41951- ANÁLISE DE SISTEMAS

AS: course presentation

Ilídio Oliveira | v2022/03/08

universidade de aveiro
departamento de eletrónica,
telecomunicações e informática
deti

Key resources

Web page at Moodle

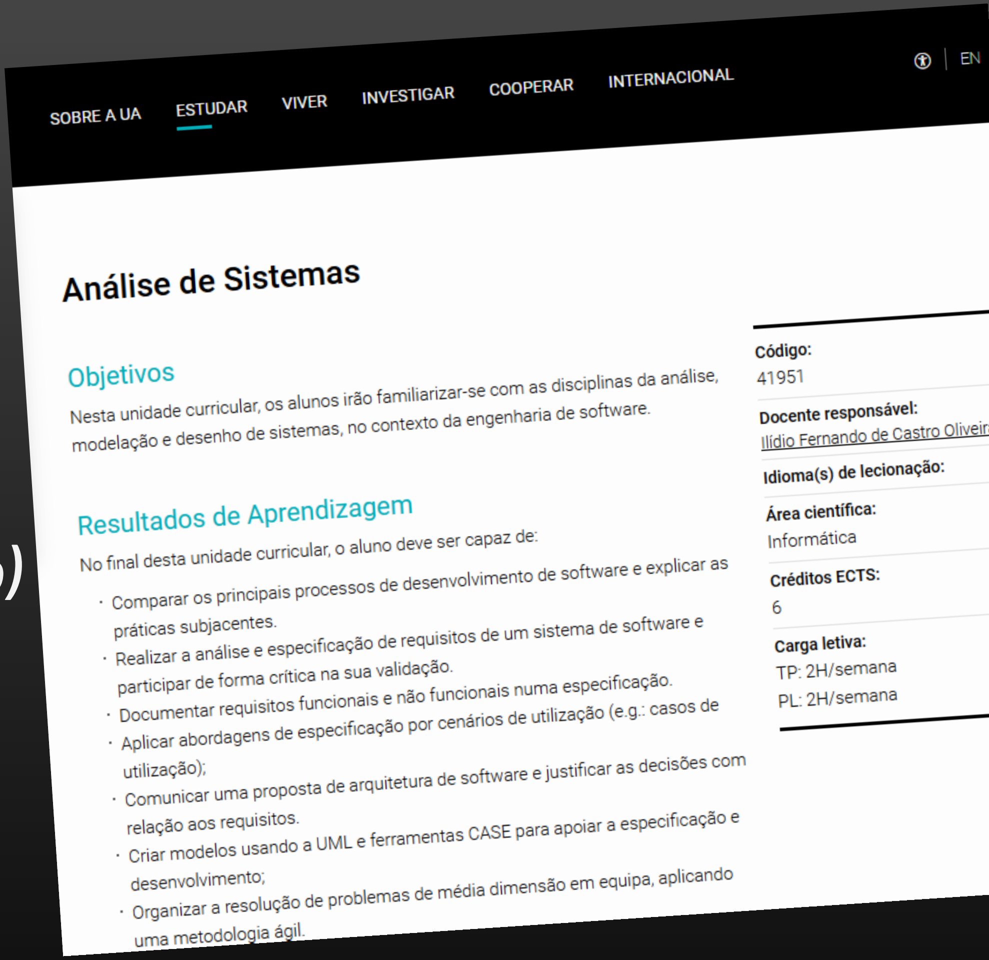
All learning materials
Assignments submission

Syllabus (*dossier pedagógico*)

Subjects covered
Grading (and other) rules

Course Calendar

Weekly plan



Análise de Sistemas

Objetivos

Nesta unidade curricular, os alunos irão familiarizar-se com as disciplinas da análise, modelação e desenho de sistemas, no contexto da engenharia de software.

Resultados de Aprendizagem

No final desta unidade curricular, o aluno deve ser capaz de:

- Comparar os principais processos de desenvolvimento de software e explicar as práticas subjacentes.
- Realizar a análise e especificação de requisitos de um sistema de software e participar de forma crítica na sua validação.
- Documentar requisitos funcionais e não funcionais numa especificação.
- Aplicar abordagens de especificação por cenários de utilização (e.g.: casos de utilização);
- Comunicar uma proposta de arquitetura de software e justificar as decisões com relação aos requisitos.
- Criar modelos usando a UML e ferramentas CASE para apoiar a especificação e desenvolvimento;
- Organizar a resolução de problemas de média dimensão em equipa, aplicando uma metodologia ágil.

