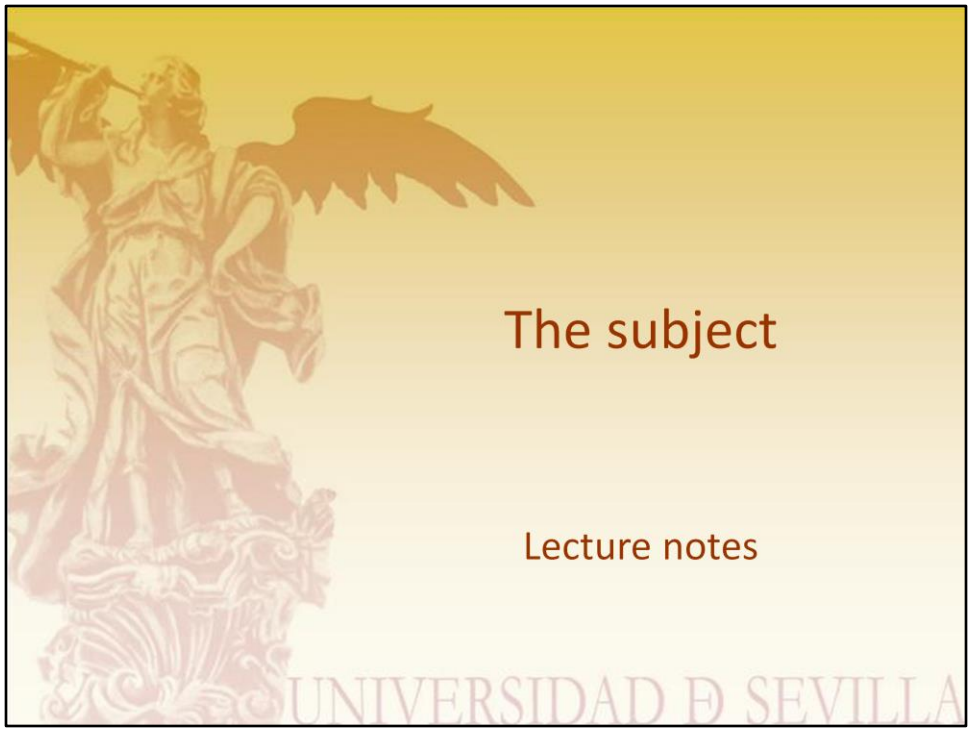
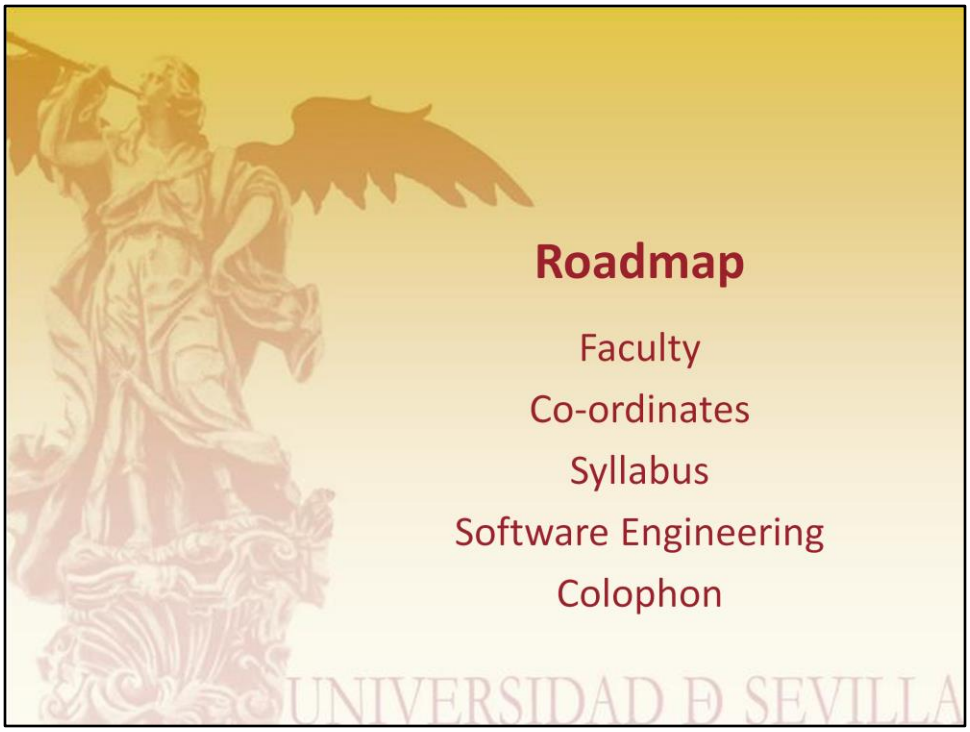


Welcome to this lecture! Today, our goal's to introduce Design & Testing.

--



This is our roadmap: first, we'll present the faculty, we'll then check our co-ordinates, we'll next introduce the syllabus, then we'll delve into what Software Engineering means in our context, and we'll conclude with a short colophon in which our goal is to recap on the role of D&T in your studies and your career as a professional Software Engineer.

NOTE: acronyms "D&T" or variants like "DT" and "DP" refer to this subject, "Design and Testing".

Let's start with the faculty, that is, the lectures who are in charge of D&T during this academic course.

The co-ordinator

Carlos Müller
cmuller@us.es

This is the co-ordinator of D&T, who is responsible for the subject. If you have any problems during the course, please, first talk to your lecturer and try to solve the problem with him or her; if you can't find a solution, then resort to the co-ordinator.

The seniors

Rafael Corchuelo
corchu@us.es

Inma Hernández
inmahernandez@us.es

Pablo Trinidad
ptrinidad@us.es

Patricia Jiménez
patriciajimenez@us.es

UNIVERSIDAD D SEVILLA

5

Here are the senior lecturers.

The rookies

Fernando O. Gallego
fogallego@us.es

José M. Luna
jmluna@us.es

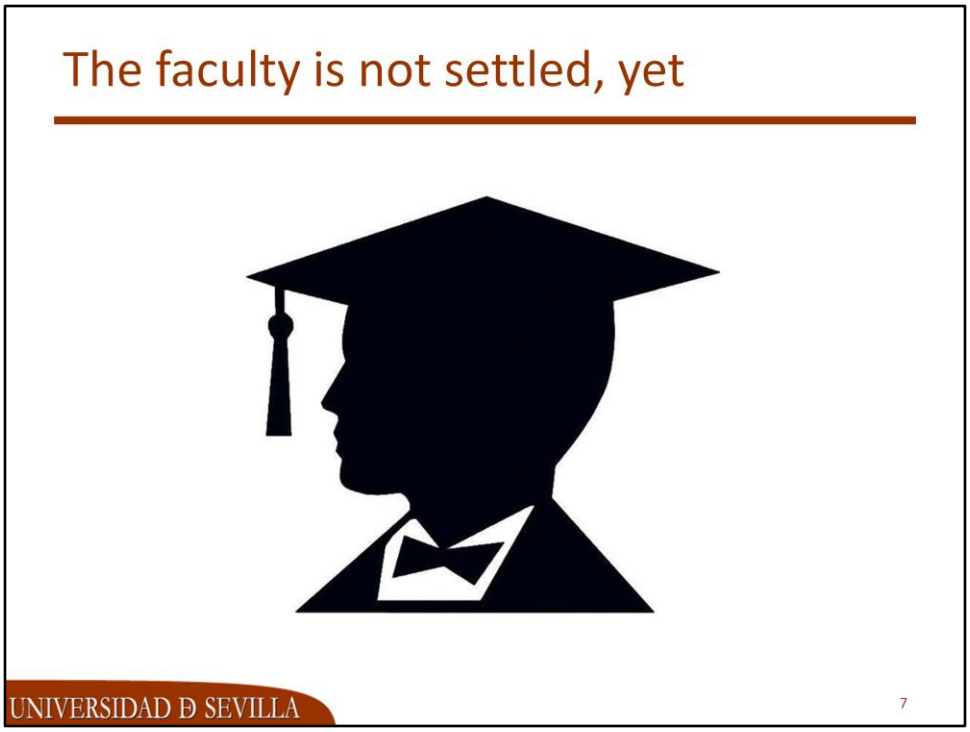
José Á. Galindo
jagalindo@us.es

Javier J. Gutiérrez
javierj@us.es

UNIVERSIDAD D SEVILLA

6

And here are the rookies.



Unfortunately, our faculty is not settled, yet. That means that some of the lecturers might quit the subject and some others might join us. But we don't expect too many changes.

Contacting your lecturers

UNIVERSIDAD D SEVILLA

8

Please, note that you must contact your lecturers from your corporate email account, which is provided by the USE. Messages that come from Gmail, Outlook, Yahoo!, and other such services are considered spam. Bear also in mind that there are some recommendations to address them. Please, consult document “On your tutorials” to learn more.

Let's now review the co-ordinates, that is, the groups, the schedules, the lecturing rooms, and the like.

Group 1 (Spanish)

- Mondays
  - 12:40 – 14:30
  - F1.30, F1.31, F1.32
- Thursdays
  - 10:40 – 12:30
  - A1.13

UNIVERSIDAD D SEVILLA

10

This is the schedule for group one. Please, bear in mind that the lectures start on time and there are no breaks. Please, check the work programme to know what we're planning on doing every day along the course.

Group 2 (Spanish)

- Mondays
  - 10:40 – 12:30
  - F0.31, F1.31, F1.33
- Thursdays
  - 08:30 – 10:20
  - H0.12

UNIVERSIDAD D SEVILLA

11

This is the schedule for group two. Please, bear in mind that the lectures start on time and there are no breaks. Please, check the work programme to know what we're planning on doing every day along the course.

Group 3 (Spanish)

- Mondays
  - 15:30 – 17:20
  - F0.31, F1.30, F1.32
- Thursdays
  - 17:40 – 19:30
  - A1.13

UNIVERSIDAD D SEVILLA

12

This is the schedule for group three. Please, bear in mind that the lectures start on time and there are no breaks. Please, check the work programme to know what we're planning on doing every day along the course.

Group 4 (English)

- Mondays
  - 08:30 – 10:20
  - F0.31
- Fridays
  - 08:30 – 10:20
  - A1.13

UNIVERSIDAD D SEVILLA

13

This is the schedule for group four, which is taught in English. Please, bear in mind that the lectures start on time and there are no breaks. Please, check the work programme to know what we're planning on doing every day along the course.

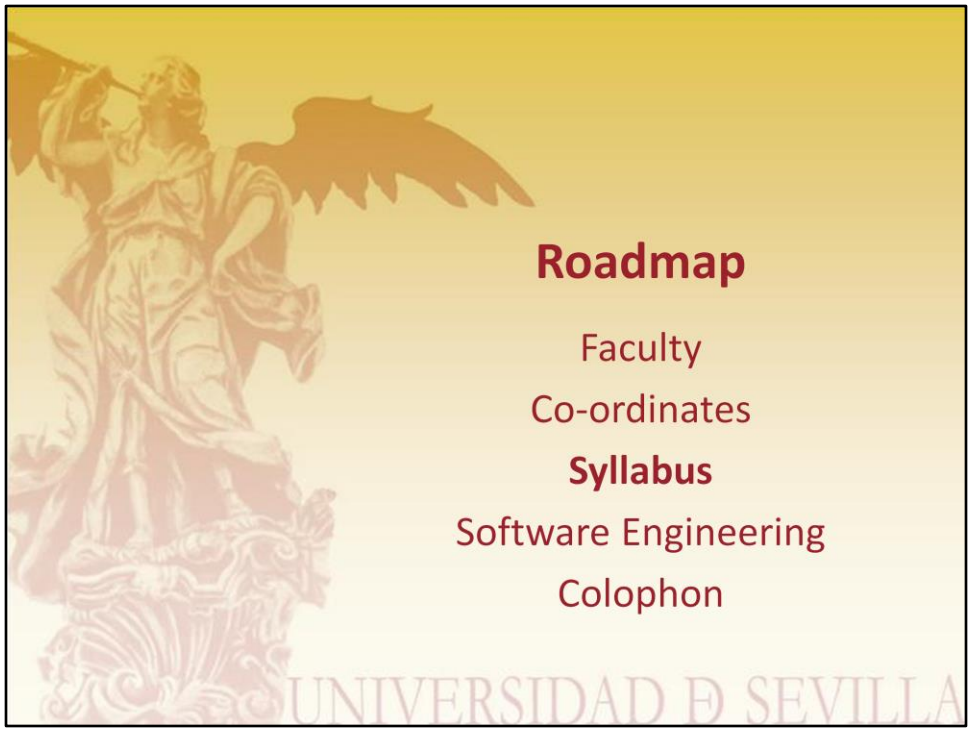
Hey! Realise this!

UNIVERSIDAD Ð SEVILLA

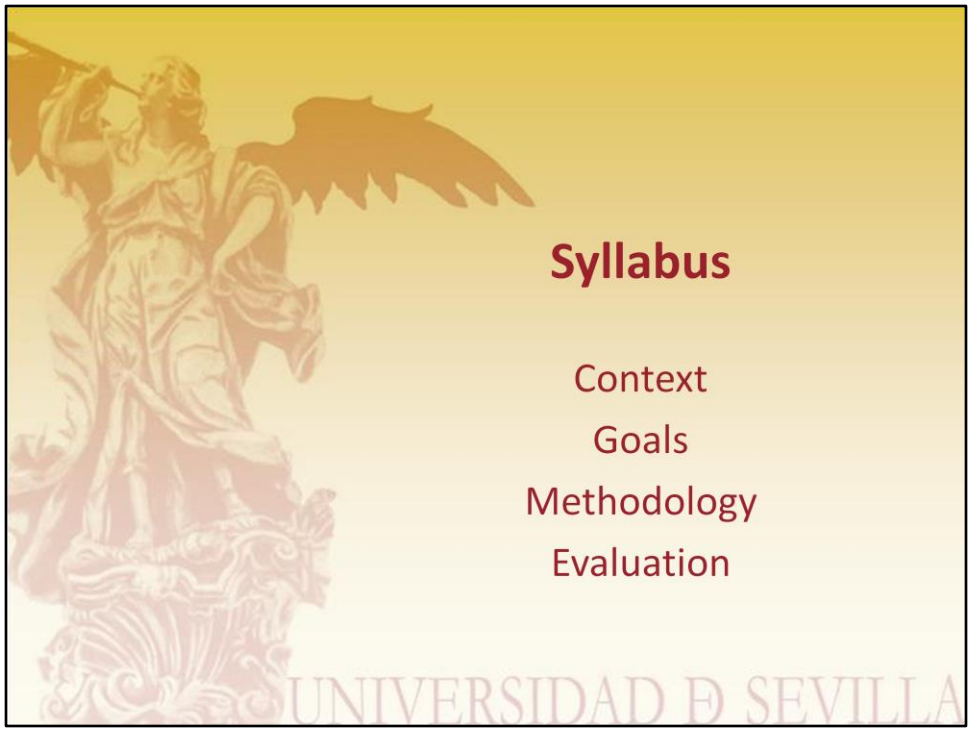
14

Please, note that the goal of the English group is not to teach English, but to help our students improve their linguistic skills. Note, too, that English should not be a barrier; in other words, if you don't understand something in English, please, ask your lecturer to explain it in Spanish. There's no problem with that. All of our students must be fluent at reading technical documentation in English, and this is the reason why D&T materials are available in English only. But, spoken English must never be a barrier to your achieving the goals of this subject.

