



## Teaching Guide

Identifying Data					2013/14
<b>Subject (*)</b>	Deseño de sistemas de información		<b>Code</b>	614502007	
<b>Study programme</b>	Mestrado Universitario en Enxeñaría Informática (plan 2012)				
Descriptors					
Cycle	Period	Year	Type	Credits	
Official Master's Degree	1st four-month period	First	Obligatoria	6	
<b>Language</b>	GalicianEnglish				
<b>Prerequisites</b>					
<b>Department</b>	Computación				
<b>Coordinador</b>	Sanchez Penas, Juan Jose		<b>E-mail</b>	juan.jose.sanchez.penas@udc.es	
<b>Lecturers</b>	Sanchez Penas, Juan Jose		<b>E-mail</b>	juan.jose.sanchez.penas@udc.es	
<b>Web</b>					
<b>General description</b>	<p>Revisaremos conceptos avanzados relacionados con todos os aspectos do deseño software, incluíndo patróns de deseño e arquitectura, deseño orientado a componentes, calidade no deseño, evolución do software, métricas e complexidade software, ou accesibilidade. O obxectivo será consolidar eses conceptos estudando proxectos do mundo real dende unha perspectiva profesional. O idioma principal da asignatura será o inglés.</p> <p>We will review advanced concepts related to all the aspects of software design, including design and architectural patterns, component-based design, design quality, software evolution, metrics and software complexity or software accessibility. We will focus on consolidating those concepts by studying complex real world projects from a professional perspective.</p>				

## Study programme competences

Code	Study programme competences
A1	Capacidade para a integración de tecnoloxías, aplicacións, servizos e sistemas propios da enxeñaría informática, con carácter xeneralista, e en contextos máis amplos e multidisciplinares.
A2	Capacidade para a planificación estratéxica, elaboración, dirección, coordinación, e xestión técnica e económica nos ámbitos da enxeñaría informática relacionados, entre outros, con: sistemas, aplicacións, servizos, redes, infraestruturas ou instalacións informáticas e centros ou factorías de desenvolvemento de sóftware, respectando o adecuado cumprimento dos criterios de calidade e ambientais e en contornos de traballo multidisciplinares.
A3	Capacidade para a dirección de proxectos de investigación, desenvolvemento e innovación en empresas e centros tecnolóxicos, con garantía da seguridade para as persoas e bens, a calidade final dos produtos e a súa homologación.
A4	Capacidade para modelar, deseñar, definir a arquitectura, implantar, xestionar, operar, administrar e manter aplicacións, redes, sistemas, servizos e contidos informáticos.
A6	Capacidade para asegurar, xestionar, auditar e certificar a calidade dos desenvolvementos, procesos, sistemas, servizos, aplicacións e produtos informáticos.
A7	Capacidade para deseñar, desenvolver, xestionar e avaliar mecanismos de certificación e garantía de seguridade no tratamento e acceso á información nun sistema de procesamento local ou distribuído.
A8	Capacidade para analizar as necesidades de información que se presentan nun contorno e levar a cabo en todas as súas etapas o proceso de construción dun sistema de información.
A9	Capacidade para deseñar e avaliar sistemas operativos e servidores, e aplicacións e sistemas baseados en computación distribuída.
A11	Capacidade de deseñar e desenvolver sistemas, aplicacións e servizos informáticos en sistemas encaixados e ubicuos.
A13	Capacidade para utilizar e desenvolver metodoloxías, métodos, técnicas, programas de uso específico, normas e estándares de computación gráfica.
A14	Capacidade para conceptualizar, deseñar, desenvolver e avaliar a interacción persoaordenador de produtos, sistemas, aplicacións e servizos informáticos.
B1	Capacidade de resolución de problemas.
B2	Traballo en equipo.
B3	Capacidade de análise e síntese.
B6	Toma de decisións.



B7	Preocupación pola calidade.
B9	Capacidade para xerar novas ideas (creatividade).
C2	Dominar a expresión e a comprensión de forma oral e escrita dun idioma estranxeiro.
C3	Utilizar as ferramentas básicas das tecnoloxías da información e as comunicacións (TIC) necesarias para o exercicio da súa profesión e para a aprendizaxe ao longo da súa vida.
C4	Desenvolverse para o exercicio dunha cidadanía aberta, culta, crítica, comprometida, democrática e solidaria, capaz de analizar a realidade, diagnosticar problemas, formular e implantar solucións baseadas no coñecemento e orientadas ao ben común.
C5	Entender a importancia da cultura emprendedora e coñecer os medios ao alcance das persoas emprendedoras.
C6	Valorar criticamente o coñecemento, a tecnoloxía e a información dispoñible para resolver os problemas cos que deben enfrontarse.
C7	Asumir como profesional e cidadán a importancia da aprendizaxe ao longo da vida.
C8	Valorar a importancia que ten a investigación, a innovación e o desenvolvemento tecnolóxico no avance socioeconómico e cultural da sociedade

Learning outcomes			
Subject competencies (Learning outcomes)	Study programme competences		
Know advanced concepts and techniques of Software Engineering	AJ3	BJ1	CJ2
	AJ6	BJ2	CJ3
	AJ7	BJ7	CJ6
	AJ9	BJ9	CJ7
	AJ11		
Know the languages and methodologies used in industry for the design of complex software systems	AJ3	BJ1	CJ2
	AJ4	BJ2	CJ3
	AJ9	BJ3	CJ6
Know advanced tools used in industry for the design of complex software systems	AJ1	BJ1	CJ2
	AJ9	BJ2	CJ3
	AJ11		CJ6
	AJ13		
Understand the main challenges and methodologies for evaluating design quality and software complexity	AJ4	BJ1	CJ2
	AJ6	BJ2	CJ3
	AJ9	BJ7	CJ7
			CJ8
Understand the importance of software accessibility and its technical implications	AJ1	BJ1	CJ2
	AJ6	BJ2	CJ4
	AJ14	BJ7	CJ7
Understand the challenges of real world complex systems, and the trade offs that their designers have to make	AJ2	BJ1	CJ2
	AJ6	BJ2	CJ3
	AJ8	BJ3	CJ4
	AJ14	BJ6	CJ5
		BJ7	CJ6
		BJ9	CJ7
			CJ8