Código:
41951

Docente responsável:
Ilídio Fernando de Castro Oliveir

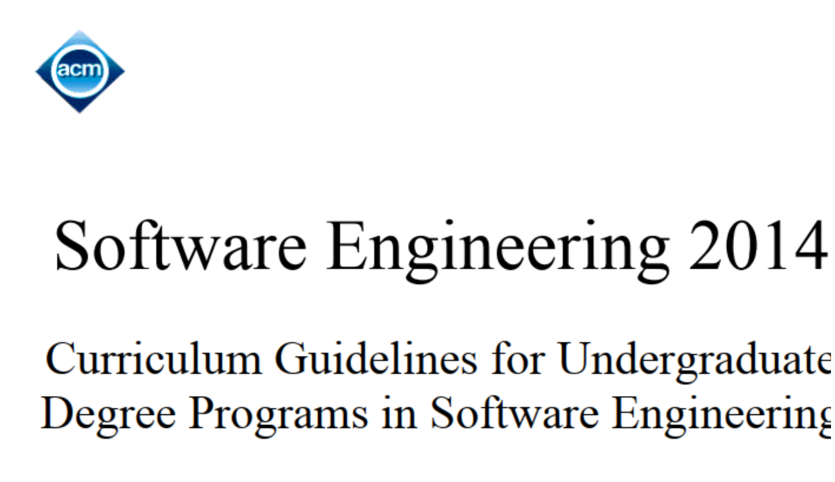
Idioma(s) de lecionação:

Área científica:
Informática

Créditos ECTS:
6

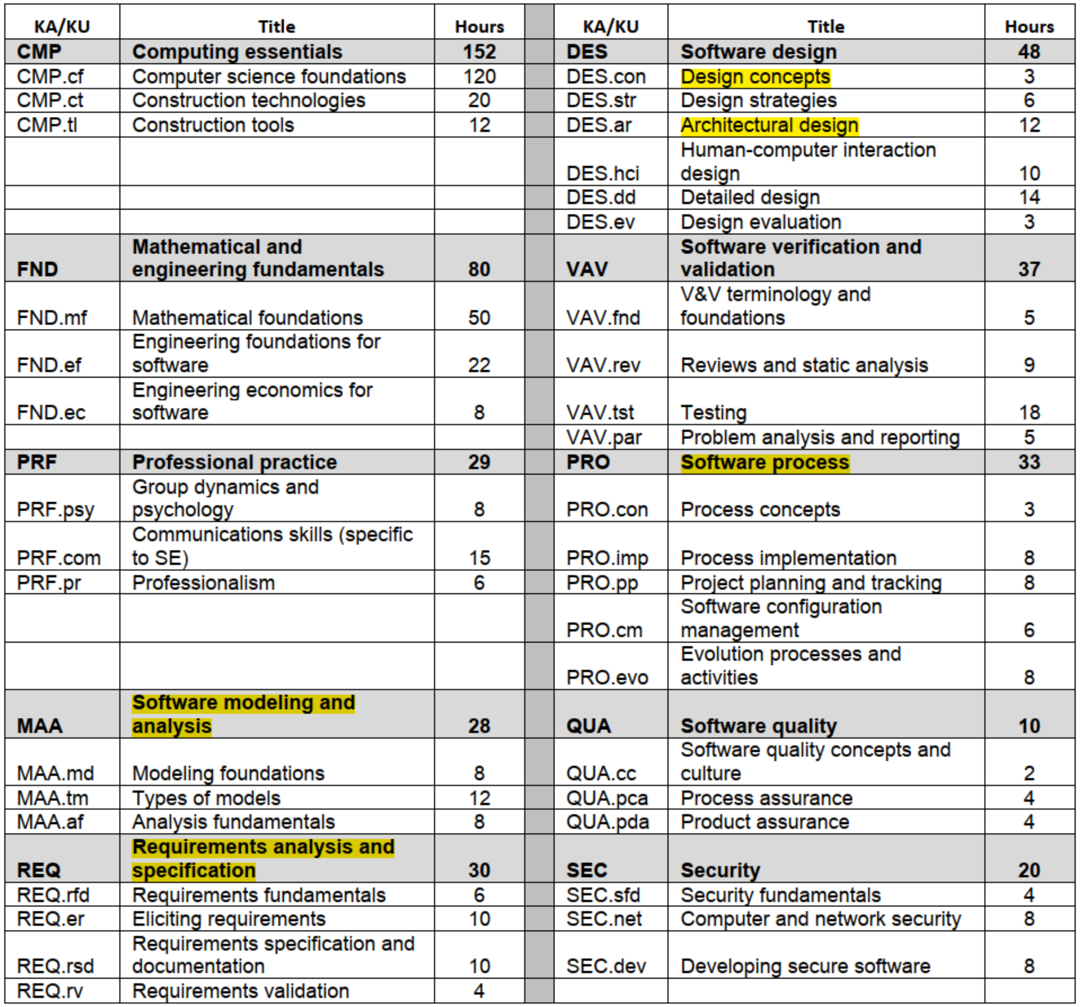
Carga letiva:
TP: 2H/semana
PL: 2H/semana