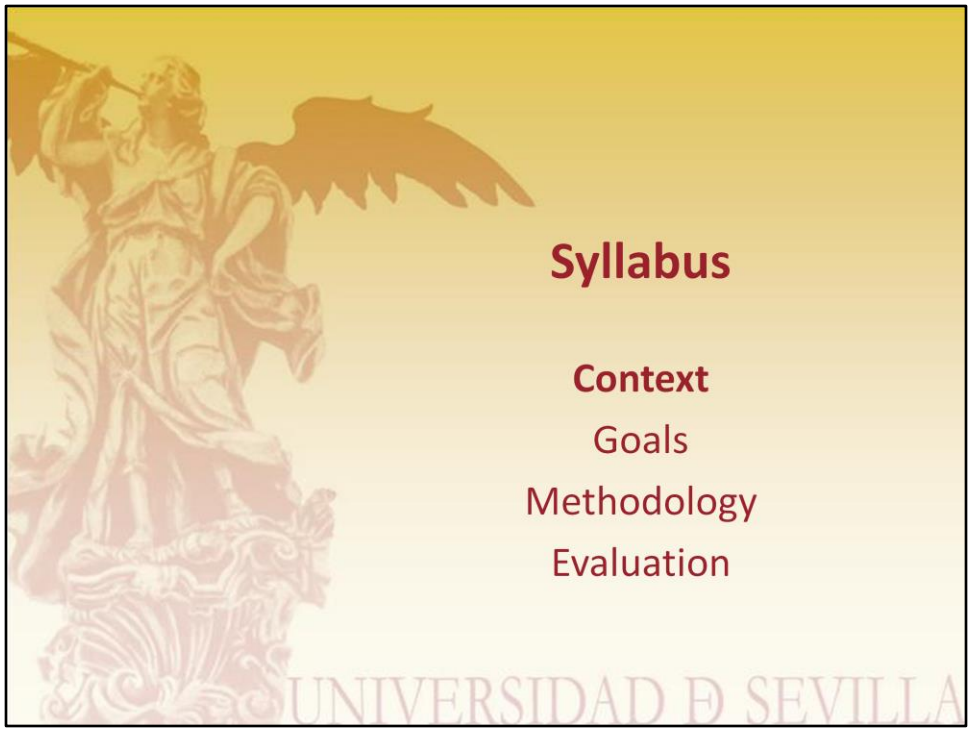
Please, consider moving to the English group, at least for your laboratory lectures. The Spanish groups are quite crowded. If you move to the English group for your laboratory lectures, the lecturer will help you out in Spanish; there's no problem with that.



We hope we didn't make any mistakes with the co-ordinates. So, let's start with the syllabus.



A syllabus is a document that describes the context, the goals, the methodology, and the evaluation of a subject. Our complete syllabus is available at the USE's e-learning platform, so that you can download it and review it at home. In this lecture, we'll just summarise it.



First, let's analyse the context of this subject.

Degree on Software Engineering

UNIVERSIDAD Ð SEVILLA

19

This is a subject within the degree on Software Engineering.

What's Software Engineering?

It's the application of a systematic, disciplined, quantifiable approach to the design, development, operation, and maintenance of software, and the study of these approaches

UNIVERSIDAD Ð SEVILLA

20

So we think that it's a good idea to start asking: what's Software Engineering? This slide provides quite an accepted definition: Software Engineering is the application of a systematic, disciplined, quantifiable approach to the design, development, operation, and maintenance of software, and the study of these approaches. This definition puts an emphasis on the fact that designing, developing, operating, or maintaining software isn't a matter of art and artists, but a matter of systematic, disciplined, and quantifiable approaches. Note that Software Engineering encompasses both studying and applying such approaches.

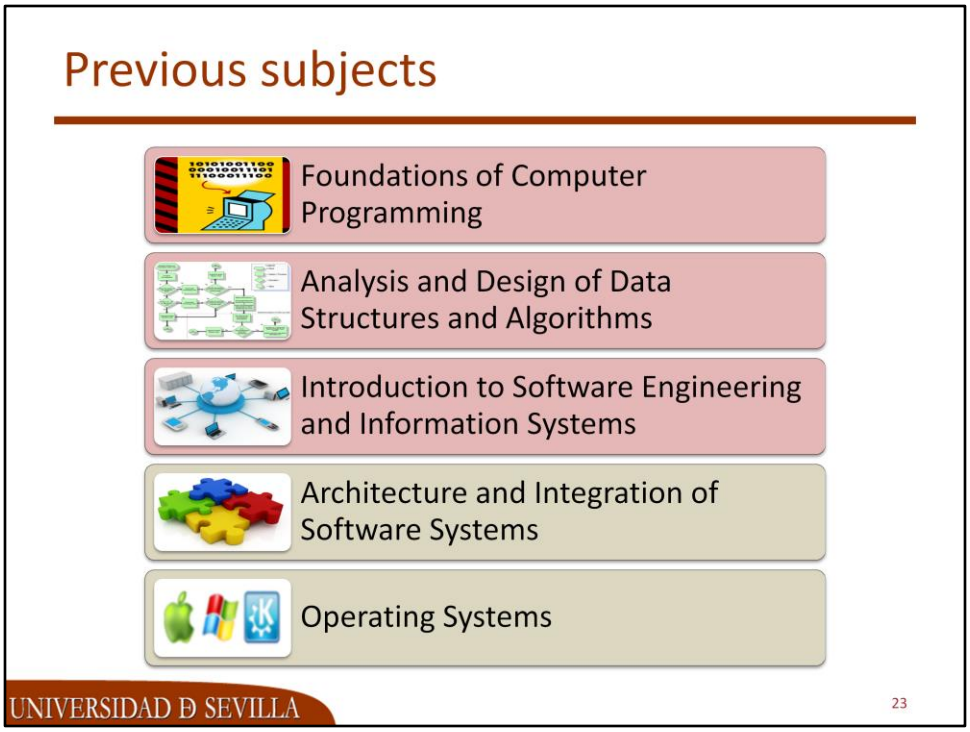
Our degree on Software Engineering provides an overview of Software Engineering, but focuses on developing so-called Web Information Systems, or WISs for short. Later, we'll provide an insight into what a web information system is; so far, we just expect that you understand that they are business management systems that provide a web interface to their users.

Let's review the subjects

UNIVERSIDAD Ð SEVILLA

22

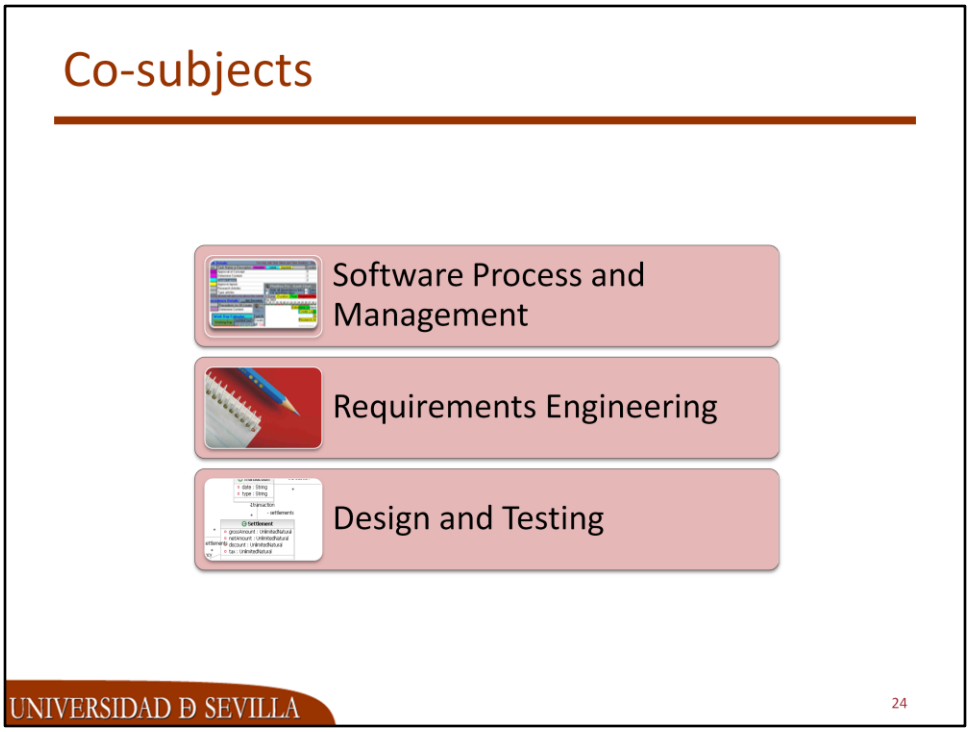
Right, it's now time to review the subjects that you're are expected to have already passed within the degree on Software Engineering, the subjects that you are expected to attend together with D&T, and the subjects that you're expected to attend next year. They all constitute the foundations of your degree on Software Engineering. There are many other complementary subjects that are intended to provide additional competences, but the core are the subjects that we're going to review in the next few slides.



These are the subjects that you're assumed to have passed before you enrol D&T. If you didn't pass any of these subjects, we strongly recommend that you first focus on passing them, and then come back to D&T.

- Foundations of Computer Programming: this subject teaches on the rudiments of Java, which is one of the most popular languages to implement typical web information systems.
- Analysis and Design of Data Structures and Algorithms: this subject teaches on designing data structures and algorithms to process them, i.e., data types.
- Introduction to Software Engineering and Information Systems: this subject provides an introduction to developing a simple web information system. It's a very good introduction to D&T!
- Architecture and Integration of Software Systems: this subject teaches that applications, particularly web information systems, can be viewed as components that can be integrated into your own applications; it elaborates on the technology required to achieve this goal.
- Operating Systems: this subject provides a good overview of how operating systems work and the services that they offer.

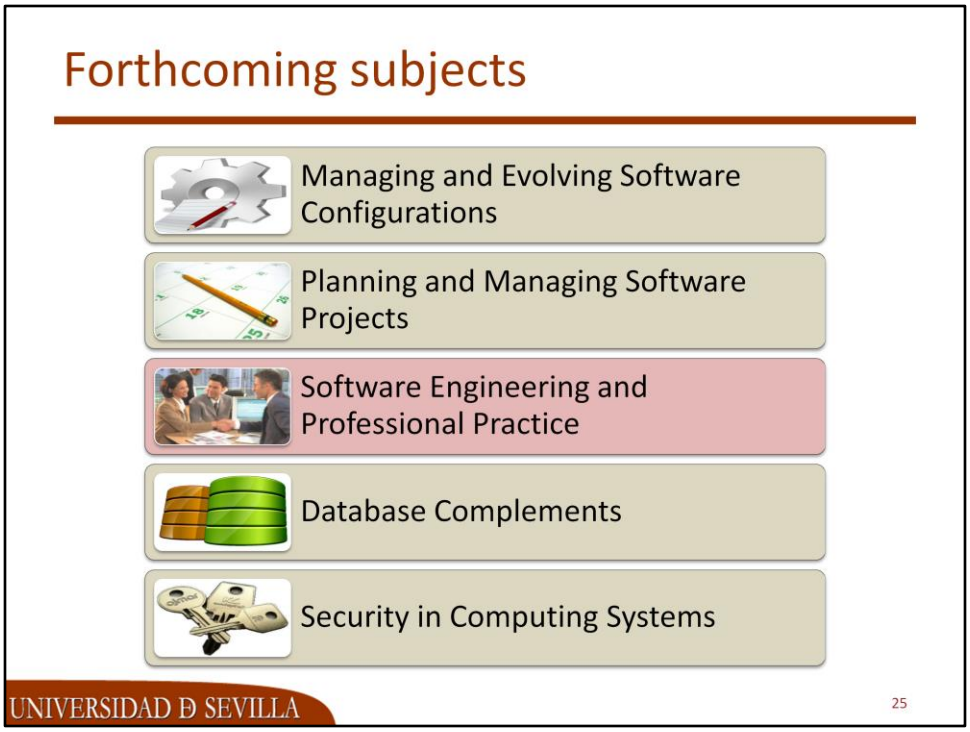
The subjects that are in red are absolute musts: definitely, it's not a good idea to enrol D&T unless you have already passed them before. The subjects in grey definitely provide good complements, but they are not a must.



These are the subjects that we strongly recommend that you should enrol this year, namely:

- Software Process and Management: this subject provides a broad introduction to software projects and how to manage them.
- Requirements Engineering: this subject teaches on how to model your customers' requirements.
- Design and Testing: this subject teaches on how to implement your customers' requirements in cases in which they are about a web information system; we'll provide additional details later.

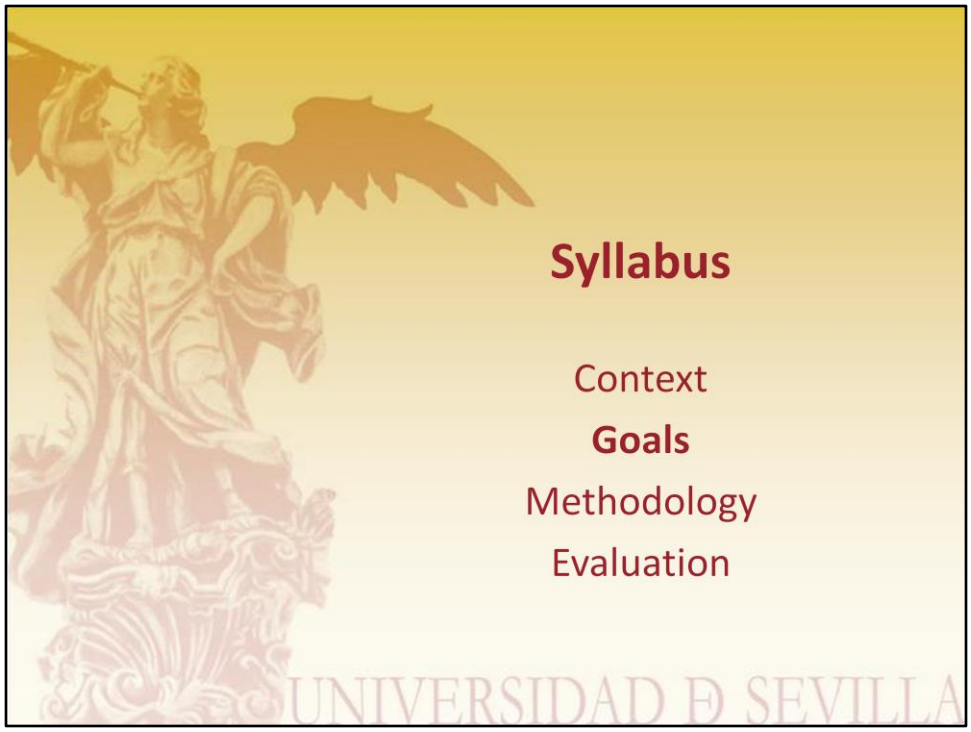
Since D&T must be taken the same academic year as Software Process and Management and Requirements Engineering, that implies that we can't assume that you're proficient with the knowledge that is delivered in these subjects. Where appropriate, we'll teach exactly what you need from these subjects so that you can successfully pass D&T. Don't worry about that.



And these are the subjects that you're expected to take next course:

- Managing and Evolving Software Configurations: this subject teaches on how to set up the different configurations that are involved in the development of a typical project, and how to manage the evolution of the artefacts that are produced or consumed by the project. By "configuration", we mean a piece of hardware and some software that are needed to carry out a stage of a project; by "artefact", we mean any object that is involved in a stage of a project, ranging from a model to a database schema or a Java class.
- Planning and Managing Software Projects: this subject provides additional details on managing software projects.
- Software Engineering and Professional Practice: this is expected to be the "last subject"; its goal is to put your knowledge into practice and teach you on how professional Software Engineers behave and work in their workplaces.
- Database Complements: this is an elective subject that provides an insight into database technology.
- Security in Computing Systems: this is an additional elective subject whose focus is on securing information systems.

Definitely, it's not a good idea to enrol the subjects in red unless you have already passed D&T; the other subjects provide good complements and they can be taken at any time.



