



Management and Deployment of ITC Services

Syllabus

Academic year:	2012-2013	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), Zulqarnain Rashid (practical computer sessions)	
Group:	1	

Syllabus

1. Project data

- **Academic year:** 2012-2013
- **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 (coordinator), Zulqarnain Rashid (practical computer sessions)
- **Department:** Information and Communications Technologies
- **Group:** 1
- **Working language:** English
- **Building where the project is executed:** Roc Boronat (Ca l'Aranyó)
- **Hours:** Mondays 12.30-14.30, Wednesdays 8.30-10.30 (only April 10th) and Thursdays 10.30-12.30

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

The following competences to be developed in the course are described in Catalan so as to be fully compliant with the ANECA descriptions.

General competences	Specific competences
Instrumentals G1. Capacitat d'anàlisi i síntesi G3. Capacitat per aplicar els coneixements a l'anàlisi de situacions i la resolució de problemes G4. Habilitat en la cerca i la gestió de la informació G6. Capacitat de comunicar-se amb propietat de forma oral i escrita en català i en castellà, tant davant audiències expertes com a inexpertes.	Competències Específiques Professionals H3. Capacitat per a la redacció i desenvolupament de projectes en l'àmbit de la seva especialitat. H4. Aprendre de manera autònoma nous coneixements i tècniques adequats per a la concepció, el desenvolupament o l'explotació de sistemes informàtics.
Interpersonals G8. Capacitat de treball en equip	Competències Específiques d'Enginyeria en Telemàtica TE1. Capacitat de construir, explotar i gestionar les xarxes, serveis, processos i aplicacions de telecomunicacions, enteses aquestes com a sistemes de captació, transport, representació, processament, emmagatzematge, gestió i presentació d'informació multimèdia, des del punt de vista dels serveis telemàtics.
Sistèmiques G11. Capacitat d'aplicar amb flexibilitat i creativitat els coneixements adquirits i d'adaptar-los a contextos i situacions noves G12. Capacitat per progressar en els processos de formació i aprenentatge de manera autònoma i contínua G14. Capacitat de motivació per la qualitat i per l'assoliment G15. Capacitat de generació de noves idees	TE2. Capacitat per aplicar les tècniques en què es basen les xarxes, serveis i aplicacions telemàtiques, tals com a sistemes de gestió, senyalització i commutació, encaminament i enrutament, seguretat (protocols criptogràfics, tonelatje, tallafocs, mecanismes de cobrament, d'autenticació i de protecció de continguts), enginyeria de tràfic (teoria de grafs, teoria de cues i teletràfic) tarificació i fiabilitat i qualitat de servei, tant en entorns fixos, mòbils, personals, locals o a gran distància, amb diferents amplituds de banda, incloent telefonia i dades. TE3. Capacitat de construir, explotar i gestionar serveis telemàtics, incloent internet, web, disseny arquitectònic (dades i protocols), enginyeria i tecnologies programari, gestió del coneixement distribuït i de la informació multimèdia, utilitzant eines analítiques de planificació, de dimensionament i d'anàlisi.

	TE4. Capacitat de descriure, programar, validar i optimitzar protocols i interfícies de comunicació en els diferents nivells d'una arquitectura de xarxes. Competències Específiques d'Enginyeria en Informàtica IN21. Ser capaç de dissenyar aplicacions distribuïdes aplicant els coneixements d'arquitectures i protocols de xarxa.
--	---

5. Evaluation

Since this course is based on teamwork, more than half the grade (60%) will be based on the documentation and presentations produced by the group. The other part of the grade is determined by the preparations of the seminars and practical sessions with computers, which have to be submitted individually or by pairs. The course has no final exam.

The overall grade is the weighted sum of four evaluation components as indicated in the following table:

Component	Group/Individual	Weight	Recoverable
Prototype and final documentation	Group	35%	Yes
Intermediate documentation of design	Group	10%	Yes
Final presentation	Group	15%	No
Seminar preparations	Individual/pairs	20%	Yes
Practical sessions with computers	Individual/pairs	20%	No

35% of the overall grade for each student is determined by the final results of the groups, which is due at the end of the trimester and consists of the prototype and final documentation including a description of user requirements, system architecture, system design, user interface design, test specifications and a security assessment of the application.