Mapping AS in the ACM/IEEE curriculum guidelines



Software Engineering 2014

Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering



KA/KU	Title	Hours	KA/KU	Title	Hours
CMP	**Computing essentials**	**152**	**DES**	**Software design**	**48**
CMP.cf	Computer science foundations	120	DES.con	Design concepts	3
CMP.ct	Construction technologies	20	DES.str	Design strategies	6
CMP.tl	Construction tools	12	DES.ar	Architectural design	12
			DES.hci	Human-computer interaction design	10
			DES.dd	Detailed design	14
			DES.ev	Design evaluation	3
FND	**Mathematical and engineering fundamentals**	**80**	**VAV**	**Software verification and validation**	**37**
FND.mf	Mathematical foundations	50	VAV.fnd	V&V terminology and foundations	5
FND.ef	Engineering foundations for software	22	VAV.rev	Reviews and static analysis	9
FND.ec	Engineering economics for software	8	VAV.tst	Testing	18
			VAV.par	Problem analysis and reporting	5
PRF	**Professional practice**	**29**	**PRO**	**Software process**	**33**
PRF.psy	Group dynamics and psychology	8	PRO.con	Process concepts	3
PRF.com	Communications skills (specific to SE)	15	PRO.imp	Process implementation	8
PRF.pr	Professionalism	6	PRO.pp	Project planning and tracking	8
			PRO.cm	Software configuration management	6
			PRO.evo	Evolution processes and activities	8
MAA	**Software modeling and analysis**	**28**	**QUA**	**Software quality**	**10**
MAA.md	Modeling foundations	8	QUA.cc	Software quality concepts and culture	2
MAA.tm	Types of models	12	QUA.pca	Process assurance	4
MAA.af	Analysis fundamentals	8	QUA.pda	Product assurance	4
REQ	**Requirements analysis and specification**	**30**	**SEC**	**Security**	**20**
REQ.rfd	Requirements fundamentals	6	SEC.sfd	Security fundamentals	4
REQ.er	Eliciting requirements	10	SEC.net	Computer and network security	8
REQ.rsd	Requirements specification and documentation	10	SEC.dev	Developing secure software	8
REQ.rv	Requirements validation	4			

I. Oliveira

Course subject: analysis and specification of software systems

Systems analysis

Disciplines related to the characterization of the problem and specification of the technical solution

Development Process

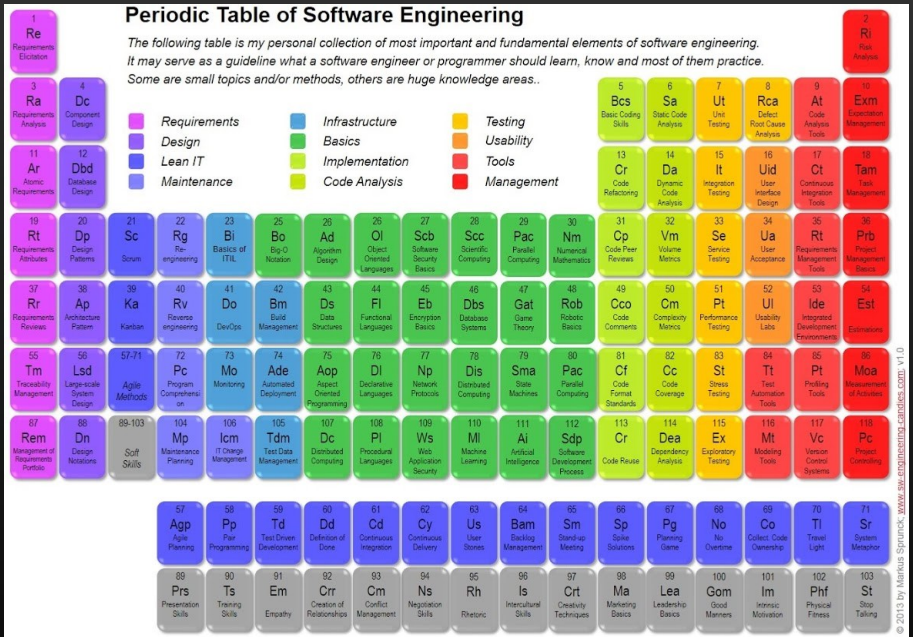
Systematic engineering method. Defines activities, roles and outcomes

Visual modeling

Unified Modeling Language - UML

CASE tools (computer-aided software engineering)

