DEGREE CURRICULUM

FREE SOFTWARE ENGINEERING

Coordination: GIMENO ILLA, JUAN MANUEL

Academic year 2020-21

FREE SOFTWARE ENGINEERING 2020-21

Subject's general information

<table>
<tr><td>Subject name</td><td>FREE SOFTWARE ENGINEERING</td></tr>
<tr><td>Code</td><td>102056</td></tr>
<tr><td>Semester</td><td>1st Q(SEMESTER) CONTINUED EVALUATION</td></tr>
<tr><td>Typology</td><td>
<table>
<tr><th>Degree</th><th>Course</th><th>Character</th><th>Modality</th></tr>
<tr><td>Bachelor's Degree in Computer Engineering</td><td>4</td><td>COMPULSORY</td><td>Attendance-based</td></tr>
</table>
</td></tr>
<tr><td>Course number of credits (ECTS)</td><td>6</td></tr>
<tr><td>Type of activity, credits, and groups</td><td>
<table>
<tr><th>Activity type</th><th>PRALAB</th><th>TEORIA</th></tr>
<tr><th>Number of credits</th><td>3</td><td>3</td></tr>
<tr><th>Number of groups</th><td>1</td><td>1</td></tr>
</table>
</td></tr>
<tr><td>Coordination</td><td>GIMENO ILLA, JUAN MANUEL</td></tr>
<tr><td>Department</td><td>COMPUTER SCIENCE AND INDUSTRIAL ENGINEERING</td></tr>
<tr><td>Teaching load distribution between lectures and independent student work</td><td>40% Presential (equivalent to 60h)
60% Autonomous work(equivalent to 90h)</td></tr>
<tr><td>Important information on data processing</td><td>Consult this link for more information.</td></tr>
<tr><td>Language</td><td>20% on-site
20% virtual
60% autonomous work</td></tr>
<tr><td>Distribution of credits</td><td>Juan Manuel Gimeno (3)
Montserrat Sendín (3)</td></tr>
</table>

FREE SOFTWARE ENGINEERING 2020-21

Teaching staff	E-mail addresses	Credits taught by teacher	Office and hour of attention
GIMENO ILLA, JUAN MANUEL	juanmanuel.gimeno@udl.cat	3	
SENDÍN VELOSO, MONTSERRAT	montse.sendin@udl.cat	3	

Subject's extra information

To follow this subject properly some previous skills on Software Engineering are recommended.

Learning objectives

- Knowing the concept of Free Sofware and its consequences.
- Knowing the organization of free projects.
- Basic use of the tools used both in GNU and Java projects.
- Get a perspective on free software both from its history and current existent projects.
- Knowing the main bussiness models and financing methods experimented around free software.
- Knowing the use of free software in the public sector.
- Knowing the support infrastructure for the development of free software.

Competences

Strategic competences of the UdL

- **CT2**: Mastering a foreign language, especially English.
- **CT3**: Training Experience in the use of the new technologies and the information and communication technologies.

Cross-disciplinary competences

- **EPS-11**: Capacity to understand the needs of the user expressed in a no technical language

Specific competencies of the degree

- **GII-IS1:** Capacity to develop, maintain and evaluate services and software systems that satisfy all the requirements of the user and behave in a reliable and efficient way, they can develop, keep and fulfil quality requirements, applying the theories, principles, methods and uses of the software engineering.
- **GII-IS3**: Capacity to give solution to problems of integration taking into account the strategies, standards and available technologies.
- **GII-IS4**: Capacity to identify and analyse problems and design, develop, implement, verify and find software solutions on the base of a suitable knowledge of the theories, models and current techniques
- **GII-IS6**: Capacity to design suitable solutions in one or more fields of application using methods of software engineering that integrate ethical, social, legal and economic issues.

