

<b>MODELLO D (inglese)</b>	
<b>General Information</b>	
Academic subject	Social Computing
Degree course	Computer Science (second-level degree in Computer Science)
Curriculum	Software and Services Engineering
ECTS credits	6
Compulsory attendance	No
Language	English

<b>Subject teacher</b>	<b>Name Surname</b>	<b>Mail address</b>	<b>SSD</b>
	Filippo Lanubile	filippo.lanubile@uniba.it	INF/01

<b>ECTS credits details</b>			
Basic teaching activities	Lectures	Tutorials and lab	

<b>Class schedule</b>	
Period	1st semester
Year	2nd
Type of class	Lecture- workshops

<b>Time management</b>	
Hours	62
Hours of lectures	32 (4 credits)
Tutorials and lab	30 (2 credits)

<b>Academic calendar</b>	
Class begins	24/09/2019
Class ends	10/01/2020

<b>Syllabus</b>	
<b>Prerequisites/requirements</b>	
Expected learning outcomes (according to Dublin Descriptors) (it is recommended that they are congruent with the learning outcomes contained in A4a, A4b, A4c tables of the SUA-CdS)	<p><i>Knowledge and understanding</i> The students will know the foundations of Social Computing, that is systems in which users interact, directly or indirectly, with what they believe to be other users or other users' contributions.</p> <p><i>Applying knowledge and understanding</i> The students will be able to apply analytics methods and tools to recognize social dynamics and put in relationship interaction traces and outcomes.</p> <p><i>Making informed judgements and choices</i> The students will learn how to discover what are the factors behind successful collaboration and which ones are actionable.</p> <p><i>Communicating knowledge and understanding</i> The students will learn how to communicate in teamwork through individual and collaborative exercises.</p> <p><i>Capacities to continue learning</i> The students will be able to autonomously learn theoretical concepts and empirical evidence by reading research papers.</p>

<p>Contents</p>	<p>Lectures</p> <ul style="list-style-type: none"> <li>- (Part 1) Social computing: fundamentals and application areas <ul style="list-style-type: none"> <li>o Introduction to Social Computing</li> <li>o Social Network Analysis</li> <li>o Social media sentiment analysis</li> </ul> </li> <li>- (Part 2) Social computing meets software engineering: collaborative software engineering <ul style="list-style-type: none"> <li>o Collaborative tools for software development</li> <li>o Version control systems: Git and GitHub Flow</li> <li>o Agile development</li> <li>o DevOps and Continuous Delivery</li> <li>o DataOps and Continuous Delivery for Machine Learning</li> <li>o Version control systems for managing ML experiments</li> <li>o Distributed software development</li> <li>o Open source software communities</li> </ul> </li> </ul> <p>Tutorials and Lab</p> <ul style="list-style-type: none"> <li>- Slack and Trello</li> <li>- Gephi</li> <li>- SentiStrength and Senti4SD</li> <li>- Git and GitHub</li> <li>- Gradle and Travis CI</li> <li>- DVC</li> <li>- Sim GSD</li> <li>- Open source guides by GitHub</li> </ul>
<p>Course program</p>	
<p>Bibliography</p>	<p>Part 1</p> <ul style="list-style-type: none"> <li>- K. Crowston. Introduction to ACM Transactions on Social Computing. Trans. Soc. Comput. 1, 1, Article 1e (February 2018), DOI: 10.1145/3181713</li> <li>- P. Zaphiris, C.S. Ang, A. Laghos (2012). Online Communities. In A. Sears &amp; J. Jacko (Eds.), The Human-Computer Interaction Handbook. Lawrence Erlbaum &amp; Associates, 2006 (available at <a href="https://www.scribd.com/document/140824708/Zaphiris-Ang-Laghos-2012-Online-Communities-The-Human-Computer-Interaction-Handbook">https://www.scribd.com/document/140824708/Zaphiris-Ang-Laghos-2012-Online-Communities-The-Human-Computer-Interaction-Handbook</a>)</li> <li>- R. Hanneman, M. Riddle. 2005. Introduction to social network methods. (available at <a href="http://faculty.ucr.edu/~hanneman">http://faculty.ucr.edu/~hanneman</a>).</li> <li>- Albert-Laszlo Barabasi. 2016. Network Science. Cambridge University Press, (available at <a href="http://networksciencebook.com/">http://networksciencebook.com/</a>)</li> <li>- C. Potts, Sentiment Symposium Tutorial (available at <a href="http://sentiment.christopherpotts.net/lingstruc.html">http://sentiment.christopherpotts.net/lingstruc.html</a>)</li> </ul> <p>Part 2</p> <ul style="list-style-type: none"> <li>- M. Storey, A. Zagalsky, F. Filho L. Singer, D. German. 2016. How Social and Communication Channels Shape and Challenge a Participatory Culture in Software Development, IEEE Trans. on Software Engineering, DOI: 10.1109/TSE.2016.2584053</li> </ul>