Let's go on! Now, we know about the organisation and the context within which D&T falls, but we know very little about the subject itself. Let's start with the goals.

Our goals in D&T

Learn how to produce a typical small- to mid-scale web information system using industrial-strength components, tools, and methods.

UNIVERSIDAD Ð SEVILLA

27

Right, we talk about goals, in plural, but, actually, we only have one goal: when you pass this subject, you're expected to have learnt how to produce a typical small- to mid-scale web information system using industrial-strength components, tools, and methods. Pretty simple!

General competences

G05. Ability to design, develop, and maintain systems, services, and applications using the methods of Software Engineering as a tool for quality assurance.

G09. Ability to solve problems and make decisions with initiative, autonomy, and creativity. Ability to learn to communicate and get the knowledge across, skills, and abilities of the Technical Computing Engineer profession.

This goal is a summary of the general and specific competences that the degree on Software Engineering defines for this subject. This slide reports on the general competences as they are defined by the degree. Please, read them carefully; you must prove that you've learnt this competences in order to pass this subject. Does it mean that you may fail this subject because you aren't able to solve problems autonomously or you aren't creative? Yes, that is what they mean!

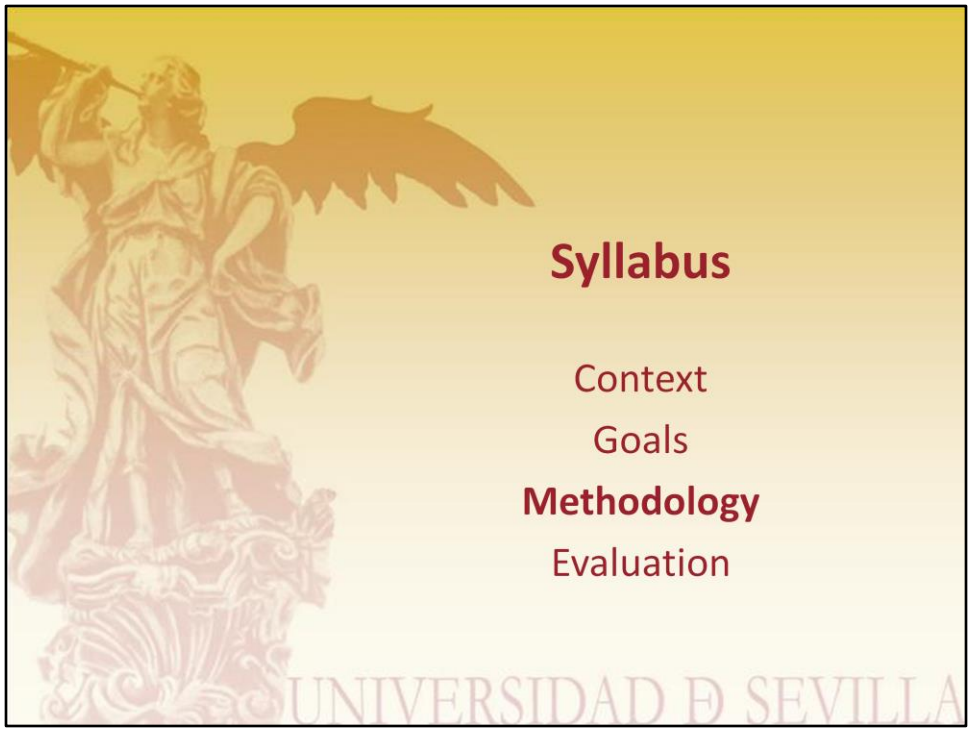
Specific competences

E25. Ability to develop, maintain, and evaluate software systems and services that satisfy the user requirements and behave reliably and efficiently, are affordable to develop and maintain, and meet quality standards, applying the theories, principles, methods, and practices of Software Engineering.

E28. Ability to identify and analyse problems and design, develop, implement, verify, and document software solutions based on adequate theories, models, and techniques.

E30. Ability to design appropriate solutions in one or more application domains using Software Engineering methods that integrate ethical, social, legal, and economic issues.

And this slide shows the specific competences that you're expected to learn. Please, read them carefully since you can't pass the subject unless you learn them and prove that you have learnt them.



It’s now time to provide a few details on the methodology we’re going to use.

What we expect from you (I)

To be interested in working as a software engineer, with a focus on developing web information systems

UNIVERSIDAD Ð SEVILLA

31

We start by making it explicit what we expect from you. First, and foremost important: we expect you to be interested in working as a software engineer, with a focus on developing web information systems.

What we expect from you (II)

To be active, to have a strong sense of ethics, and to know how to behave in a formal setting.

UNIVERSIDAD Ð SEVILLA

32

Second, but also very important: we expect you to be active (that is, problem-solving, dynamic, curious, inquiring), to have a strong sense of ethics (in particular, we expect you not to tolerate cheating, including, obviously, plagiarism), and to know how to behave in a formal setting (that is, addressing people, wearing appropriate attires, and observing appropriate hygiene and etiquette customs).

What we expect from you (III)

To have good abstraction skills, good modelling skills, a good command of the Java platform, and a good command of the Eclipse integrated development environment.

We also expect you to have good abstraction skills, good modelling skills, a good command of the Java platform, and a good command of the Eclipse integrated development environment.

What we expect from you (IV)

To have a *little* experience at developing simple web information systems

Finally, we also expect you to have a little experience at developing simple web information systems. Not much experience regarding this, believe us. The experience from subject “Introducción a la Ingeniería del Software y los Sistemas de Información” is a good starting point to pass this subject.

The naked truth

UNIVERSIDAD Ð SEVILLA

35

Unfortunately, there are many cases in which our expectations aren't met. We have found, and find recurrently, that many students aren't interested in working as developers of web information systems, and that they just don't know why they have enrolled this subject (or even this degree); those students are totally passive; they don't wish to leader anything, but expect everything to be done for them; those students cheat and plagiarise their mates, and they find it "natural"; they don't know how to behave in a formal setting because they address their lecturers informally or wear t-shits, shorts, and flip-flops in their lectures. We have also found a few students (very few, but they exists), who don't even observe the minimum hygiene and etiquette customs.

Europe's motto

In varietate concordia

UNIVERSIDAD Ð SEVILLA

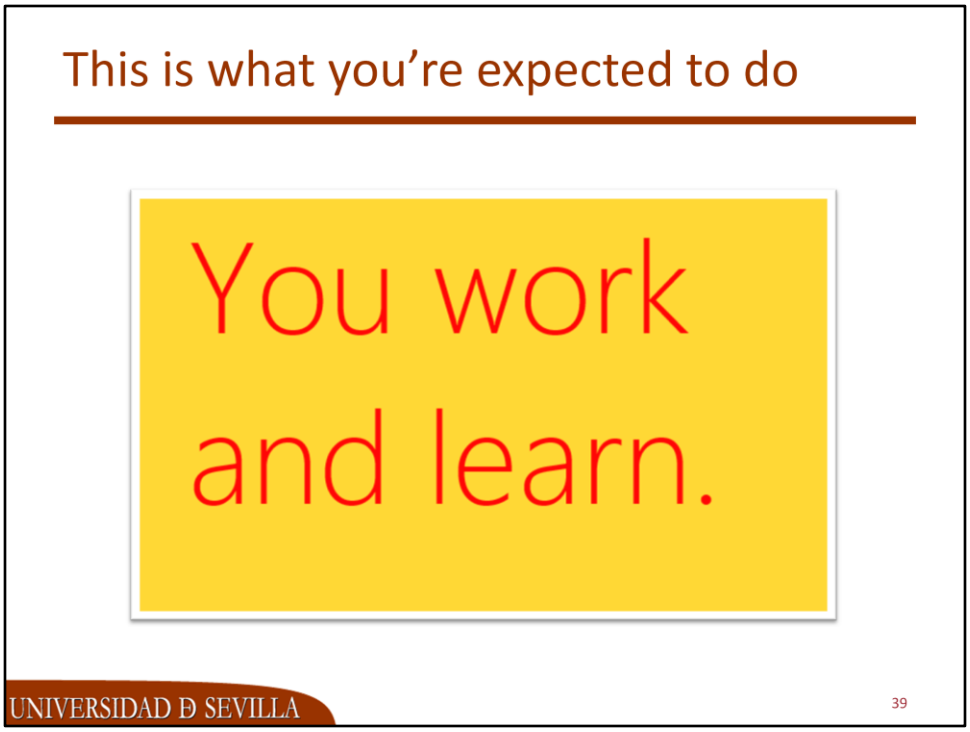
36

That means that we have a variety of students with very diverging motivations. Paraphrasing Europe's motto, our methodology has to unite them all in their diversity.

An extremely simple methodology

That means that our methodology must be very simple. Extremely simple.

This what we, the lecturers, do: we teach and evaluate.



This is what you, the students, are expected to do: you work and learn.

And this is what we strongly disprove: we don't tolerate cheating. If you're planning on cheating, please, drop out. That's not tolerated in a University context. Unfortunately, we have found a few cases of students who cheated in this subject. That's not a good piece of news!

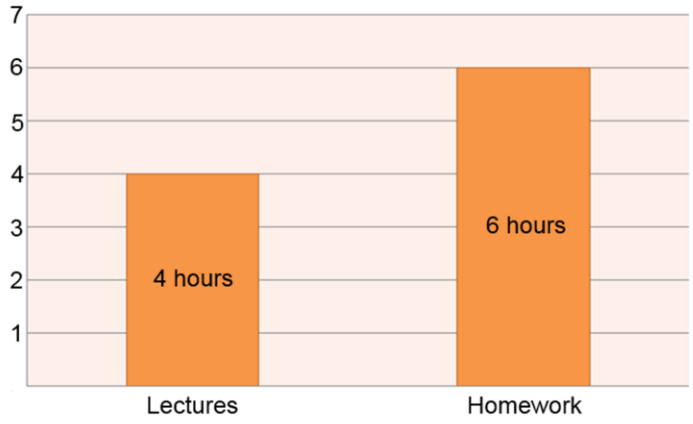
Consult our work programme!

UNIVERSIDAD Ð SEVILLA

41

We have a work programme that is available at the USE's e-learning platform. Please, download it and consult it as soon as possible. The work programme provides a detailed description of the activities on which we are going to work every day. Please, note that the work programme is the spine of D&T and that it isn't likely to change once it is published; mistakes are obviously corrected, but they aren't likely to happen; on authorised student meetings or strikes, it's re-scheduled, but the work programme's always followed. Please, download it and keep it close to you.

Expected workload

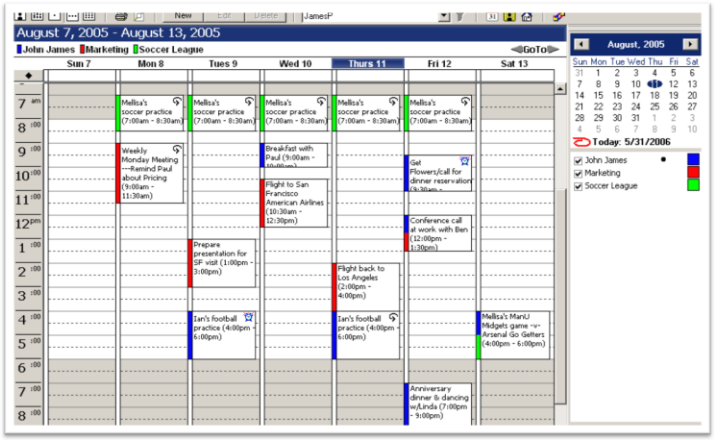


UNIVERSIDAD D SEVILLA

42

The expected workload is shown in this slide: you're expected to work four hours a week in our lectures plus six hours a week at home. That amounts to 10 hours per week or 15 x 10 = 150 hours per semester, that is, 6 ECTS per semester or 12 ECTS per academic year.

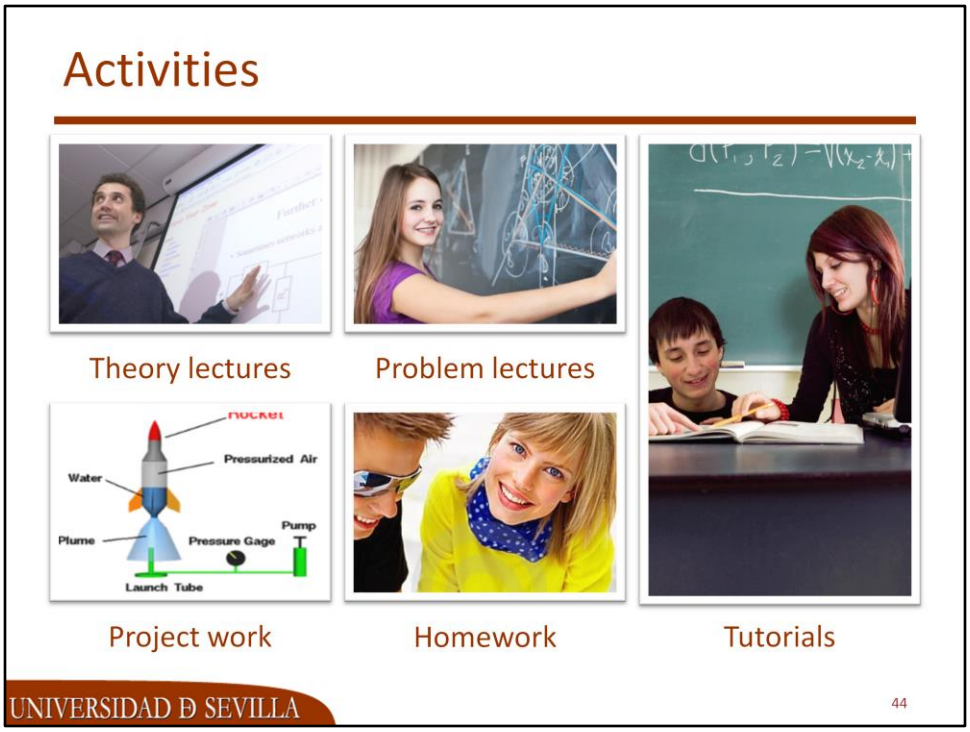
Instantiate *your* work programme



UNIVERSIDAD D SEVILLA

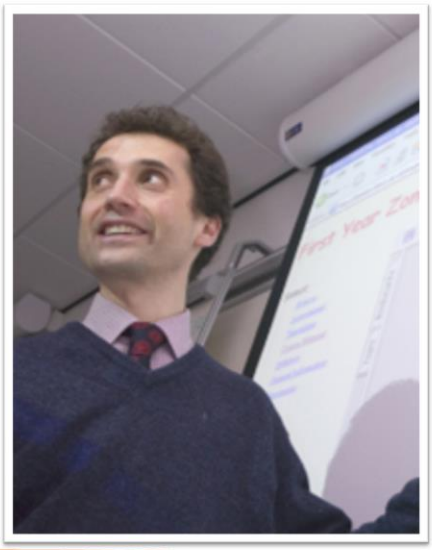
43

Please, note that *our* work programme is generic. You should instantiate *your* own work programme building on ours. Instantiating your work programme means that you must allocate six hours per week to do homework regarding this subject, plus, obviously, the four lecturing hours that are pre-scheduled. And you must obviously take into account that you have lectures and homework regarding other subjects, and that you should have a little time to meet your family, your friends, and have fun, too.



The work programme consists of five kinds of activities: theory lectures, problem lectures, project work lectures, homework, and tutorials.