Subject contents

Conceptual Part

1. Introductotion to FOSS

2. A little bit of history

3.Free Software Engineering: The Cathedral and the Bazaar

4. Programming languages as FOSS projects

5. Unicode and Application Internationalization

6. Documentation for FOSS

6.1 Free documentacion licenses

6.2 Free documentacion creation tools

7. Economic aspects

7.1 Financing methods

7.2 Bussiness models

8. FOSS projects

8.1 Development models for FOSS projects

8.2 Case studies

9. Additional support infrastructure to FOSS development

9.1 Communication tools

9.2 Project repositories

9.3 Other

10. FOSS in the public sector

10.1 General aspects

10.2 Case studies

In a parallel way, in laboratory sessions will be presented these contents:

- GNU build tools (make i autotools)
- Java build tools (ant i maven)
- JavaScript build tools (npm)
- Application deplayment using containers (docker)
- Toos for internationalization (GNU gettext, Java resource bundles)
- Project management tools (forgeries)

Methodology

- Continued assessment, in which the presentation of contents by the student and the exchange of experiences among classmates are integrated in a natural manner.
- Work in group.
- Both, lectures and laboratory sesions are combined to use all the tools presented.
- Students will need to deepen in the study of all themes autonomously and be able to asses the diferent options that are presented.
- In the development of the theoric homeworks the student will need to be critic to select and justify its

choices and conclusions.
- The works get completed with an oral presentation in which to defend all of the used criteria.
- The evaluation system (detailed in the corresponding section) is composed of: 1) written test (the 1rst partial exam); and 2) diverse practical works (to develop individually or in group, depending on each case).
- In the formative activities combine case studies (to be developed in pairs), individual work and application to concrete problems.

Development plan

Week	Theory (GG)	Laboratory (GG)/Experiences interchange sessions	Autonomous Work
1	History	Introduction to FOSS	Study
2	History	Make + automake	Estudi & selected reading
3	History	Maven	Study
4	History	Npm	Study & chapter reading
5	History Cat & Baz	Docker	Study & chapter reading
6	Unicode	gettext + bundles	Study & presentation preparation
7	ProgLang	Presentations	Study & presentation preparation
8	Presentations	Presentations	Study & i18n project
9	First midterm		
10	FOSS documentation	Presentation of the chosen tool	Study, i18n project and chapter reading
11	Economic aspects		Study, chapter reading and wiki deployment
12	FOSS Projects		Study, chapter reading and use case development over wiki support chosen
13	FOSS Projects		Study and use case development over wiki support chosen
14	Support infraestructure for FOSS	Presentation of the chosen tool	Study, chapter reading and use case development over wiki support chosen
15	FOSS and Public Sector		Chapter reading, use case development over wiki support chosen and preparation of the presentations
16	Presentation of group activities		
17	Presentation of group activities		
18	Tutories		
19	Recovery		

Evaluation

Actv.	Description	Weight	Minimum Grade	In group	Presential	Mandatory	Recoverable
Parc1	First midterm Basic concepts	20%	No	No	Yes	Yes	No
Actv1	Applications Architecture	20%	No	No	No[1]	Yes	No
Actv2	Internationalization project	10%	No	No	No	Yes	No
Actv3	Experiences with the usage of libre software tools	10%	No	No	Yes (50%)[1]	Yes	No
Actv4	Desployment and use of a wiki	10%	No	Yes	No	Yes	No
Actv5	Presentation of a case study	20%	No	Yes	No[1]	Yes	No
Actv6	Practical application of FOSS in the public sector	10%	No	Yes	No[1]	Yes	No

[1] These activities are materialized through one or more oral presentations in front of the class, which could be carried out online if the pandemic restrictions require it.

Final grade = 0,20 * Parc1 + 0,20 * Actv1 + 0,10 * Actv2 + 0,10 * Actv3 + 0,10 * Actv4 + 0,20 * Actv5 + 0,10 * Actv6

- Subject is passed if final grade is greater or equal than 5,0

Other considerations:

- Type of exam: concept fixation
- For all activities: programmed deliveries, unmovable dates.
- All the activities that include an oral presentation, each student will be avaluated individually by both, the professor and also the rest of the classmates (co-avaluation).

Bibliography

Basic bibliography

- Jesús González Barahona, Joaquín Seoane Pascual, Gregorio Robles, Introducción al Software Libre**.** Grupo de Sistemas y Comunicaciones, ESCET, Universidad Rey Juan Carlos de Madrid. 2ª Ed. (2007)
- Karl Fogel, Producing Open Source Software**.** Published under creative commons, (2013)
- Sam Williams (Second edition revisions by Richard M. Stallman). Free as in Freedom (2.0): Richard Stallman and the Free Software Revolution. Published under GNU free documentation license, (2010)

Additional bibliography

- John Calcote, AutoTools. A practitioner's guide to GNU Autoconf, automake, and libtool**.** No Starch Press (2010)
- Steven Weber, The success of open source. Harvard University Press (2004).