	<ul style="list-style-type: none"> <li>- F. Lanubile, C. Ebert, R. Prikladnicki, A. Vizcaino, "Collaboration Tools for Global Software Engineering", IEEE Software, ISSN: 0740-7459, vol. 27, 2010, pp.52-55 DOI: 10.1109/MS.2010.39</li> <li>- Scott Chacon and Ben Straub. Pro Git. 2nd Edition (2014). Apress (available at <a href="https://git-scm.com/book/en/v2">https://git-scm.com/book/en/v2</a>)</li> <li>- S. Chacon. GitHub Flow (available at <a href="http://scottchacon.com/2011/08/31/githubflow.html">http://scottchacon.com/2011/08/31/githubflow.html</a>)</li> <li>- C. Brindescu et al. 2014. How do centralized and distributed version control systems impact software changes? ICSE 2014, DOI: 10.1145/2568225.2568322 (available at <a href="http://dig.cs.illinois.edu/papers/ICSE14_Caius.pdf">http://dig.cs.illinois.edu/papers/ICSE14_Caius.pdf</a>)</li> <li>- Manifesto for Agile Software Development (available at <a href="https://agilemanifesto.org/">https://agilemanifesto.org/</a>)</li> <li>- K. Schwaber, J. Sutherland. The Scrum Guide (available at <a href="http://www.scrumalliance.org/learn-about-scrum/the-scrum-guide">www.scrumalliance.org/learn-about-scrum/the-scrum-guide</a>)</li> <li>- C. Ebert, G. Gallardo, J. Hernantes and N. Serrano, "DevOps," in IEEE Software, vol. 33, no. 3, pp. 94-100, May-June 2016. doi: 10.1109/MS.2016.68</li> <li>- M. Fowler. Continuous Delivery. (available at <a href="https://martinfowler.com/bliki/ContinuousDelivery.html">https://martinfowler.com/bliki/ContinuousDelivery.html</a>)</li> <li>- The DataOps Manifesto (available at <a href="https://www.dataopsmanifesto.org/">https://www.dataopsmanifesto.org/</a>)</li> <li>- D. Sato. Continuous Delivery for Machine Learning (available at <a href="https://martinfowler.com/articles/cd4ml.html">https://martinfowler.com/articles/cd4ml.html</a>)</li> <li>- D. Smite, M. Kuhrmann and P. Keil (2014). Virtual Teams [Guest editors' introduction]. IEEE Software, 31(6), 41-46. DOI: 10.1109/MS.2014.149</li> <li>- Open Source Guides (available at <a href="https://opensource.guide/">https://opensource.guide/</a>)</li> </ul>
Notes	Bibliography will be integrated with the slides available on the ADA platform.
Teaching methods	Lectures and tutorials supported by slides and demos.
Assessment methods (indicate at least the type written, oral, other)	<p>Oral assessment:</p> <ul style="list-style-type: none"> <li>- presentations of recent research papers selected by the lecturer (for students regularly attending the course)</li> <li>- oral test, including questions about the course program (for students not regularly attending the course)</li> </ul> <p>Lab assessment:</p> <ul style="list-style-type: none"> <li>- tasks assigned and supervised by the lecturer (for students regularly attending the course)</li> <li>- a contribution (bug fixing, improvement, documentation, extension) submitted to an open source software project (for students not regularly attending the course)</li> </ul>
Evaluation criteria (Explain for each expected learning outcome what a student has to know, or is able to do, and how many levels of achievement there are.	The students should know the concepts presented and discussed during classes and be familiar with the tools introduced in the tutorials and lab sessions.
Further information	