E.g .: VisualParadigm



Periodic Table of Software Engineering

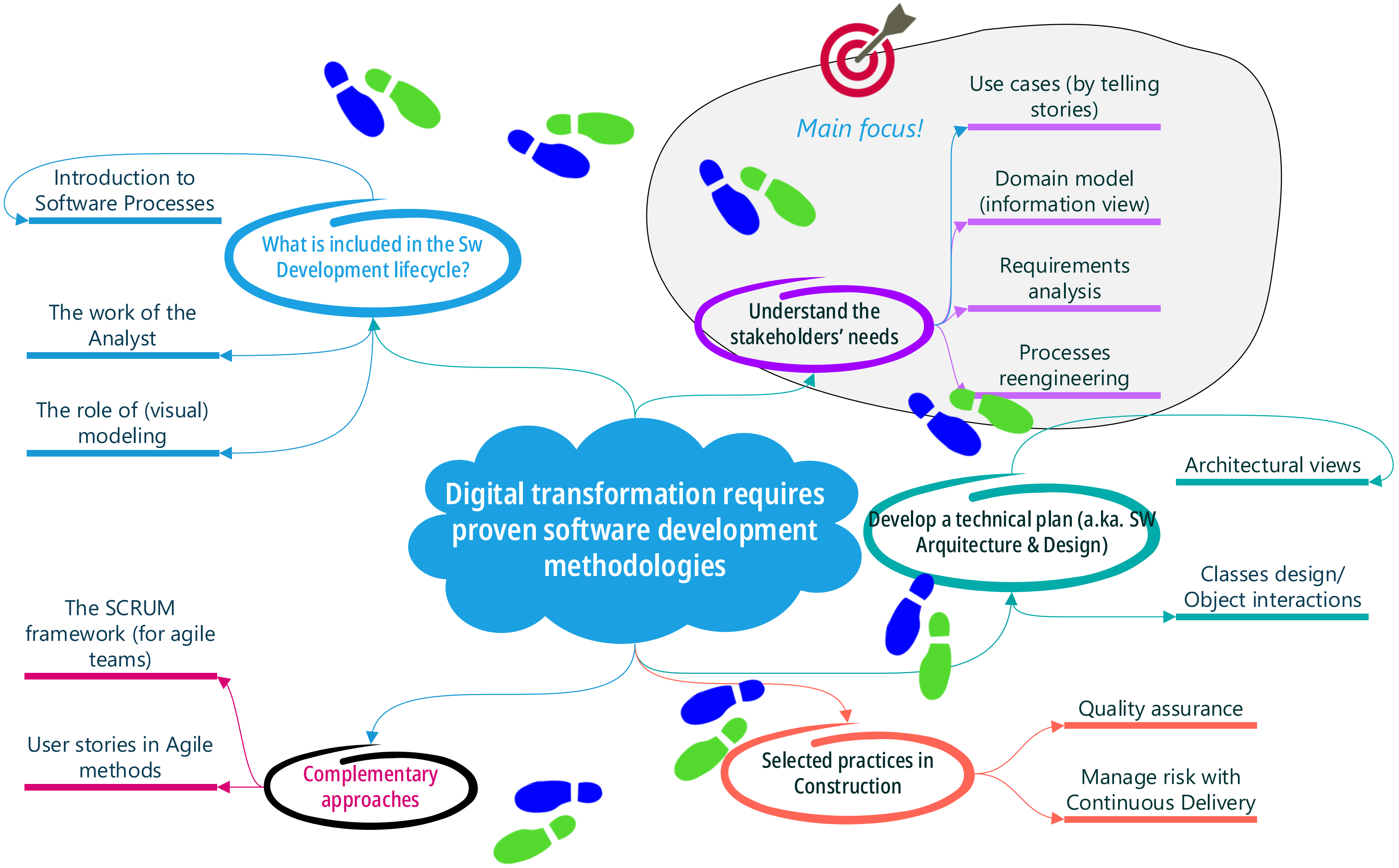
The following table is my personal collection of most important and fundamental elements of software engineering.
It may serve as a guideline what a software engineer or programmer should learn, know and most of them practice.
Some are small topics and/or methods, others are huge knowledge areas..