Theory lectures



- The lecturers elaborate on a topic
- They use slide-based presentations
- The slides have detailed annotations
- There's a little time for questions and answers
- Typically, one theory lecture per lesson

UNIVERSIDAD Ð SEVILLA

45

Theory lectures are the primary source of knowledge. In a theory lecture, a lecturer elaborates on a topic; the goal is to provide a good understanding of the foundations behind that topic. These lectures consists of slide-based presentations, like this one; the slides have detailed annotations that summarise what the lecturers say and they are available to you at the USE's e-learning platform. Usually, there's a little time for questions and answers at the end of every lecture; please, note that you don't have to wait till the end of a lecture to ask your questions; it's better to raise your hand and pose your questions during the presentation. Typically, there's one theory lecture per lesson.

Problem lectures

- The lecturers publish a number of problems
- The lecturers assign the problems to students
- The students present their solutions
- The lecturers provide feedback
- Typically, one or two problem lectures per lesson

UNIVERSIDAD Ð SEVILLA

46

Problem lectures are different: the lecturers talk very little in problem lectures. They publish a number of problems and they assign each problem to a pair of students (we recommend that you work in pairs, but you can work alone if you wish). The students are expected to solve the problems and present their solutions in public; the lecturers will comment on their solutions and will provide corrections if necessary. Typically, there are one or two problem lectures per lesson.

Project work lectures

- The lecturers publish the description of some small-sized projects
- The students must work on them and deliver them timely
- The work must be carried out in group
- Typically, one or more project work lectures per lesson

UNIVERSIDAD Ð SEVILLA

47

In these lectures, you're expected to bring your laptop to the laboratory and work on several projects during the course. The statements that describe the projects will be available in WebCT. We'll provide additional details on the projects later. You must work on your projects in groups of several students. Typically, there's one or more project work lectures per lesson.

NOTE: if you don't have a laptop, then you can use the computers that are available in our laboratories and use an OpenLab virtual computer (http://openlabs.us.es). We strongly recommend that you should have your own laptop, since this is like a phonendoscope for a doctor. Sooner or later, you'll have to buy a professional laptop.

Homework

- Reviewing lectures
- Working on problems
- Working on projects
- Prepare control checks

UNIVERSIDAD Ð SEVILLA

48

Homework's quite an important activity. Please, remember that you're assumed to spend four hours a week at your lectures, and six hours a week doing homework. Homework basically consists of reviewing the material delivered in the lectures, working on problems, working on your projects, and preparing your control checks (more on this later). Make sure that you allocate your homework regarding D&T in your work programme.

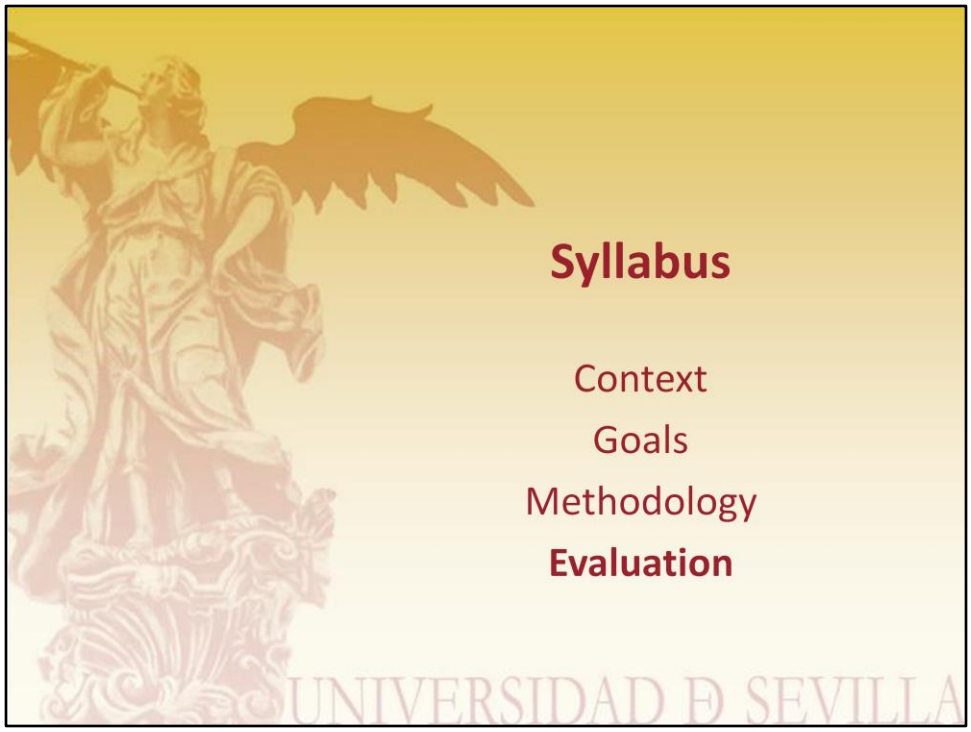
Tutorials

- Book your tutorials by email
- No tutorial should exceed 10-15 minutes
- Please, read the “On your tutorials” document

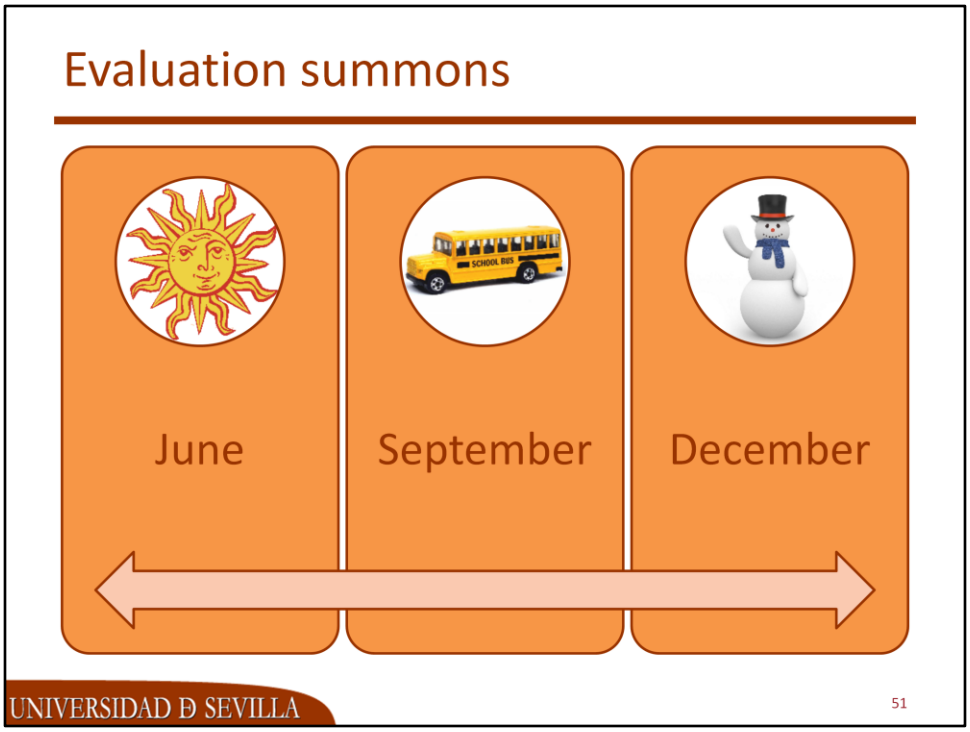
UNIVERSIDAD Ð SEVILLA

49

Tutorials are the last kind of activities. They aren’t pre-scheduled, you need to book them on demand. Book your tutorials by email and make sure that they don’t require more than 10-15 minutes. We have produced a document called “On your tutorials” that will definitely help you book them; please, read it carefully before contacting your lecturers.



We're close to the end of our syllabus. It's time to report on something that is very important for our students: how they will be evaluated.



In accordance to the current regulations, there are three evaluation summons: June, September, and December.

What do they consist in?

Deliverables

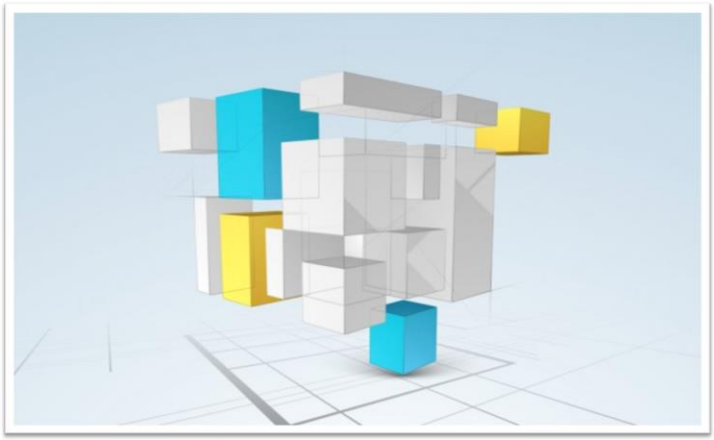
Control checks

UNIVERSIDAD Ð SEVILLA

52

The evaluations build solely on your deliverables and some control checks. First, we're going to present some core ideas and then will highlight the few differences amongst the summons.

Core ideas



Let’s start with the core ideas.

The deliverables

- They help put theory into practice
- The lecturers state the requirements to earn an A^+, an A, a B, or a C
- Deliver them by the deadlines
- Work in groups of up to six students

UNIVERSIDAD Ð SEVILLA

54

Along the course, the lecturers will release a number of statements that describe some projects on which you must work. They are the best way to put theory into practice. The lecturers will also state the minimum requirements that your deliverables must meet so that you can earn an A^+, an A, a B, or a C; if your deliverable doesn't meet the requirements to earn a C, then it will be awarded an F. Please, consult the work programme to find the deadlines for each deliverable. You can work alone on your deliverables, but it's strongly recommended that you recruit and set up a workgroup of up to six students. Note that the workload is independent from the number of students in your group so we strongly recommend that you should not to work alone or in tiny groups.

The hackathon

- The last deliverable is a hackathon
- This deliverable is devised by the students, but must get approved by a lecturer
- Start thinking of it!!

We'd like to highlight a "special" deliverable, to which we refer to as the hackathon. It's special because the lecturers do not provide a requirements specification for that deliverable. The students must devise it and present it to a lecturer, who must approve it. Start thinking of your hackathon as soon as possible!

The control checks

- They check if the deliverables correlate to a student's knowledge level
- They must be passed individually, without any collaboration

UNIVERSIDAD Ð SEVILLA

56

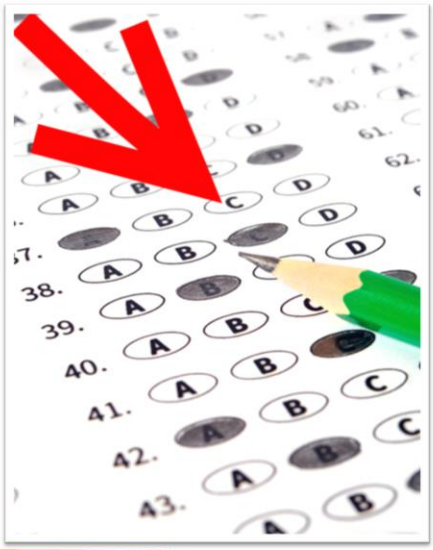
Unfortunately, we can't blindly trust you: we know there are students who cheat and we must ensure that no student who cheats passes this subject. There are also some students who don't meet the goals of this subject, even though they have produced every deliverable in time. We are talking about students who manage to assemble a project that works, but don't actually understand what they've done; or students who literally gobble up their partners, who cannot learn because their mates don't give them the opportunity to work; or students who search for a "gobbler" partner and ride his or her coattail. That means that we have to make control checks. A control check is a meeting in which the students, individually, have to make a few changes to their deliverables: changing the domain models, adding new use cases, ... OK, you're not expected to know what a domain model or a use case is, yet. Simply put: we assume that you've worked on your projects and that you've learnt from them; we'll ask you to make changes to your project; if you've actually worked on your projects and you've actually learnt from them, then passing your control checks should be like ringing a bell. If you do not pass them, the lecturers might investigate your case in an attempt to discern the reason why you didn't pass them.

Each deliverable or control check is accompanied by a specific evaluation procedure. There are, however, some sufficient conditions for failure, namely:

- An interaction with your system results in an HTTP error.
- An interaction with your system results in a panic.
- Submitting a form with wrong data is not detected.
- There are missing CRUD use cases.
- An actor can list, edit, or delete data that belongs to another actor.

If your deliverable or your control meets any of the above conditions, it'll be considered failed automatically. Note that the above conditions are sufficient conditions for failure; passing them does not entail that your deliverable or control check is passed; you must also pass the specific evaluation procedure.

The critiques



- Lecturer mistakes
  - Mistakes are checked
  - Marks are corrected
- Student mistakes
  - Lecturers may ask students to fix them
  - Maximum mark C or PASSED

UNIVERSIDAD Ð SEVILLA

58

Theirs is a critique after each evaluation. Generally speaking, a critique helps correct evaluation mistakes. Exceptionally, if you have failed a deliverable or a control and the lecturer who has evaluated it thinks that you might fix it in a reasonable time, he or she will let you know. If you manage to fix your deliverable or your control during the critique, you may pass it but your maximum mark will be C for deliverables and PASSED for controls.

Summon differences

UNIVERSIDAD Ð SEVILLA

59

Let’s now delve into the few differences amongst the June summon and the other ones.

June summon - Deliverables

- A deliverable per lesson (six projects)
  - D01 is optional
  - D02-D12 are mandatory
- Check the deadlines in our work programme
- Workgroups can be re-configured for D01, D02-D07, and D08-D12

UNIVERSIDAD Ð SEVILLA

60

In June, you have to produce twelve deliverables regarding six different projects; each deliverable's associated with a lesson and must be produced by a deadline that is specified in our work programme. Deliverable D01 will not be evaluated since its only purpose is to help you get familiar with the subject, the delivery procedure, and your mates. Recall that you're expected to work in groups to produce your deliverables: during the June summon, you can re-configure your group when you start working on deliverable D02 or when you start working on deliverable D08. No other re-configurations are allowed.

June summon – Control checks

- Two control checks
  - By the end of the autumn semester
  - By the end of the spring semester
- Check the dates in our work programme

You also have to pass two control checks: one by the end of the autumn semester and another by the end of the spring semester. Check the dates in our work programme, please.

June summon – Grading general rule

- If you happen to
  - Pass the controls
  - And the deliverables
- Then your grade will be
  - The rounded average grade of your deliverables

The general grading rule in June is as follows: if you pass both control checks and every deliverable, then your grade will be the rounded average grade of your deliverables.

June summon – Grading exception #1

- If you
  - Earn at least a B in deliverables D02-D07
  - Earn at least an A in deliverables D08-D12
  - Pass the spring control
- Then your grade will be
  - The rounded average grade of your deliverables
- Even if you have failed
  - Some deliverables
  - The autumn control check

UNIVERSIDAD Ð SEVILLA

63

There are two exceptions to the general rule. Let's see the first one: if you earn at least a B in deliverables D02-D07, you earn at least an A in deliverables D08-D12, and you pass the spring control check, then your grade will be the rounded average grade of your deliverables even if you have failed some deliverables and/or the autumn control check. This gives our students a chance to fail a part of the evaluation, but pass the subject in June as long as their performance is very good.

June summon – Grading exception #2

- If you
  - Earn at least a B in deliverables D02-D07
  - Earn at least an A in deliverables D08-D12
- But you
  - Don't pass the spring control
- Then
  - You can sit for an exceptional control check in September
  - If you pass it, your grade will be the rounded average grade of your June deliverables
  - Else you'll fail the subject
- Even if you have failed
  - Some deliverables
  - The autumn control check

UNIVERSIDAD Ð SEVILLA

64

The second exception is as follows: if you earn at least a B in deliverables D02-D07, you earn at least an A in deliverables D08-D12, but you don't pass the spring control check, then you can sit for an exceptional control check in September even if you have failed some deliverables and/or the autumn control check. If you pass that exceptional control check, then your grade will be the rounded average grade of your June deliverables; else you'll fail the subject. That is, you don't have to work on new deliverables for the September summon; you only have to prepare your control check in this case.