Contents	
Topic	Sub-topic



Introduction to advanced software design	<p>Importance of software design</p> <p>Software design and software development processes and methodologies</p> <p>Design and architectural patterns, component-based design</p> <p>Software evolution, design quality, metrics and software complexity</p> <p>Software accessibility</p> <p>Real world examples of complex software design</p>
Advanced concepts of software design	<p>Languages and tools used for software design</p> <p>Design patterns</p> <p>Architectural patterns</p> <p>User interface and User experience patterns</p> <p>Introduction to software refactoring and evolution</p>
Advanced concepts of quality in software design	<p>Software and design quality</p> <p>Metrics and software complexity</p> <p>Evaluation and verification of software systems</p>
Advanced concepts of software accessibility	<p>Importance of software accessibility</p> <p>Software accessibility and software design</p> <p>Software accessibility standards</p> <p>Tools and technologies for software accessibility</p> <p>Case studies of software accessibility</p>
Real world case studies	<p>Overview of some well known, complex software systems</p> <p>Software design in industry-used open source projects</p> <p>In depth analysis of the design, tools, quality and accessibility in several open source projects (e.g. WebKit, GNOME&amp;KDE, Linux, MeeGo/Tizen, etc.)</p>

## Planning

Methodologies / tests	Ordinary class hours	Student?s personal work hours	Total hours
Guest lecture / keynote speech	10	15	25
Case study	10	20	30
Objective test	5	0	5
Workshop	10	20	30
Workbook	0	10	10
Laboratory practice	10	20	30
Events academic / information	0	8	8
Online forum	0	10	10
Personalized attention	2	0	2

(\*)The information in the planning table is for guidance only and does not take into account the heterogeneity of the students.

## Methodologies

Methodologies	Description
Guest lecture / keynote speech	We will invite relevant engineers and managers from the IT industry, in order to give the students guest lectures aligned with the contents of the course.
Case study	We will review real world projects and discuss how the theoretical content that we have studied in the lectures is applied there. We plan to focus mostly on open source projects, as we have full access to the design material and source code.
Objective test	Written exam, where the student will need to show both the theoretical knowledge acquired, and the capacity to resolve practical problems.
Workshop	Practical discussion, analysis and design sessions, with the students organized in groups, supervised by the teacher.
Workbook	The teacher will provide the student with relevant book chapters and articles, related to the theoretical content of the lectures, and the student will need to do a critical read of them, and prepare an executive summary that will be either reviewed by the teacher or by the whole classroom, depending on the case.



Laboratory practice	Practical design and coding exercises, with the students organized in groups, supervised by the teacher.
Events academic / information	We will complement the theoretical and practical lectures with attendance (either onsite or virtually) to conferences related to software design and development.
Online forum	All the topics discussed in lectures, workshops and practical lab time will have a follow up in the virtual forums. We will try to stimulate the discussion there and open new topics proposing links to extra content so that the students can complement their knowledge in the topics they have personal interest in.

## Personalized attention

Methodologies	Description
Online forum Workbook Laboratory practice	<p>The personal attention to the students includes, in this case, not only the classical supervision time (tutorship), or the virtual help using the online resources, but also the following actions:</p> <ul style="list-style-type: none"><li>- We will follow constantly the work of the student in all the supervised tasks that will be proposed along the duration of the subject.</li><li>- Assessment of the results obtained in the practical assignments developed by the student.</li><li>- Constant communication with the goal of solving the problems found by the student to understand the contents discussed in the lectures or the difficulties of the tasks proposed by the teacher.</li></ul>

## Assessment

Methodologies	Description	Qualification
Objective test	Written exam with 3 parts: short theoretical questions, more practical questions where the students can elaborate a bit more the answers to the problems explained, and a specific complete design challenge.	50
Workshop	<p>The assesment of practical tasks in workshops will be continuous along the course, and will be based on a final presentation to the teacher. We will consider as part of the assesment the following aspects:</p> <ul style="list-style-type: none"><li>- Capacity to work as part of a group.</li><li>- Personal capacity to carry out work and explain it.</li><li>- Capacity to cover all the goals of the task.</li><li>- Capacity to apply the knowledge acquired during the theoretical lessons.</li><li>- Critical thinking and capacity to innovate and find solutions to problems.</li><li>- Capacity to deliver the tasks on time.</li></ul>	50

## Assessment comments

The summary of the qualification distribution is that we will obtain 50% of the marks from the written exam and 50% from a collection of workshops and practical activities that will be carried out during the course.

## Sources of information

Basic	
Complementary	

## Recommendations

### Subjects that it is recommended to have taken before

Dirección de proxectos/614502002  
Calidade. seguridade e auditoría informática/614502003  
Arquitecturas e plataformas móbiles/614502005  
Prácticas en empresa/614502011  
Traballo fin de mestrado/614502012

### Subjects that are recommended to be taken simultaneously



Análise de sistemas de información/614502006

Subjects that continue the syllabus

Other comments

(\*)The teaching guide is the document in which the URV publishes the information about all its courses. It is a public document and cannot be modified. Only in exceptional cases can it be revised by the competent agent or duly revised so that it is in line with current legislation.