Requirements	*Infrastructure*	*Testing*
Design	*Basics*	*Usability*
Lean IT	*Implementation*	*Tools*
Maintenance	*Code Analysis*	*Management*

1 Re Requirements Elicitation																	2 Ri Risk Analysis
3 Ra Requirements Analysis	4 Dc Component Design											5 Bcs Basic Coding Skills	6 Sa Static Code Analysis	7 Ut Unit Testing	8 Rca Defect Root Cause Analysis	9 At Code Analysis Tools	10 Exm Expectation Management
11 Ar Atomic Requirements	12 Dbd Database Design											13 Cr Code Refactoring	14 Da Dynamic Code Analysis	15 It Integration Testing	16 Uid User Interface Design	17 Ct Continuous Integration Tools	18 Tam Task Management
19 Rt Requirements Attributes	20 Dp Design Patterns	21 Sc Scrum	22 Rg Re-engineering	23 Bi Basics of ITIL	25 Bo Big-O Notation	26 Ad Algorithm Design	26 Ol Object Oriented Languages	27 Scb Software Security Basics	28 Scc Scientific Computing	29 Pac Parallel Computing	30 Nm Numerical Mathematics	31 Cp Code Peer Reviews	32 Vm Volume Metrics	33 Se Service Testing	34 Ua User Acceptance	35 Rt Requirements Management Tools	36 Prb Project Management Basics
37 Rr Requirements Reviews	38 Ap Architecture Pattern	39 Ka Kanban	40 Rv Reverse engineering	41 Do DevOps	42 Bm Build Management	43 Ds Data Structures	44 Fl Functional Languages	45 Eb Encryption Basics	46 Dbs Database Systems	47 Gat Game Theory	48 Rob Robotic Basics	49 Cco Code Comments	50 Cm Complexity Metrics	51 Pt Performance Testing	52 Ul Usability Labs	53 Ide Integrated Development Environments	54 Est Estimations
55 Tm Traceability Management	56 Lsd Large-scale System Design	57-71 *Agile Methods*	72 Pc Program Comprehension	73 Mo Monitoring	74 Ade Automated Deployment	75 Aop Aspect Oriented Programming	76 Dl Declarative Languages	77 Np Network Protocols	78 Dis Distributed Computing	79 Sma State Machines	80 Pac Parallel Computing	81 Cf Code Format Standards	82 Cc Code Coverage	83 St Stress Testing	84 Tt Test Automation Tools	85 Pt Profiling Tools	86 Moa Measurement of Activities
87 Rem Management of Requirements Portfolio	88 Dn Design Notations	89-103 *Soft Skills*	104 Mp Maintenance Planning	106 Icm IT Change Management	105 Tdm Test Data Management	107 Dc Distributed Computing	108 Pl Procedural Languages	109 Ws Web Application Security	110 Ml Machine Learning	111 Ai Artificial Intelligence	112 Sdp Software Development Process	113 Cr Code Reuse	114 Dea Dependency Analysis	115 Ex Exploratory Testing	116 Mt Modeling Tools	117 Vc Version Control Systems	118 Pc Project Controlling
			57 Agp Agile Planning	58 Pp Pair Programming	59 Td Test Driven Development	60 Dd Definition of Done	61 Cd Continuous Integration	62 Cy Continuous Delivery	63 Us User Stories	64 Bam Backlog Management	65 Sm Stand-up Meeting	66 Sp Spike Solutions	67 Pg Planning Game	68 No No Overtime	69 Co Collect. Code Ownership	70 Tl Travel Light	71 Sr System Metaphor
			89 Prs Presentation Skills	90 Ts Training Skills	91 Em Empathy	92 Crr Creation of Relationships	93 Cm Conflict Management	94 Ns Negotiation Skills	95 Rh Rhetoric	96 Is Intercultural Skills	97 Crt Creativity Techniques	98 Ma Marketing Basics	99 Lea Leadership Basics	100 Gom Good Manners	101 Im Intrinsic Motivation	102 Phf Physical Fitness	103 St Stop Talking

© 2013 by Markus Sprunck; www.sw-engineering-candies.com; v1.0

http://www.sw-engineering-candies.com/blog-1/periodic-table-of-software-engineering-know-how



Digital transformation requires proven software development methodologies

What is included in the Sw Development lifecycle?

- Introduction to Software Processes
- The work of the Analyst
- The role of (visual) modeling

Main focus!

Understand the stakeholders' needs

- Use cases (by telling stories)
- Domain model (information view)
- Requirements analysis
- Processes reengineering

Develop a technical plan (a.ka. SW Arquitecture & Design)

- Architectural views
- Classes design/ Object interactions

Selected practices in Construction

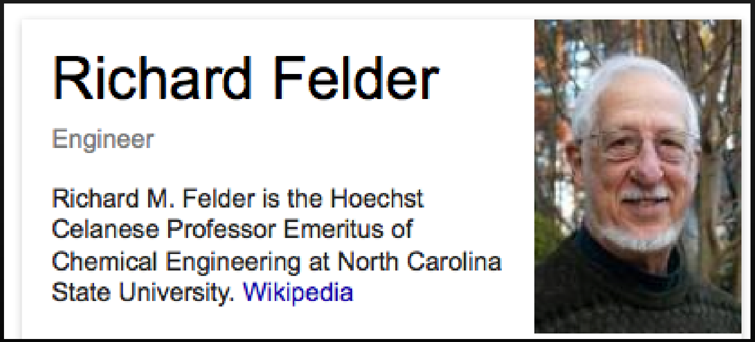
- Quality assurance
- Manage risk with Continuous Delivery

