Management and Deployment of ITC Services

Syllabus

Academic year:	2011-2012	**Trimester:**	3
Project name:	Management and Deployment of ITC Services		
Course code:	21737		
Studies:	Telematics		
ECTS credits:	4		
Total number of student hours:	100		
Time line:			
Year:	third		
Type:	trimester		
Period:	third trimester		
Professors:	Johan Zuidweg (coordinator)		
Group:	1		

Syllabus

1. Project data

- **Academic year**: 2011-2012
- **Project name:** Management and Deployment of ITC Services **Code:** 21737
- **Type of project:** trunk course
- **Studies:** Telematics Engineering
- **ECTS credits:** 4
- **Total number of student hours:** 100
- **Time line:**
  - Year: *third*
  - Type: *trimester*
  - Period: *third trimester*
- **Coordinator:** Johan Zuidweg
- **Department:** Information and Communications Technologies
- **Professors:** Johan Zuidweg
- **Department:** Information and Communications Technologies
- **Group:** 1
- **Working language:** English
- **Building where the project is executed:** Roc Boronat (Ca l'Aranyó)
- **Hours:** Mondays 8.30-10.30, Tuesdays 10.30-12.30 and Thursdays 12.30-14.30

2. Introduction to the project

This document is the teaching guide for the "Management and Deployment of ITC Services" ("Gestió i Desplegament de serveis TIC") course, a mandatory course in the third year of Telematics Engineering studies of the Pompeu Fabra University. The objective of this project is for the students to acquire hands-on experience with software engineering processes and information technologies by designing and prototyping a real distributed computer application.

The Management and Deployment of ITC Services course follows the principles of Project Based Learning (PBL), where students learn by doing. Central to this project is the notion of apprenticeship, which prevails over simple knowledge transfer.

In this project, students will work together in groups of 4-5 to design a distributed application involving desktop computers, servers and mobile terminals. Students will learn to do a full software development cycle including user requirements capturing, user interface design, design by patterns and testing, applying Agile modelling and Extreme Programming techniques.

The Management and Deployment of ITC Services course is programmed in the third trimester of the third year of the Telematics Engineering studies and is a mandatory component of this study program.

This guide describes the different aspects of the project, including its objectives, technical content, evaluation, study material and planning of the sessions.

3. Prerequisites for this project

In order to be able to participate in this project, students should have a working knowledge of mathematics, programming and telecommunications theory corresponding to the third year of Telematics Engineering studies.

More specifically, students should have the following competences:

- To be able to apply basic mathematics (analysis, statistics, algebra and logic)
- To have basic experience in object oriented programming (Java)
- To have good knowledge of the OSI and TCP/IP protocol stacks and their protocols

Moreover the student will be expected to have a positive attitude towards teamwork, and to be prepared to solve not only technical problems, but also to participate in project planning, management and reporting.

4. Competences to develop in the course of the project

General competences	Specific competences
Instrumentals 1. Analysis and synthesis. Capacity to analyse real cases and resolve problems using knowledge acquired in the project. 2. Problem resolution, both individually and in group. 3. Planning. Creating and executing a realistic work plan, both individually and for the group. 4. Documentation. Creating complete and well structured documentation. 5. Presentation. Presentation of the solutions and final results of the project. **Interpersonals** 6. Collaboration and teamwork. 7. Resolving conflicts and problems that may occur in the group. **Systemics** 7. Capacity to apply theoretical knowledge in practice. 8. Comprehension and analysis of situations. 9. Resolving assignments and problems individually and with the group.	1. Applying Agile modelling and programming techniques 2. Capturing and analyzing user requirements with the involvement of real users. 3. Designing user interfaces. 4. Applying design patterns and frameworks. 5. Using virtualized infrastructure. 6. Implementing a working prototype involving desk top computers, servers and Android phones. 7. Defining and executing software tests.

5. Evaluation

Since this course is based on teamwork, all members of a group will receive the same grade. The results of each group will be evaluated on the basis of the material presented in the seminars, practical sessions and project deliverables where the following is evaluated with the corresponding weight:

- □ Software design (20%)
- □ Working prototype (20%)
- □ Communication and teamwork (20%)
- □ Practical sessions (20%)
- □ Final project presentation (10%)

This course depends heavily on group work in practical sessions and seminars. For this reason, the following penalizations apply in case of missed sessions:

1 or 2 session missed*	no penalization
3 sessions missed*	-0,5 points on overall grade
4 sessions missed*	-1,0 points on overall grade
5 sessions missed*	-2,0 points on overall grade
6 sessions or more missed*	student fails course regardless of overall grade

*out of a total of 4 practical sessions + 4 seminars.

In addition the coordinator reserves the right to adjust grades of underperforming group members on an individual basis.

Each group is required to submit two incremental sets of documentation in the course of the project. The last deliverable will be considered the final documentation of the project. The documentation will be evaluated for its technical content, structure and readability. Late submission of the deliverables will be penalized.

The dates for the deliverables, numbered E1-E3 are the following:

Deliverable	Due date	Description
D1	14 May 2012	First deliverable: user requirements and user interface
D2	25 June 2012	Final project documentation, including also system design and test results

Each final presentation will be 20 minutes (15 minutes for the presentation and 5 minutes for answering questions). The final presentation will be evaluated by the following:

- Clarity of the presentation
- Credibility, convincingness
- Technical content
- Time control
- Answers to questions

6. Content

This section defines the global assignment for the project, as well as the contents of the course.

