Engineering for Information Systems (LM-18)
<b>Semester</b>	1
<b>Year</b>	1
<b>Credits</b>	6
<b>Modular</b>	No
<b>Total lecturing hours</b>	40
<b>Total exercise hours</b>	20
<b>Attendance</b>	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.
<b>Prerequisites</b>	Basic knowledge of software engineering activities and processes, open mindset and willingness to work under uncertainty.
<b>Course page</b>	<a href="https://ole.unibz.it/">https://ole.unibz.it/</a>
<b>Specific educational objectives</b>	<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"> <li>• Understanding the root and essence of agile software development and different agile approaches</li> <li>• Applying key agile engineering and project management practices in software development projects</li> <li>• Improving teamwork using agile approaches</li> <li>• Scaling agile software development beyond agile home ground, including distributed and large software development projects.</li> </ul>
<b>Lecturer</b>	<a href="#">Xiaofeng Wang</a>
<b>Contact LA</b>	Office POS 3.15, <a href="mailto:xiaofeng.wang@unibz.it">xiaofeng.wang@unibz.it</a> , +39 0471 016181
<b>Scientific sector of lecturer</b>	INF/01
<b>Teaching language</b>	English
<b>Office hours</b>	During the lecture time span, Fridays 15:00 - 17:00, arrange beforehand by email.
<b>Lecturing Assistant (if any)</b>	--
<b>Contact LA</b>	--
<b>Office hours LA</b>	--
<b>List of topics</b>	<ul style="list-style-type: none"> <li>• Software crisis and the origin of agile software movement</li> <li>• Different agile software development approaches and key agile practices</li> <li>• From time-boxed agile methods to continuous flow: lean software development</li> <li>• Continuous experimentation and continuous software engineering</li> </ul>

	<ul style="list-style-type: none"> <li>• Teamwork in agile software development</li> <li>• Scaling agile: distributed and/or large software development projects using agile methods</li> </ul>
<p><b>Teaching format</b></p>	<p>Frontal lectures and team projects</p>
<p><b>Learning outcomes</b></p>	<p><b>Knowledge and understanding:</b>  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> <p><b>Applying knowledge and understanding:</b>  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><b>Making judgments:</b>  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> <p><b>Communication skills:</b>  D4.5 To be able to prepare and conduct technical presentations in English;  D4.6 To be able to interact and collaborate during the implementation of a project or research with peers and experts.</p> <p><b>Learning skills:</b>  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>
<p><b>Assessment</b></p>	<p><b>Exam type for regularly attending students:</b></p> <ul style="list-style-type: none"> <li>• Project work (70% of the final mark): a good demonstration of applying agile approaches in a software development project (70% team score, 30% individual score)</li> <li>• Oral exam (30% of the final mark): to test the understanding of theories and knowledge application skills, and verification of project results (30% of the mark, individual score).</li> </ul> <p><i>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.</i></p>

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