Complementary approaches

- The SCRUM framework (for agile teams)
- User stories in Agile methods

Note on cooperative learning

COOPERATIVE LEARNING It leads to more and deeper learning and longer retention of information; greater development of high-level thinking, problem-solving, communication, and interpersonal skills; more positive attitudes toward engineering and science curricula and careers and greater retention in those curricula; and better preparation for the workplace.

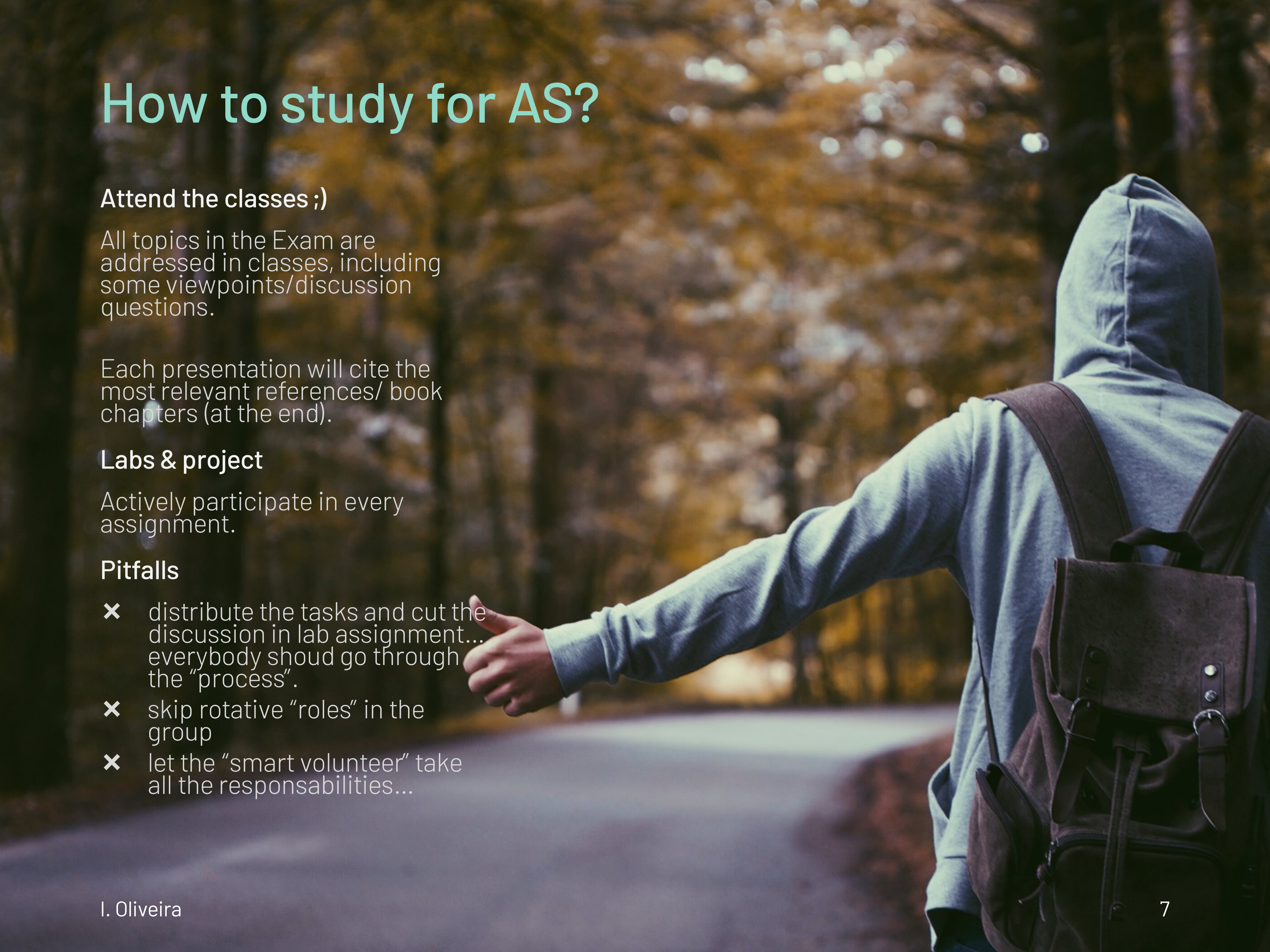


Richard Felder

Engineer

Richard M. Felder is the Hoechst Celanese Professor Emeritus of Chemical Engineering at North Carolina State University. Wikipedia

How to study for AS?