June summon – All other grading cases

- In all other cases, you fail the subject in June's summon
- Work harder, train more, and pass the subject in September or December

In all other cases, you fail the subject in June. There's only one thing you can do: work harder, train more, and pass the subject in September or December.

September & December – Deliverables

- Six deliverables, each of which of which deals with a different project
- Similar in requirements and effort to June's projects
- Workgroups must be stable in these summons

The evaluation and grading in the September and the December summons is as follows: the students will have to produce six deliverables, each of which consists in a project that will be similar in requirements and effort to the projects requested for the June summon. The students can work in groups of up to six people, but these groups must be stable; that is, once a group of students is set up, they must work together on all of the deliverables.

September & December – Controls

- Only one control per summon
- Very similar to the spring control check

In September and December, there is only one control check. It'll be very similar to the spring control check.

September & December – Grading

- If you
  - Pass every deliverable
  - Pass the control check
- Then your grade will be
  - The rounded average of the grades awarded to your deliverables
- Otherwise
  - You fail the subject

UNIVERSIDAD Ð SEVILLA

68

Grading is very simple in September and December: if you pass every deliverable and the control check, then your grade will be the rounded average of the grades awarded to your deliverables; else you'll fail the subject.

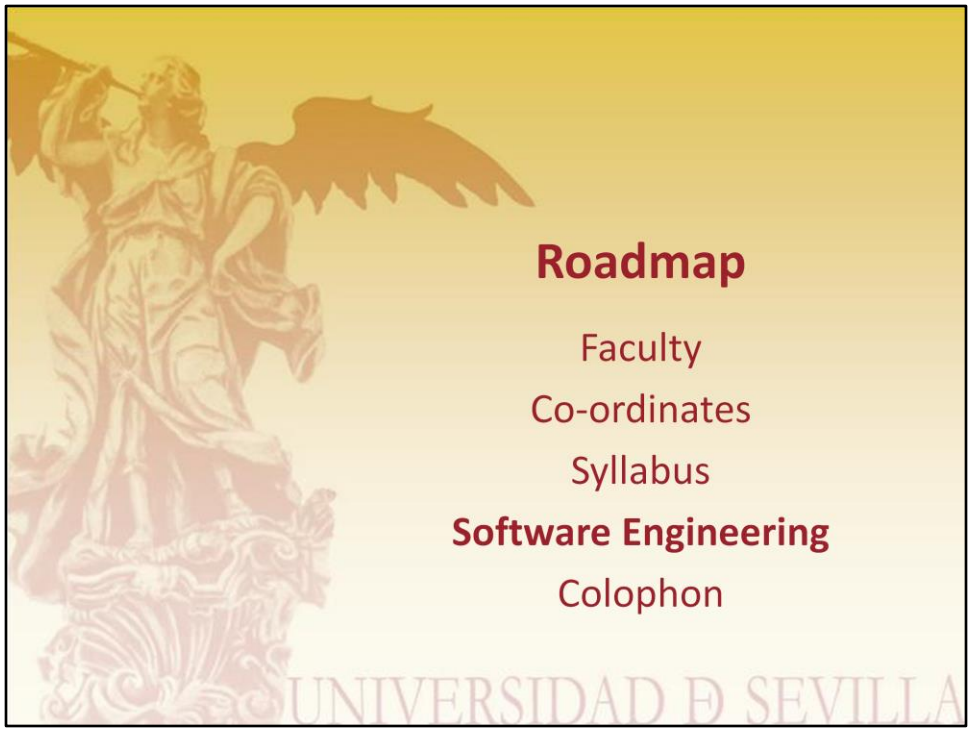
September & December – Exceptions

(*) Just see the exception regarding students who sit for an exceptional control check in September

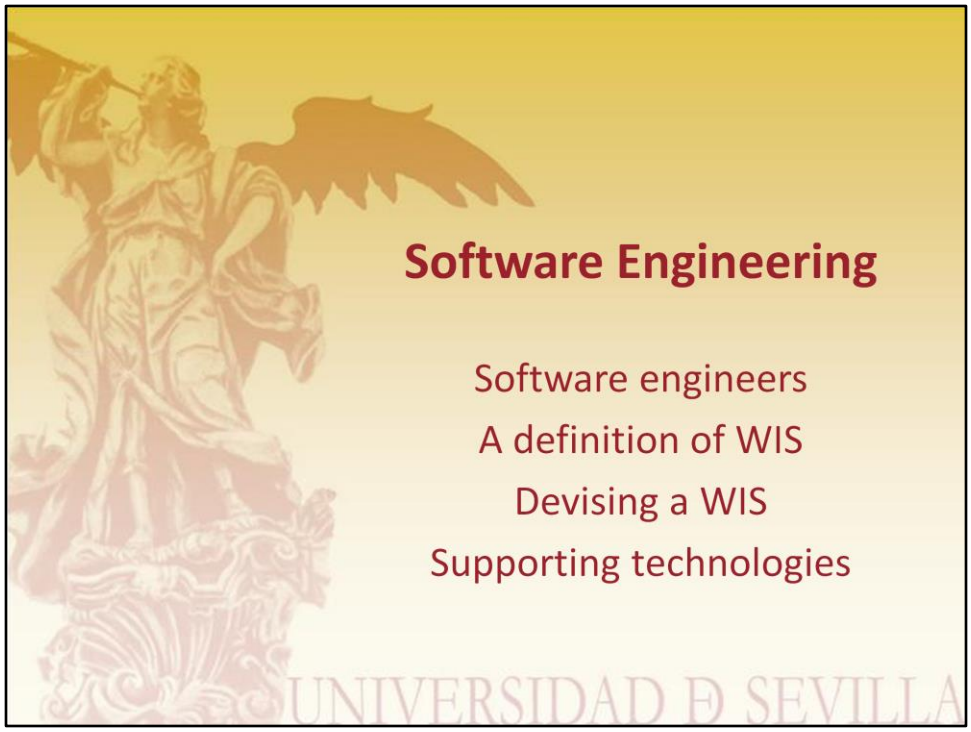
UNIVERSIDAD Ð SEVILLA

69

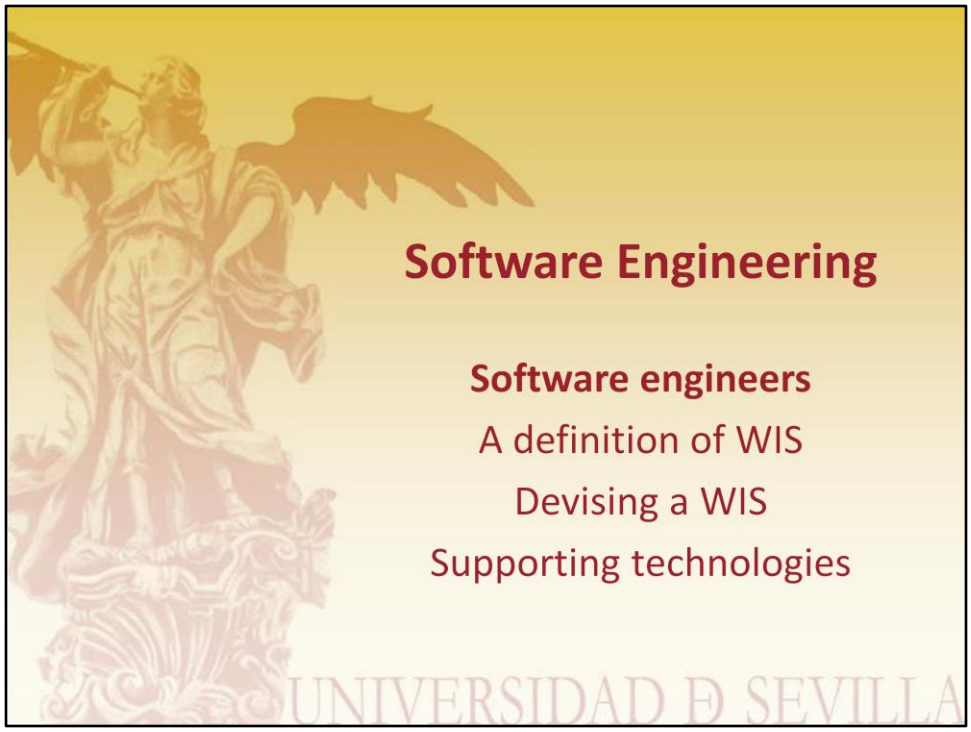
There's no exception to the grading rule in September or December, except the exception that allows some students to sit for an exceptional control check in September.



When we started this lecture, we mentioned that our goal was to summarise the syllabus, but we think that we've reported on almost everything. Please, consult the syllabus that is available at the USE's e-learning platform just to make sure that we didn't miss any important piece of information. In the previous section, we've talked several times about software engineers, web information systems, and the like. Unfortunately, our experience proves that many of our students don't actually know what these terms refer to. So we think that it's the right moment to spend a few minutes at reflecting and discussing on them.



We'll set off with a few reflections on what a software engineer is; then we'll provide a definition of web information system (or WIS for short), and will talk a little on how they are devised; we'll conclude with a short review of the technologies that we're going to use.



Come on! Let's start with our reflections on software engineers.

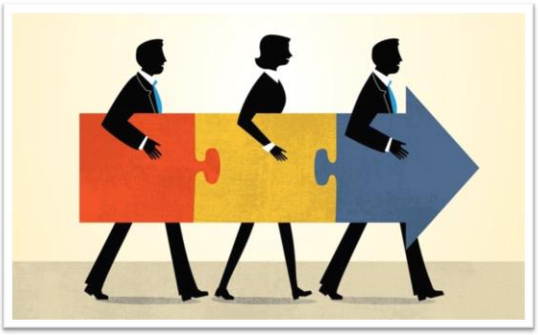
What's a Software Engineer?

UNIVERSIDAD Ð SEVILLA

73

This is a good question: what's a software engineer?

This is a good definition



A professional who is schooled and skilled in the application of an engineering discipline to the creation of software systems

UNIVERSIDAD Ð SEVILLA

74

This is a good definition: it's a professional who is schooled and skilled in the application of an engineering discipline to the creation of software systems.

It’s not a programmer

A programmer writes code according to a specification; a software engineer creates the specifications that a programmer has to implement

UNIVERSIDAD Ð SEVILLA

75

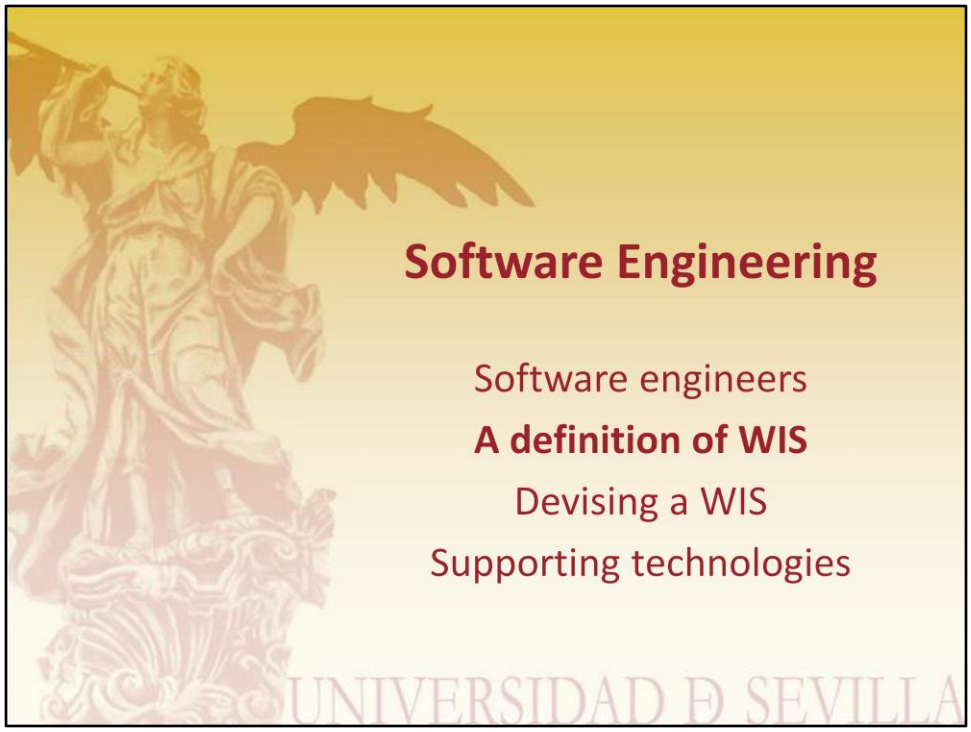
A software engineer is often confused with a programmer, but they are quite different professionals: a programmer writes code according to a specification; a software engineer creates the specifications that a programmer has to implement.

Our focus: web information systems

UNIVERSIDAD Ð SEVILLA

76

There are many kinds of software systems on which a software engineer can work. Our focus in this subject are so-called web information systems.



So it's time then to delve into a good definition of web information systems.

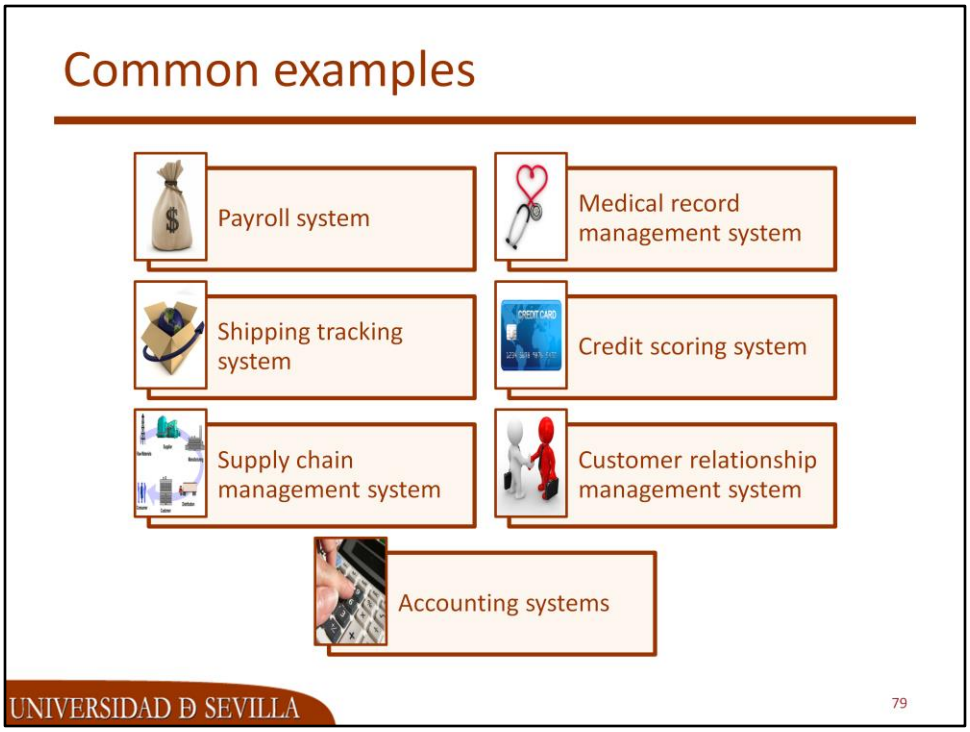
What's a web information system?