The groups will be required to submit an intermediate version of the documentation at mid-trimester, which will account for 10% of the overall grade.

The final presentations will account for 15% of the final grade and will be scheduled in the week of the exams. Each final presentation will have a maximum of 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

This course has a total of four seminars. All students will be required to submit their preparation of the seminar to the course Moodle before the seminar. Late submissions will not be accepted. These four submissions form the individual component of the grade and will account for 20% of the overall grade for the course.

This course has a total of five practical sessions with computers, the purpose of which is to gain experience with the use of the Android programming environment and virtualized servers. Students will have to submit individual assignments based on the hands-on work with Android which will account for 20% of the overall grade.

This course depends heavily on work done in seminars and practical computer sessions. For this reason, attendance of seminars and practical computer sessions will be monitored and the following penalizations apply in case of missed sessions without proper justification:

up to 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 or more sessions missed	student fails course regardless of overall grade

*out of a total of 4 seminars and 4 practicums (8 sessions). The last practicum is a troubleshooting session, and presence is not required.

Only medical conditions, pressing family circumstances such as death of a family member and official activities related to the UPF can be accepted as justification for absence in seminars or practical sessions, and these will have to be properly documented. Invalid reasons for missing seminars or practical sessions include incompatible work schedules, sports events that are not related to the UPF, travel time to or from home, non critical family events, holidays or holiday extensions, etc.

No penalizations apply for missing theory classes but attendance is strongly recommended to receive the theory necessary to complete the project and do the assignments of the seminars.

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

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

- Black box and white box testing
 - Writing test cases
- Programming for Android Devices
 - Android SDK
- Using Virtualized Infrastructures
 - Virtualization principles
 - Working with the UPF virtualized servers

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 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 to 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.

The number of hours per student inside and outside the classroom for each of these activities is given in the following table:

Contents	Contact hours	Outside the classroom
Theory	18	0
Seminars	8	12
Android practical sessions	10	7
Intermediate documentation	0	10
Prototype	0	15
Final documentation	0	10
Final presentation	2	8
Totals	38	62

Note that this course does not have a final exam, but the final presentation will be done in the exam week. The total number of hours dedicated to group work (prototype, documentation and presentation) will be the sum of the individual hours of each student in the group, indicated in this table.

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

There will be only one group for theory, seminars and practical sessions with computers. For this reason, we will not use the Wednesday classes except in the first week (Wednesday April 10th). There will be no classes in week 9 (3-7 June) and it is highly recommended that the students use this week to progress their final documentation and presentation.

The class and activity schedule is the following:

	Monday		Wednesday		Thursday		
Week	Date	12.30-14.30	Date	8.30-10.30	Date	10.30-12.30	Work outside classroom
1	8 Apr	Theory: intro, Agile	10 Apr	Practicum: Android	11 Apr	Theory: Agile, XP	
2	15Apr	Theory: XP	17 Apr	(no class)	18 Apr	Practicum:Android	Submit first practicum
3	22 Apr	Theory: Design Patterns	24 Apr	(no class)	25 Apr	Practicum: Android	
4	28 Apr	Theory: Testing	1 May	Holiday	2 May	Seminar: Testing	Submit first seminar
5	6 May	Theory: Testing	8 May	(no class)	9 May	Practicum: Android	Submit second practicum
6	13 May	Theory: User Interfaces	15 May	(no class)	16 May	Seminar: First doc.	First documentation due
7	20 May	Holiday	22 May	(no class)	23 May	Practicum: Online help	
8	27 May	Theory: User Interfaces	29 May	(no class)	30 May	Seminar: UI	Submit second seminar
9	3 Jun	(no class)	5 Jun	(no class)	6 Jun	(no class)	
10	10Jun	Theory: Security	12 Jun	(no class)	13 Jun	Seminar: Security	Submit third seminar
							Final documentation due