Attend the classes ;)

All topics in the Exam are addressed in classes, including some viewpoints/discussion questions.

Each presentation will cite the most relevant references/ book chapters (at the end).

Labs & project

Actively participate in every assignment.

Pitfalls

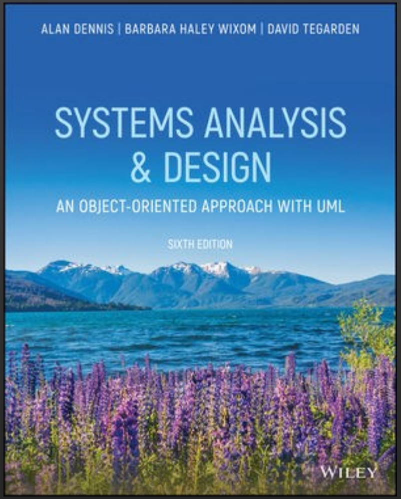
✕ distribute the tasks and cut the discussion in lab assignment... everybody shoud go through the “process”.

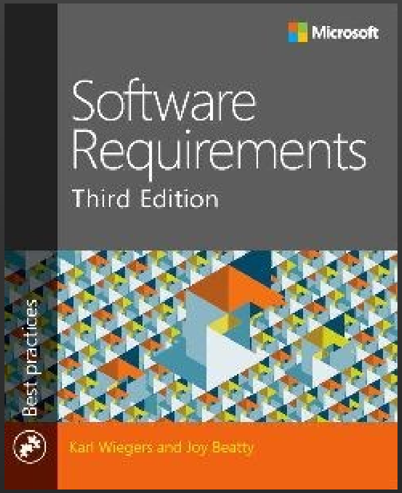
✕ skip rotative “roles” in the group

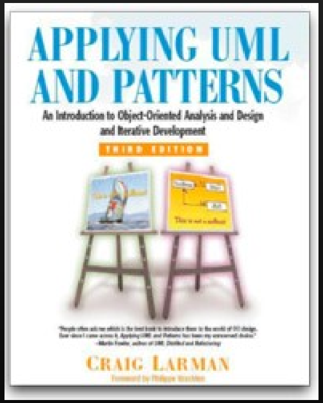
✕ let the “smart volunteer” take all the responsabilities...

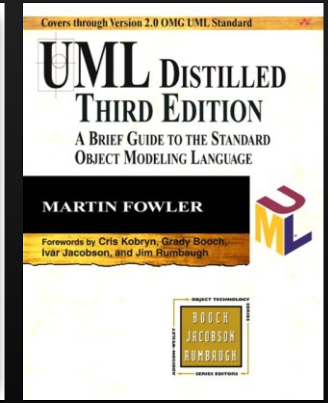
Main references

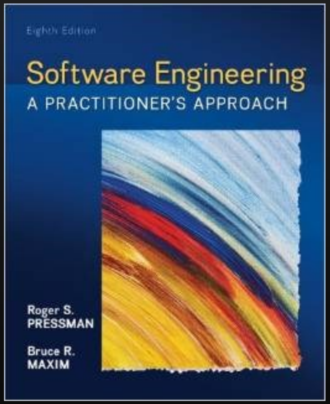
See also selected reference is Moodle

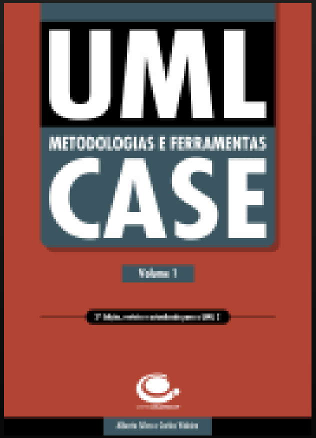

ALAN DENNIS | BARBARA HALEY WIXOM | DAVID TEGARDEN
SYSTEMS ANALYSIS & DESIGN
AN OBJECT-ORIENTED APPROACH WITH UML
SIXTH EDITION
WILEY