It's a software system that processes business data and provides a web user interface to perform business tasks

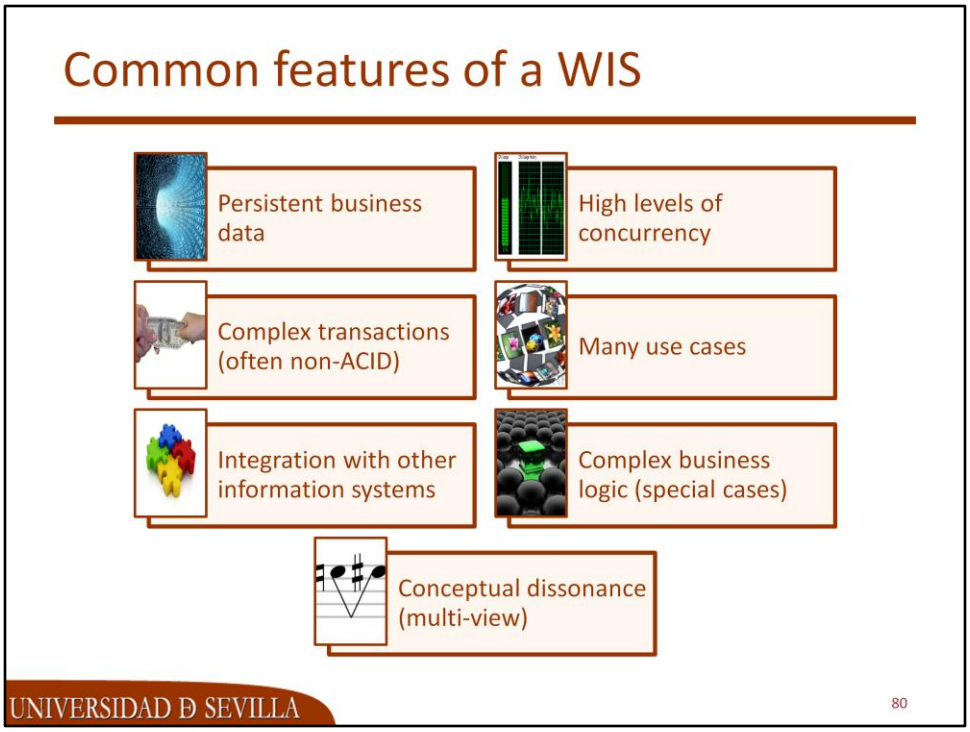
UNIVERSIDAD D SEVILLA

78

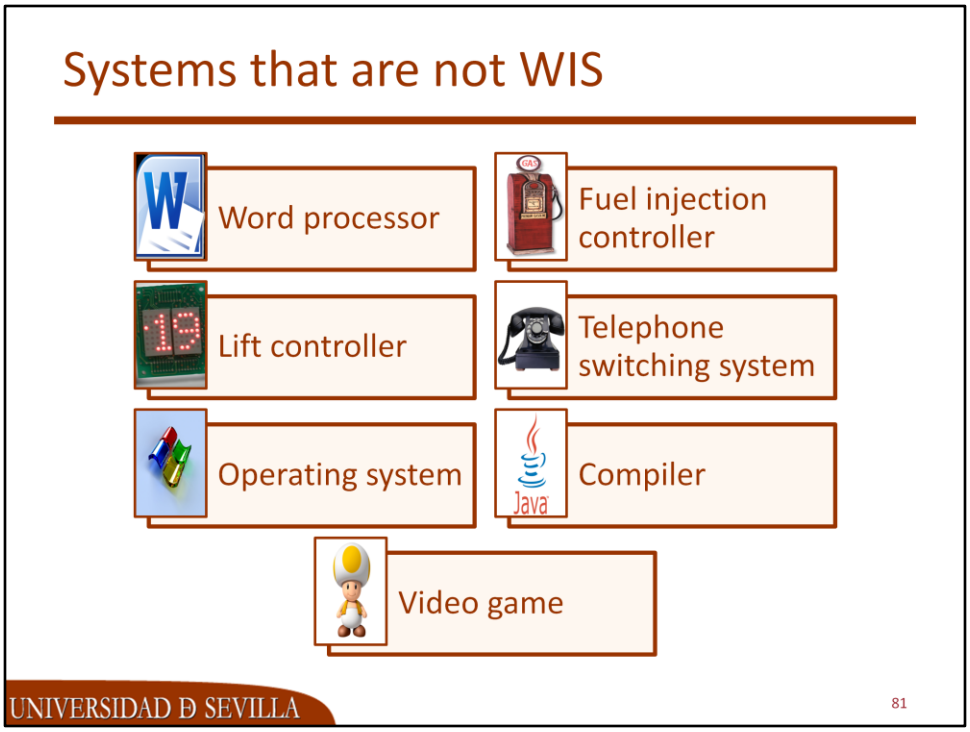
Here you are ours: it's a system that processes business data and provides a web user interface to perform business tasks, including administrative tasks. It's quite a simple definition, but powerful enough to make it clear the distinction amongst systems that can be considered web information systems and systems that cannot.



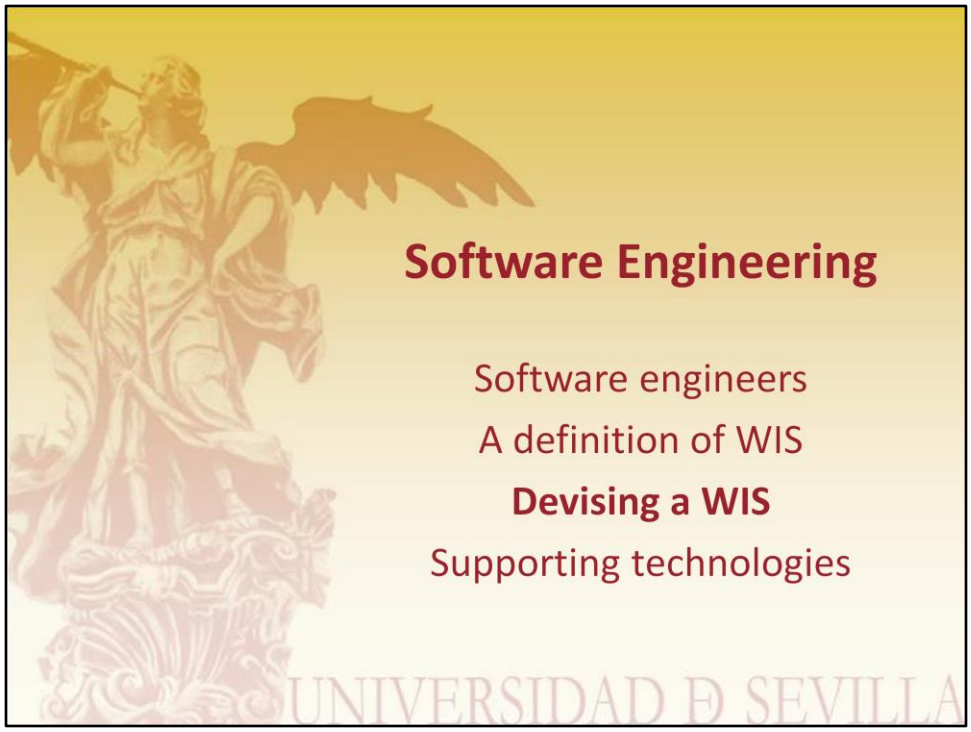
In this slide, we present several examples of common web information systems: payroll systems, medical record management systems, shipping tracking systems, credit scoring systems, supply chain management systems, customer relationship management systems, or accounting systems are good examples. It's not difficult to realise that all of the previous systems fit within our definition of web information system: they all are systems that provide a web user interface to carry out business tasks and manage large amounts of business data.



If we analysed them more carefully, we'd surely find a few more common features, namely: a) they have to deal with business data that must be stored in a database, that is, persistent data (employees, payrolls, patients, medical records, customers, packages, routes, shippings, financial data, and a long so on); b) they are inherently concurrent and the levels of concurrency are usually very high since they have to support many users working at the same time (several doctors in a clinic care of several patients at a time, several clerks analyse financial data at a time to decide on a credit, several accountants are working on provider payments at a time, and a long so on); c) they have to deal with complex transactions which aren't usually ACID, i.e., they require compensation actions (a customer pays for the shipping of a package but it doesn't reach its destination on time, so the system has to compensate for the delay; or a hotel is booked but absolutely no flight can be booked and the hotel needs to be compensated); d) typically, they offer a variety of user interfaces to interact with them, simply because they help perform a variety of business tasks; e) they have to be integrated with other information systems that are already working (for instance, the credit scoring system must be integrated with the accounting system, or the supply chain management system must be integrated with the shipping management system); e) their business logic's commonly complicated because of special cases (customers that must be awarded special offers or procedures that may change depending on the customer); and f) they are inherently conceptually dissonant, aka multi-view, which means that they provide different views on the same data (e.g., a clerk sees a customer as a number of account records, but another one sees that same customer as a number of figures that provide benefit forecasts).



Just to complete the picture, this slide presents a few examples of systems that are clearly not information systems: a word processor, a fuel injection controller, a lift controller, a telephone switching system, an operating system, a compiler, or a video game. The reason is very simple: none of them manages business data.



Let's now delve into how a typical web information system is devised. Please, pay a lot of attention since the next few slides summarise what we're going to do for from now on.

Starting point: your requirements

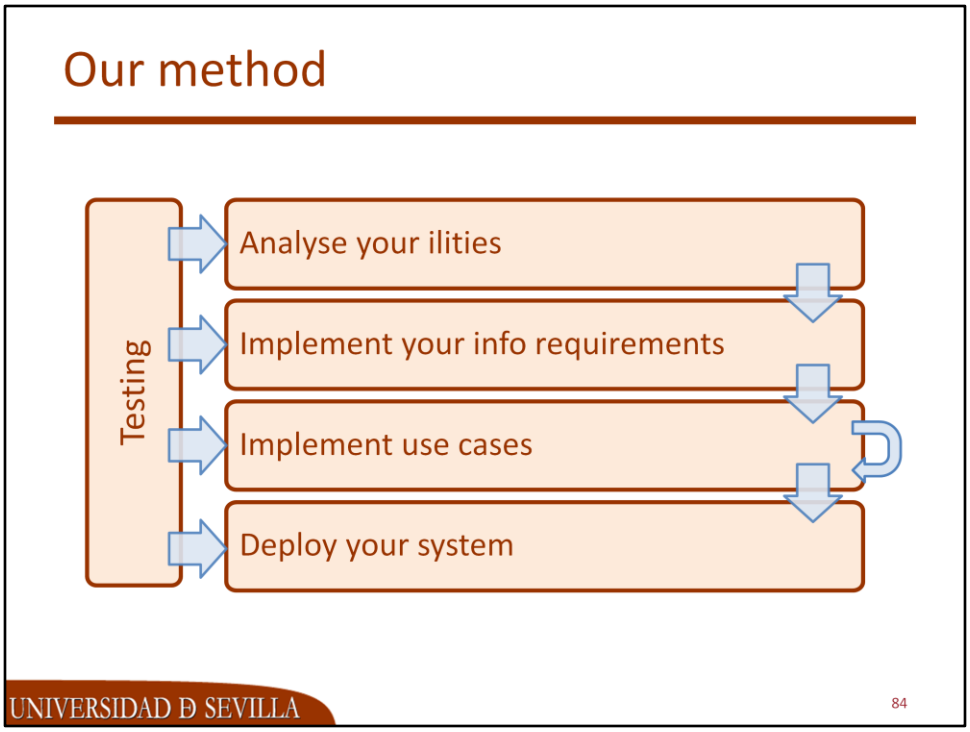
- Ilities
  - The non-functional aspects of a system
- Information
  - The data that a system has to process
- Use cases
  - The tasks that the actors can perform

UNIVERSIDAD Ð SEVILLA

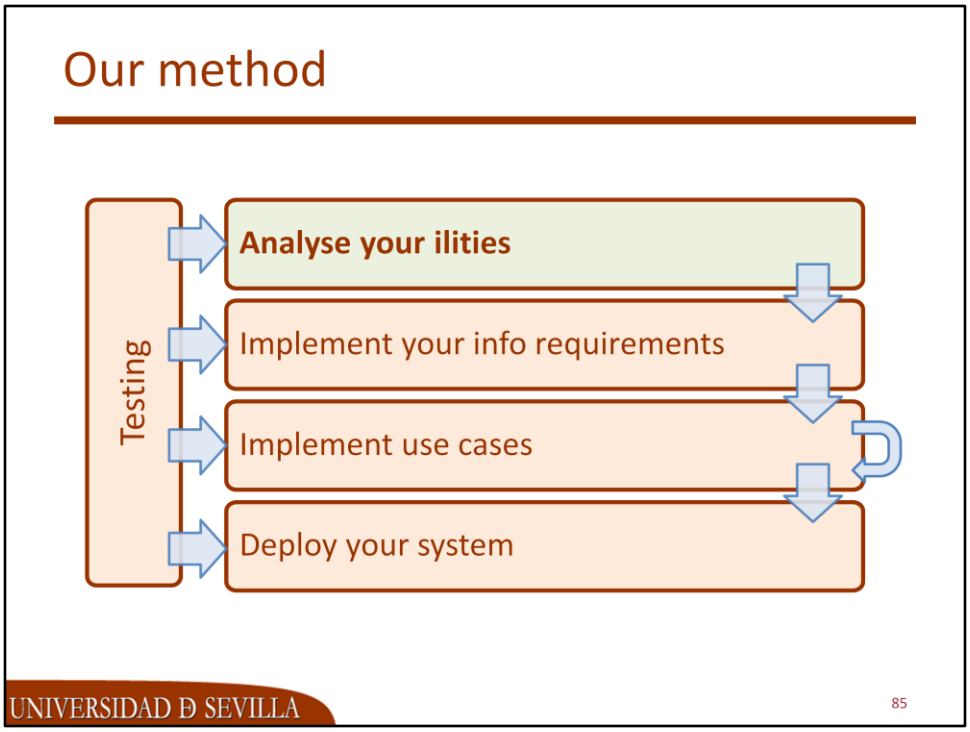
83

The starting point are your requirements elicitation documentation. Roughly speaking, this documentation provides you with a description of the non-functional requirements (aka ilities, extra-functional, quality, system-wide, or cross-cutting requirements), the data the system must handle (aka information requirements), and the tasks an actor can perform with the system (aka use cases).

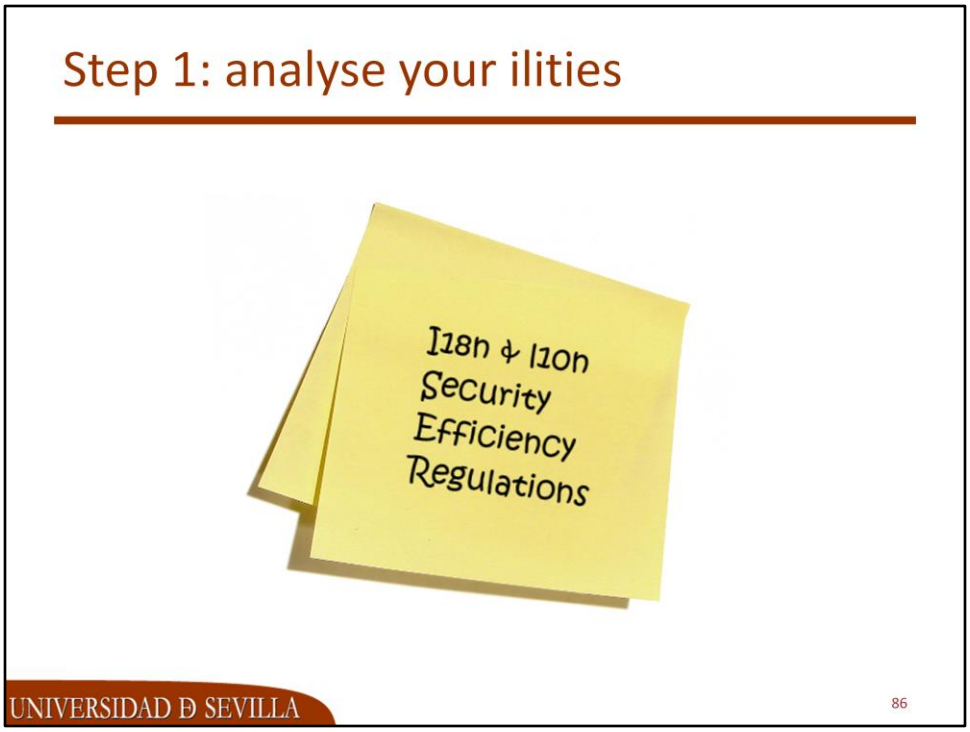
NOTE: you'll study every tiny detail about requirements in the co-subject entitled "Requirements Engineering".