6.1. Assignment

The assignment of the course is to design and prototype an ITC system that combats "password fatigue". Password fatigue refers to today's problem of having to remember and manage an increasing number of strong passwords. The system to be designed and prototyped in this course must enable a user to log in on internet resources (webmail, portals, web accounts, etc) with strong passwords without having to memorize them, using the user's mobile phone as verification and confirmation device.

The system must be based on requirements captured from real users, and the prototype must be tested and accepted by real users and by the coordinator who will act as the client of the system.

The groups have freedom to design and implement the system as they consider best, as long as it meets the description above and the user requirements captured during the project.

Alternatively, groups may propose a project of their own choosing. However, the proposed system has to satisfy the following criteria:

- It must involve at least a computer (desktop, server or both) and a mobile phone
- It must take requirements from real users who will be involved in the project
- It must be approved by the coordinator, who will act as the client of the project

6.2. Content modules

To provide the necessary theory and practice to design the system described in the assignment above, this course will address the following content in classes, seminars and practical sessions:

- Agile Development
  - Agile values
  - Agile Modelling
  - Using UML and other models
  - Communication
  - Tools
- eXtreme Programming (XP)
  - Iterative software development
  - Pair programming
- Human Machine Interfaces
  - User interface flow charts
  - Storyboards and mock ups
- Design Patterns
  - How to use design patterns
  - Model-View-Controller and other useful patterns
- Designing for Security
  - Web security
  - Network security

- Designing for Testing
    - Objectives of testing
    - Testing as integrated part of development
    - Black box and white box testing
    - Writing test cases
- Using Virtualized Infrastructures
    - Virtualization principles
    - Working with the UPF virtualized servers
- Programming for Android Devices
    - Android SDK

7. Methodology

The project's activities can be classified in four groups:

- Lectures: these are sessions with all students together, aimed at knowledge transfer. A class allows a professor to explain the basic theory necessary to solve the problems of his or her module.
- Seminars: these are sessions with maximum 3 groups of 4 or 5 students. Seminars are more interactive than classes, and are aimed at apprenticeship rather than knowledge transfer. In a seminar, students present and discuss their work, or collaborate on solving a particular problem.
- Practicums: these are sessions in a classroom with computers, where students learn to work hands-on with technologies explained during the course.
- Work outside the classroom: a large part of the time dedicated to the project will be spent outside the classroom. As this is a project aimed specifically at apprenticeship and "learning by doing", students will be expected do much of the work of their project autonomously and outside class hours. The work done outside the classroom typically includes reading articles, writing the project's documentation, preparing presentations, etc.

All formal communication in the project will be done through the Moodle platform. Only in exceptional cases will direct e-mail be used.

However, groups may use any means of internal communication they wish, including blogs, a website set up for the group, or online collaboration tools. The UPF or the project's coordinator will not provide such additional tools, however.

Each group is strongly advised to maintain a document store where all the group's documents and results are kept. Such an archive can be a physical file or (recommended) an on-line document repository such as Google Groups/Docs, Blogspot, etc (the group is free to choose the platform).

8. Information sources

8.1 Software Engineering and Server Technologies

Basic Bibliography

- Scott Ambler: “Agile Modelling: Effective Practices for eXtreme Programming and the Unified Process, ISBN 0-471-20282-7, John Wiley & Sons, 2002
- Glenford J. Myers: “The Art of Software Testing”, ISBN 0-471-46912-2, John Wiley & Sons, 2004

Complementary bibliography

- Erich Gamma, Richard Helm, Ralph Johnson,John Vlissides: “Design Patterns: Elements of Reusable Object-Oriented Software”, ISBN 0-201-63361-2, Addison-Wesley, 1995
- Martin Fowler: “UML Distilled: A Brief Guide to the Standard Object Modeling Language”, ISBN: 0-321-193687, Addison Wesley, 2003
- Jim Conallen: “Building Web Applications with UML”, ISBN: 0-201-61577-0, Addison Wesley, 1999

Other resources

- Toni Sallarès: "Introducció a UML: Enginyeria del Programari Orientada a Objectes”, UdG, http://ima.udg.es/Docencia/03-04/3105IS0008/IntUML.pdf

Please note that the basic bibliography as well as some of the complementary bibliography is publicly available on the Internet.

Activity planning

The planning of classes and seminars for the course is the following:

	Monday		Tuesday		Thursday		
Week	**Date**	**08.30-10.30**	**Date**	**10.30-12.30**	**Date**	**12.30-14.30**	**Work outside classroom**
1	9 Apr	holiday	10 Apr	Lecture	10 Apr	Lecture	Groups and assignments
2	16 Apr	(no class)	17 Apr	Lecture	19 Apr	Lecture	Requirements, initial design
3	23 Apr	holiday	24 Apr	Seminar S101	26 Apr	Seminar S102	
4	30 Apr	(no class)	1 May	holiday	3 May	Lecture	
5	7 May	(no class)	8 May	Practicum	10 May	Practicum	
6	14 May	Practicum	15 May	Lecture	17 May	Lecture	14 May: first documentation
7	21 May	(no class)	22 May	Seminar S101	24 May	Seminar S102	
8	28 May	holiday	29 May	Practicum	31 May	Practicum	
9	4 Jun	(no class)	5 Jun	Lecture	7 Jun	Lecture	
10	11 Jun	(no class)	12 Jun	Seminar S101	14 Jun	Seminar S102	Working prototype
11	18 Jun	Seminar S101	19 Jun	Seminar S102	21 Jun	start of exams	25 jun: final documentation