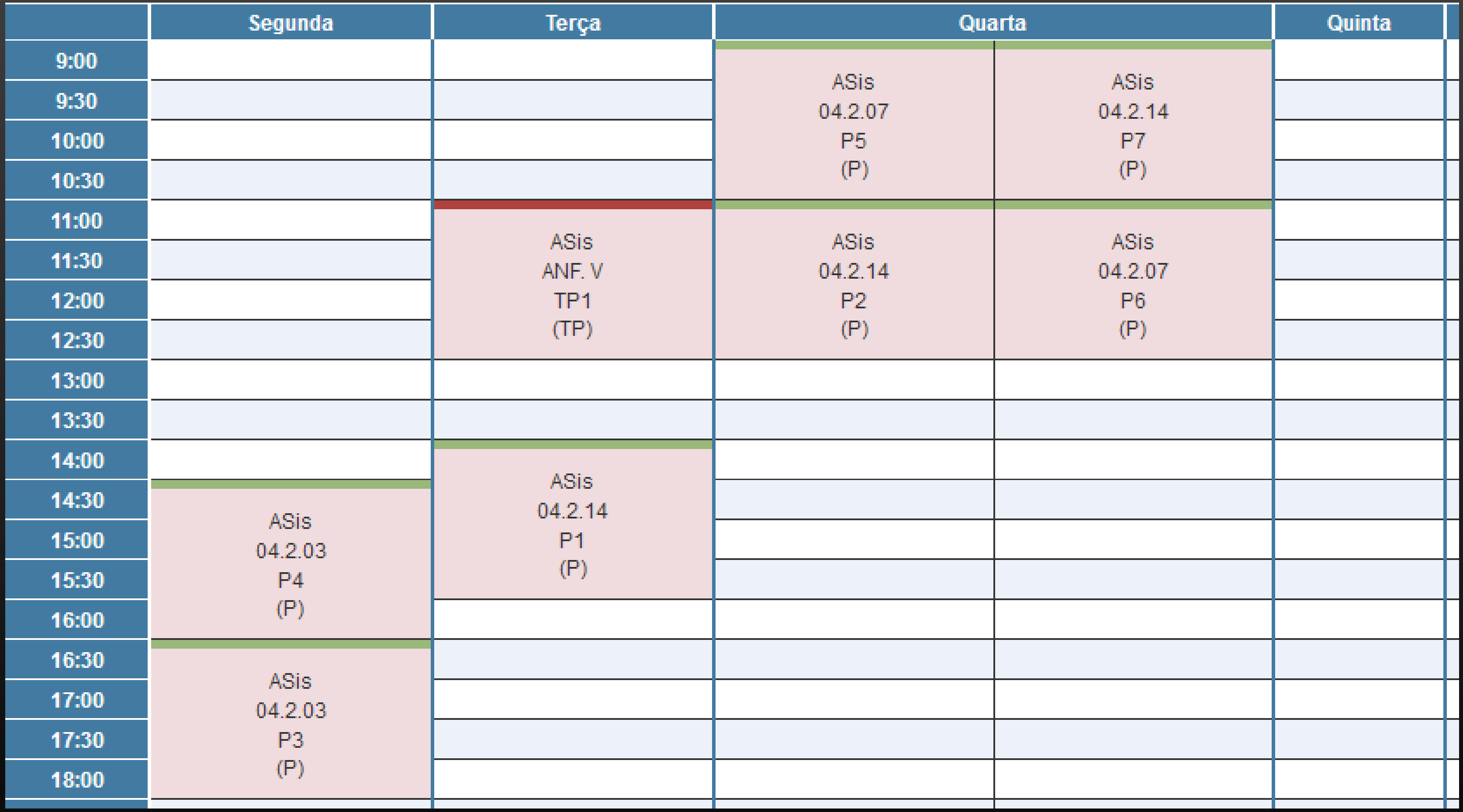

APPLYING UML AND PATTERNS
An Introduction to Object-Oriented Analysis and Design and Iterative Development
THIRD EDITION
CRAIG LARMAN


Covers through Version 2.0 OMG UML Standard
UML DISTILLED
THIRD EDITION
A BRIEF GUIDE TO THE STANDARD OBJECT MODELING LANGUAGE
MARTIN FOWLER
Forewords by Cris Kobryn, Grady Booch, Ivar Jacobson, and Jim Rumbaugh
BOOCH
JACOBSON
RUMBAUGH





Schedule & labs submissions



	Segunda	Terça	Quarta		Quinta
9:00			ASis 04.2.07 P5 (P)	ASis 04.2.14 P7 (P)	
9:30					
10:00					
10:30					
11:00		ASis ANF. V TP1 (TP)	ASis 04.2.14 P2 (P)	ASis 04.2.07 P6 (P)	
11:30					
12:00					
12:30					
13:00					
13:30					
14:00		ASis 04.2.14 P1 (P)			
14:30	ASis 04.2.03 P4 (P)				
15:00					
15:30					
16:00					
16:30	ASis 04.2.03 P3 (P)				
17:00					
17:30					
18:00					