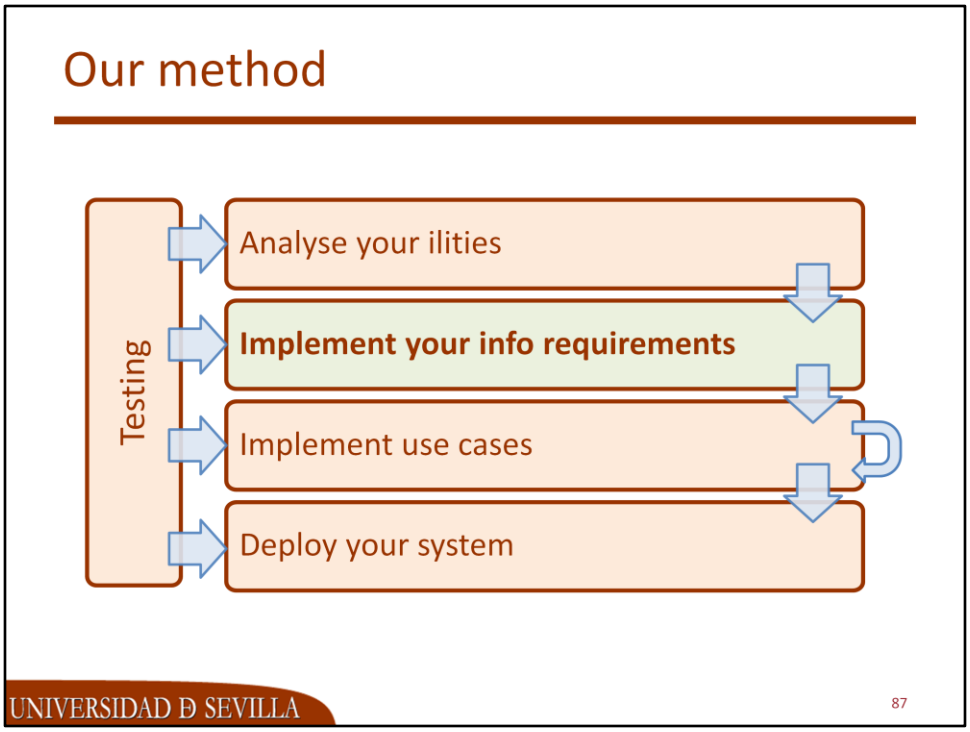
Starting from that documentation, this is the method that we're going to follow. In the following slides, we'll delve into the details.



The first step is to analyse your ilities.



Usually, the section that describes the ilities of a web information system is the last one in the requirements elicitation document, but they are very important since they may have an impact on almost everything. It's very important that you command them before your start working on a project. They are not like information or functional requirements; generally speaking, you can't prioritise them since they spread across every tiny part of your project; you generally can't add ilities to your project after it's finished; they are there all the time. In this subject, we'll focus on four kinds of ilities only: i18n & l10n, security, efficiency, and regulations. I18n means internationalisation and l10n means localisation; these ilities have to deal with adapting a system to a given culture, and it usually implies that the application must be able to deliver its interface in a number of languages, e.g., Spanish and English. Security has to deal with keeping the data safe and preventing unauthorised users from manipulating them. Efficiency has to do with consuming as little computing resources as possible. And regulations have to do with domestic and/or international laws that are applicable to software.



Once you've analysed your ilities and you command them, it's time to work on your information requirements.

Step 2: implement info requirements

UNIVERSIDAD Ð SEVILLA

88

The information requirements describe the data that has to be manipulated. We'll represent them by means of a domain model and a persistence model.

Step 2.1: create a domain model

UNIVERSIDAD Ð SEVILLA

89

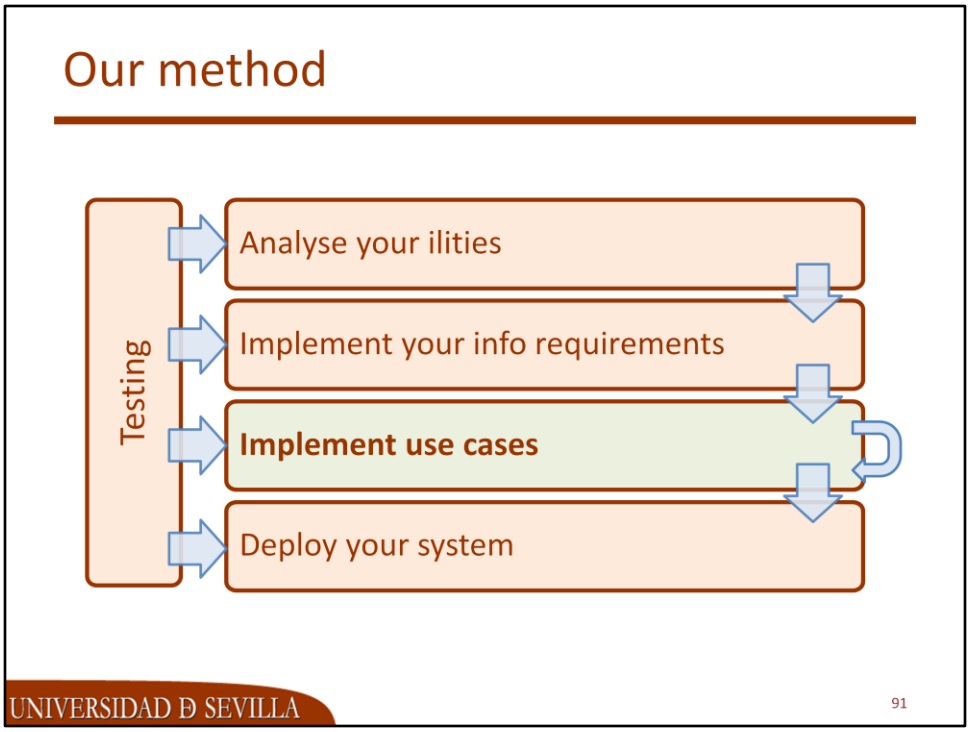
The domain model is a computer-based representation of the objects in a given domain. For instance, in a typical library domain, the domain model includes books, customers, loans, and so on. We use UML to represent domain models, but, in the end, we need to use Java to implement them.

Step 2.2: create a persistence model

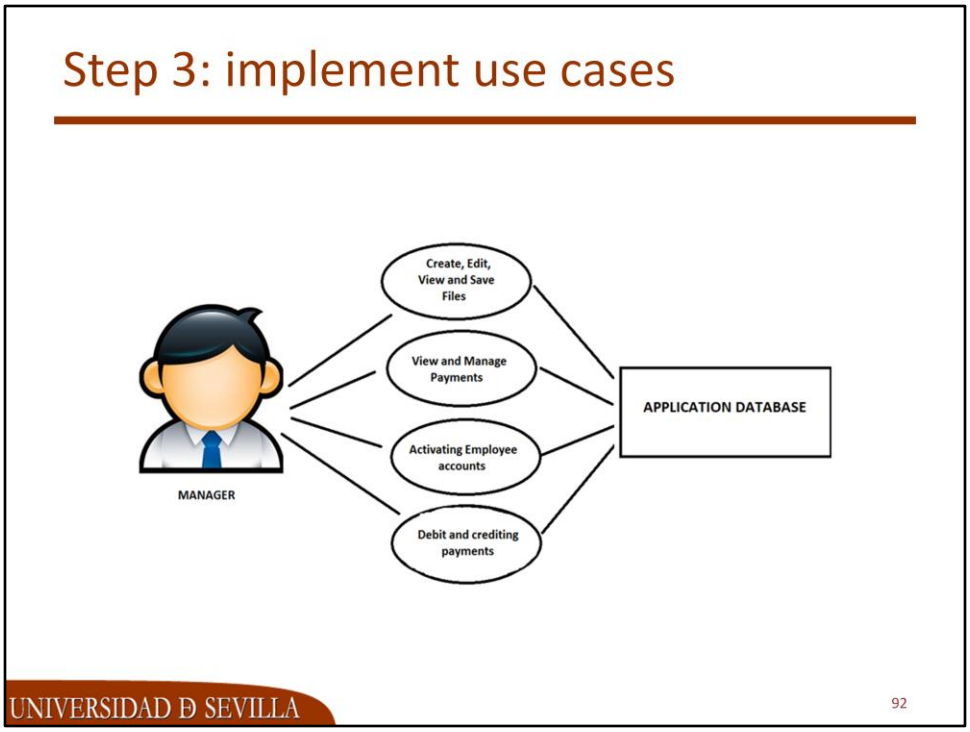
UNIVERSIDAD Ð SEVILLA

90

A persistence model is a specification of how data must be stored in or retrieved from a relational database. We'll use a technology called JPA to represent persistence models.

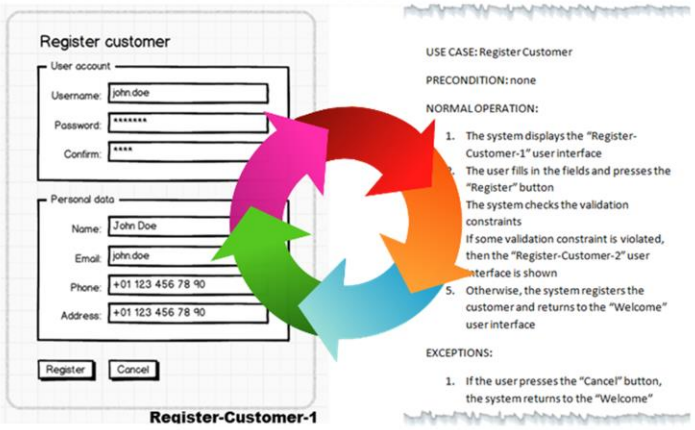


Once your information requirements have been implemented, it's time to work on your use cases.



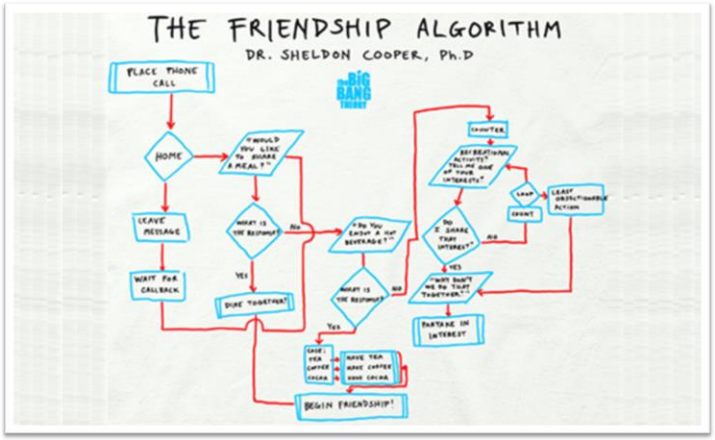
A use case describes how the user interacts with a system so that he or she can perform a business or an administrative task. Use cases are the most common technique to represent functional requirements in the context of web information systems. Note that a typical web information system may be composed of several dozens of use cases. That means that you'll commonly have to repeat this step of our method several times.

Step 3.1: design mock-ups/scripts



To specify use cases, we generally need to design mock-ups, which are sketches of the user interfaces, and scripts, which describe how the users interact with the mock-ups. They both are very useful to talk to customers since they allow them to have an overall idea of how the application will behave without having an implementation. Mock-ups and scripts usually need to be specified together. We recommend that you should use Balsamiq or Evolus Pencil to design your mock-ups; to specify the scripts, you can use a regular text editor.

Step 3.2: design algorithms



The majority of use cases in a typical web information system are CRUD, that is, they are related to creating, retrieving, updating, or deleting data. There are however use cases that are inherently algorithmic. In such cases, their description must be complemented with descriptions of the algorithms involved. Such descriptions must be simple, clear, and rely heavily on the domain model. You can use a regular text editor to specify your algorithms.

Step 3.3: design repositories/services

The next step is to design repositories and services. The repositories are classes that allow to persist data or to retrieve them; the services are classes that implement business rules and business logic. Repositories will be implemented using a technology called Spring Data and services will be implemented using Java.

Step 3.4: design views

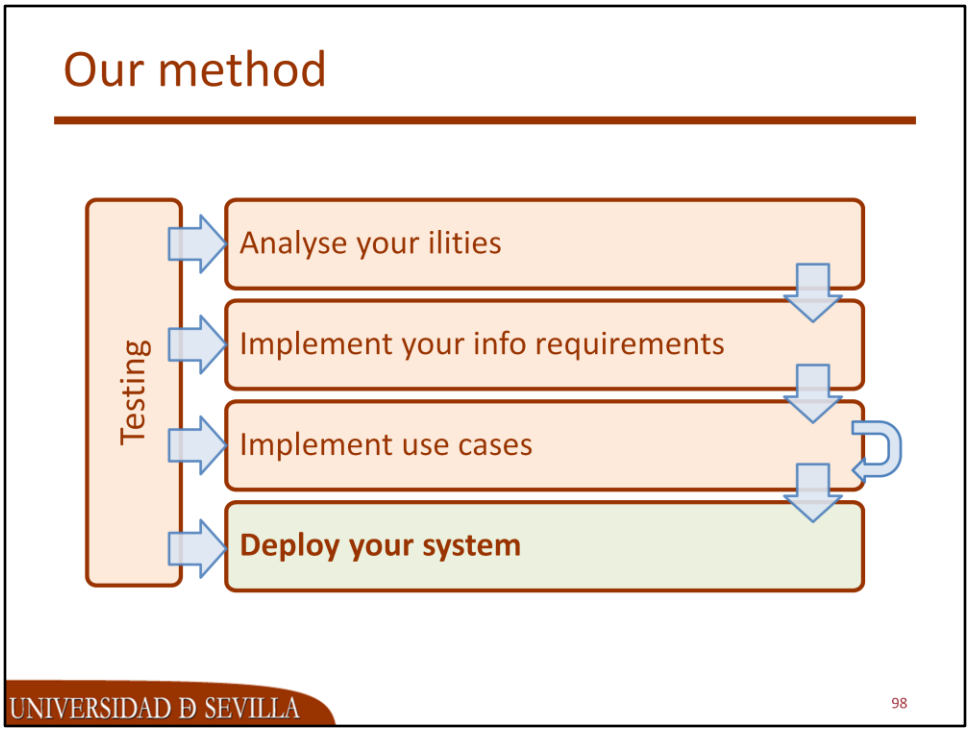
The next step is to design the views, which are computer-understandable documents that describe the mock-ups that we've designed previously. Views will be implemented using a language called JSP, a master-page framework called Apache Tiles, and several so-called tag libraries.

Step 3.5: design controllers

UNIVERSIDAD D SEVILLA

97

Finally, we have to design controllers, which are kind of orchestrators. When a user requests something from a web information system, the request is intercepted by a controller that orchestrates the services required to fulfil it and returns the appropriate view to render a user interface. We'll use a technology called Spring MVC and Java to implement controllers.



Once the use cases are implemented, it's time to deploy your system. In practice, it is common that systems are deployed partially, when a few use cases are implemented and ready to work. This is due to time-to-market constraints, since, very often, getting to the market before than other competitors makes much of a difference.

Step 4.1: create deployment artefacts

UNIVERSIDAD Đ SEVILLA

99

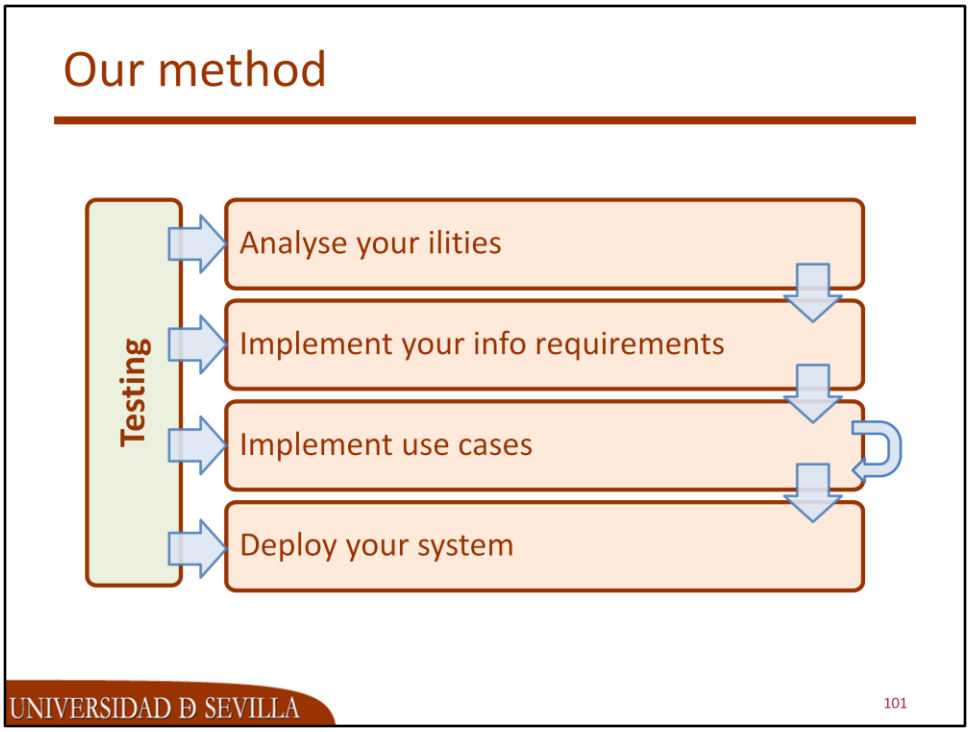
Deploying a system means creating a number of deployment artefacts…

Step 4.2: deploy to a cloud

UNIVERSIDAD Ð SEVILLA

100

… and then uploading them to an application server, which more often than not will run on a cloud.



Finally, it's time to report on testing.

Step 1-4: testing

The ultimate goal of testing is to increase the quality of your software. Note that testing isn't a step you can perform after or before another one. Testing is a stage that is woven in the remaining stages. You must take testing into account since you analyse your ilities until you deploy your system.

Step 1-4.1: functional testing

A part of testing is devoted to finding bugs, that is, errors in the code of our system. This is usually referred to as functional testing.

Step 1-4.2: performance testing

UNIVERSIDAD Ð SEVILLA

104

Another important aspect of testing is performance testing, aka performance metering. The goal is to find out how many concurrent users a system may serve. Increasing the number of concurrent users will require setting up new machines that will work in parallel. It might be the case that the performance isn't acceptable; in such cases, it'll be necessary to return to the first stage and revisit our design and our implementation.

Step 1-4.3: acceptance testing

UNIVERSIDAD Ð SEVILLA

105

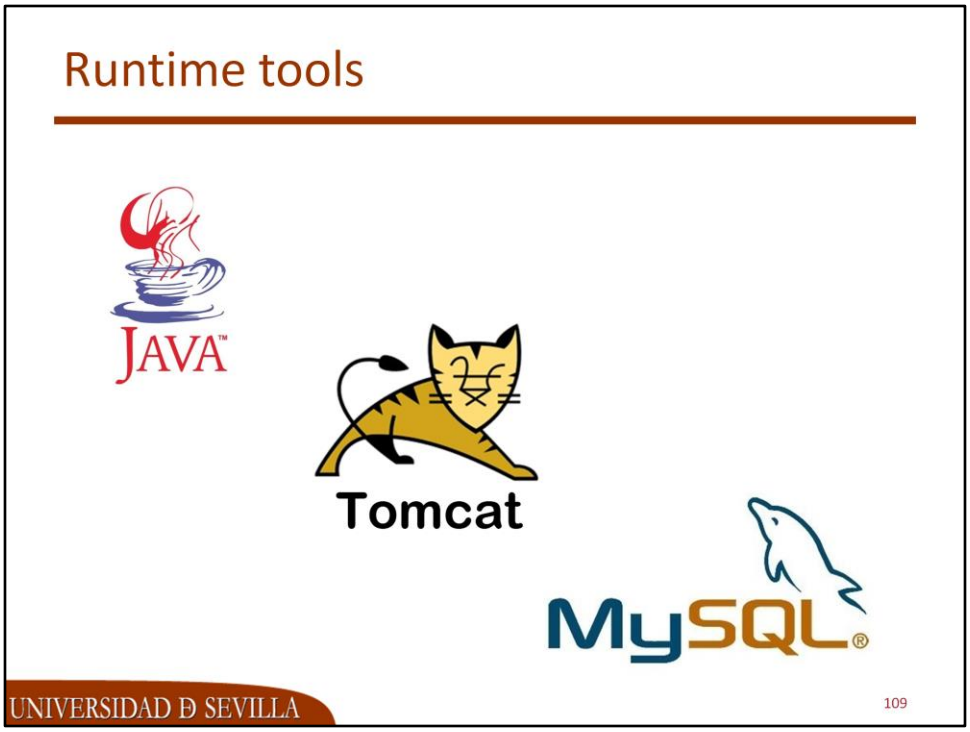
Another important part of testing is making sure that the system we're building meets our customer's expectations. That's usually referred to as acceptance testing and it is commonly carried out by means of check lists.

In the previous slides, we've mentioned some of the tools and technologies that we're going to use in this subject. In the following few slides, we'll provide an overall picture.

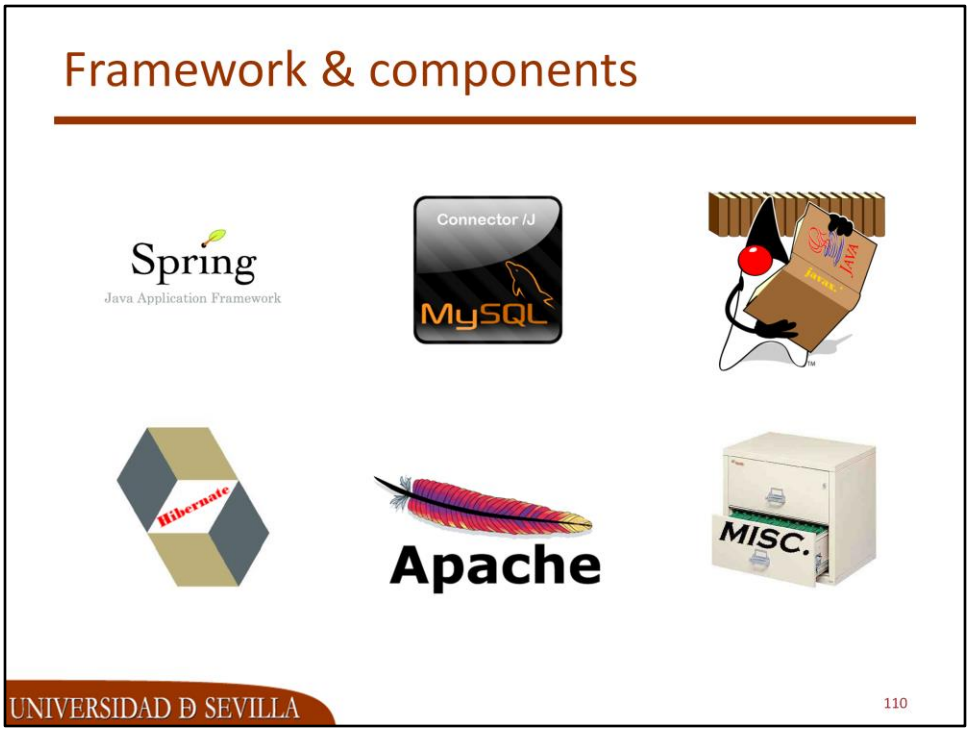
The first tool we'd like to tell you about is ProjETSII. You must get familiar with it because you're required to manage your projects in this subject using it.

We strongly recommend to use Eclipse Indigo EE SR2 as a development environment, Astah to create your UML diagrams, Apache Maven to manage your projects, and Balsamiq or Evolus Pencil create your mock-ups.

NOTE: Balsamiq is a cloud-based tool that requires a license. If you're interested in using it, please ask Prof. Corchuelo for a license by the end of the first lesson. No licenses can be awarded after that deadline.



We recommend to use Java 1.7 as your runtime engine, since it's quite common in the industry. To deploy your applications, we recommend that you should use Tomcat and MySQL. They both are also very common in the industry.



Regarding the framework, we recommend Spring plus a custom project template. It's quite a well-known framework in the industry and it helps instantiate an architecture that is very appropriate to devise web information systems. Furthermore, we'll use a variety of components: MySQL connectors, Javax components, Hibernate components, Apache components, and many others. We'll go through them in the forthcoming lectures.

By now, you should know a lot regarding this subject: we've gone through the faculty, the co-ordinates, the syllabus, and our local view of Software Engineering. It's time to conclude this lecture.

What's the role of D&T?

UNIVERSIDAD Ð SEVILLA

112

The colophon's a short, but very important section. We'd just like to reflect on the role of D&T.

Remember this?

Learn how to produce a typical small- to mid-scale web information system using industrial-strength components, tools, and methods.

Can you remember this? Our goal's to learn how to produce a typical small- to mid-scale web information system using industrial-strength components, tools, and methods. That's all, pretty simple.

We're the starting point

UNIVERSIDAD Ð SEVILLA

114

D&T will help you be ready for your first day as a Software Engineer. Nothing else.

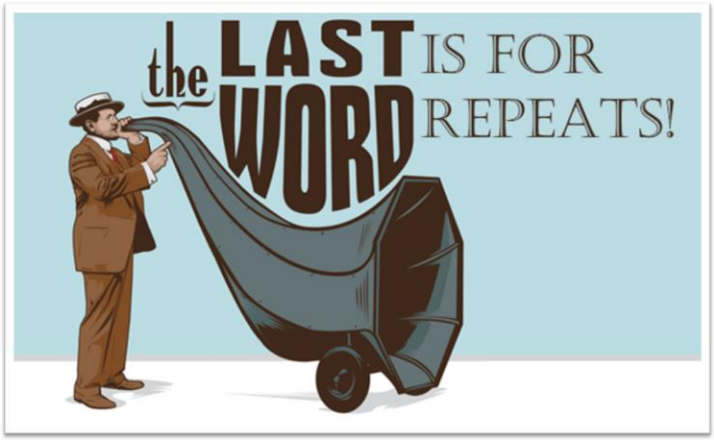
And then comes a long road ahead

UNIVERSIDAD Ð SEVILLA

115

And there's a long road ahead to become a professional. You'll walk that road in an industrial context. D&T shall not help you walk that road, it'll help you to get ready to start walking, nothing else.

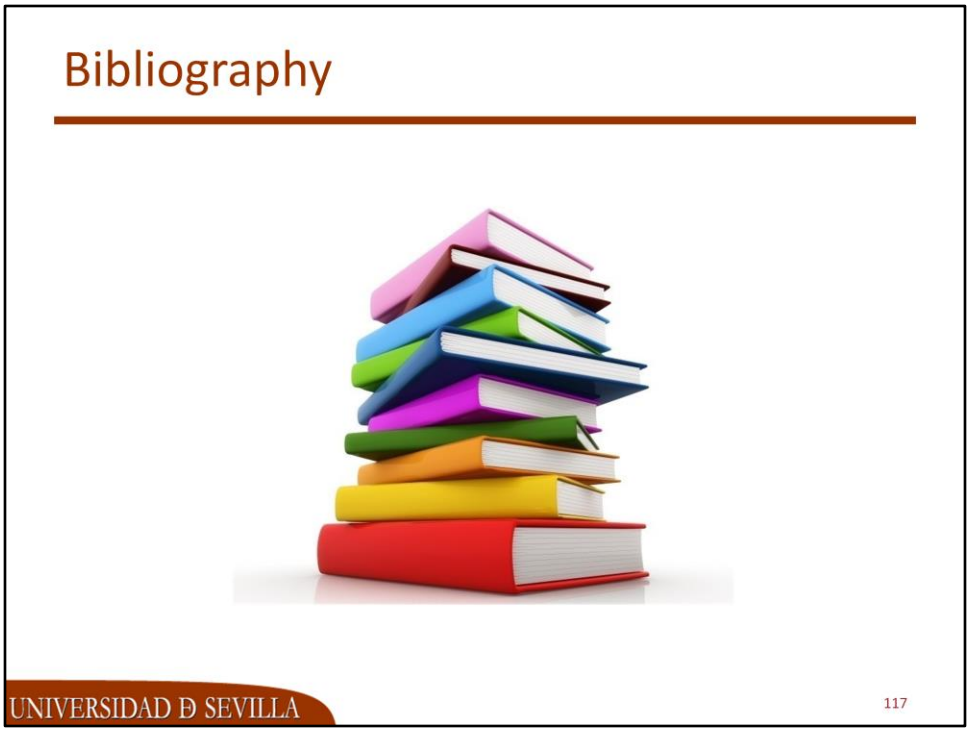
Any repeats in the room?



UNIVERSIDAD Ð SEVILLA

116

By the way, are there any repeats in the room? Please, talk a little about your experience. You must have learnt something regarding this subject and how to face it; please, share your impressions with the other students.



Should you need more information on this lecture, please, take a look at the documents that are available at the USE's e-learning platform and the following bibliography:

Web information systems
David Taniar, Johanna W. Rahayu
Idea Group Publishing, 2004

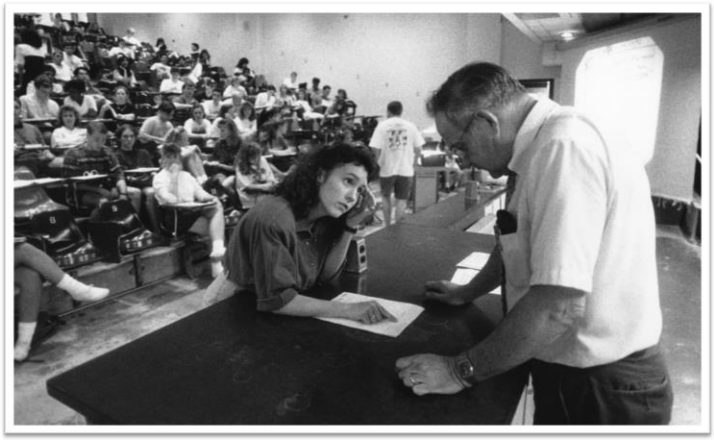
Enterprise software architecture and design
Dominic Duggan
Wiley, 2012

This bibliography is available in electronic format for our students at the USE's virtual library. If you don't know how to have access to the USE's virtual library, please, ask our librarians for help.

Time for questions, please

Time for questions, please. Note that we won't update the slides with the questions that are posed in the lectures. Please, attend them and take notes.

Please, ask now, not later!!



UNIVERSIDAD Ð SEVILLA

119

Please, ask your questions during the lecture. Don't wait till the lecture is finished!!

Thanks for attending this lecture! See you soon.