

47006- ANÁLISE E MODELÇÃO DE SISTEMAS

AMS: course presentation

Ilídio Oliveira

v2020/10/06

Key resources

Web page at [Moodle](#)

All learning materials
Assignments submission

[Syllabus](#) (*dossier pedagógico*)

Subjects covered
Grading (and other) rules

Course Calendar

[Weekly plan](#)



Mapping AMS in the ACM/IEEE curriculum guidelines



Software Engineering 2014

Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering

[Resource] Full Guidelines document @eLearning

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			

Course subject: analysis and specification of software-intensive 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.

1 Re
Requirements
Discussion

3 Ra
Requirements
Analysis

4 Dc
Component
Design

11 Ar
Atomic
Requirements

12 Dbd
Database
Design

19 Rt
Requirements
Attributes

20 Dp
Design
Patterns

21 Sc
Scrum

22 Rg
Re-engineering

23 Bi
Basics of
ITIL

24 Bo
Big O
Notation

25 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
Architectural
Patterns

39 Ka
Karlson

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
Robotics
Basics

49 Cc
Code
Comments

50 Cm
Complexity
Metrics

51 Pt
Performance
Testing

52 Ul
Usability
Labs

53 Ide
Integrated
Development
Environments

54 Est
Estimation

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
Polling
Tools

86 Moa
Management
of Success

87 Rem
Management of
Requirements
Portfolio

88 Dn
Design
Decisions

89-103
Soft
Skills

104 Mp
Maintenance
Planning

106 Icm
IT Change
Management

107 Tdm
Test Data
Management

108 Dc
Distributed
Computing

109 Pl
Procedural
Languages

110 Ws
Web
Application
Security

111 Mi
Machine
Learning

112 Ai
Artificial
Intelligence

113 Sdp
Software
Development
Process

114 Or
Code Reuse

115 Dea
Dependency
Analysis

116 Ex
Exploratory
Testing

118 Mt
Modeling
Tools

119 Vc
Version
Control
System

120 Phf
Physical
Fitness

121 Ti
Travel
Light

122 St
Stop
Talking

123 Prs
Presentation
Skills

124 Ts
Training
Skills

125 Em
Empathy

126 Crr
Creation of
Relationships

127 Cm
Conflict
Management

128 Ns
Negotiation
Skills

129 Rh
Rhetoric

130 Us
User Stories

131 Bm
Backlog
Management

132 Sm
Stand-up
Meeting

133 Sp
Spike
Solutions

134 Pg
Playing
Game

135 No
No
Owning

136 Co
Collect Code
Ownership

137 Im
Im
Imitation

138 Phf
Physical
Fitness

139 St
Stop
Talking

2 Ri
Risk
Analysis

5 Bcs
Basic Coding
Skills

6 Sa
Soft Skills
Analysis

7 Ut
Unit
Testing

8 Rca
Root Cause
Analysis

9 At
Atomic
Analysis
Tools

10 Exm
Exercises
Management

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
Team
Management

26 OI
Object
Oriented
Languages

32 Vm
Volume
Metrics

33 Se
Service
Testing

34 Ua
User
Acceptance

35 Rt
Requirements
Management
Tools

36 Prb
Project
Management
Basics

43 Ds
Data
Structures

44 Fl
Functional
Languages

45 Eb
Encryption
Basics

46 Dbs
Database
Systems

47 Gat
Game
Theory

48 Rob
Robotics
Basics

49 Cc
Code
Comments

50 Cm
Complexity
Metrics

51 Pt
Performance
Testing

52 Ul
Usability
Labs

53 Ide
Integrated
Development
Environments

54 Est
Estimation

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
Polling
Tools

86 Moa
Management
of Success

87 Rem
Management of
Requirements
Portfolio

88 Dn
Design
Decisions

89-103
Soft
Skills

104 Mp
Maintenance
Planning

106 Icm
IT Change
Management

107 Tdm
Test Data
Management

108 Dc
Distributed
Computing

109 Pl
Procedural
Languages

110 Ws
Web
Application
Security

111 Mi
Machine
Learning

112 Ai
Artificial
Intelligence

113 Sdp
Software
Development
Process

114 Or
Code Reuse

115 Dea
Dependency
Analysis

116 Ex
Exploratory
Testing

118 Mt
Modeling
Tools

119 Vc
Version
Control
System

120 Phf
Physical
Fitness

121 Ti
Travel
Light

122 St
Stop
Talking

123 Prs
Presentation
Skills

124 Ts
Training
Skills

125 Em
Empathy

126 Crr
Creation of
Relationships

127 Cm
Conflict
Management

128 Ns
Negotiation
Skills

129 Rh
Rhetoric

130 Us
User Stories

131 Bm
Backlog
Management

132 Sm
Stand-up
Meeting

133 Sp
Spike
Solutions

134 Pg
Playing
Game

135 No
No
Owning

136 Co
Collect Code
Ownership

137 Im
Im
Imitation

138 Phf
Physical
Fitness

139 St
Stop
Talking

Infrastructure

Basics

Implementation

Code Analysis

Testing

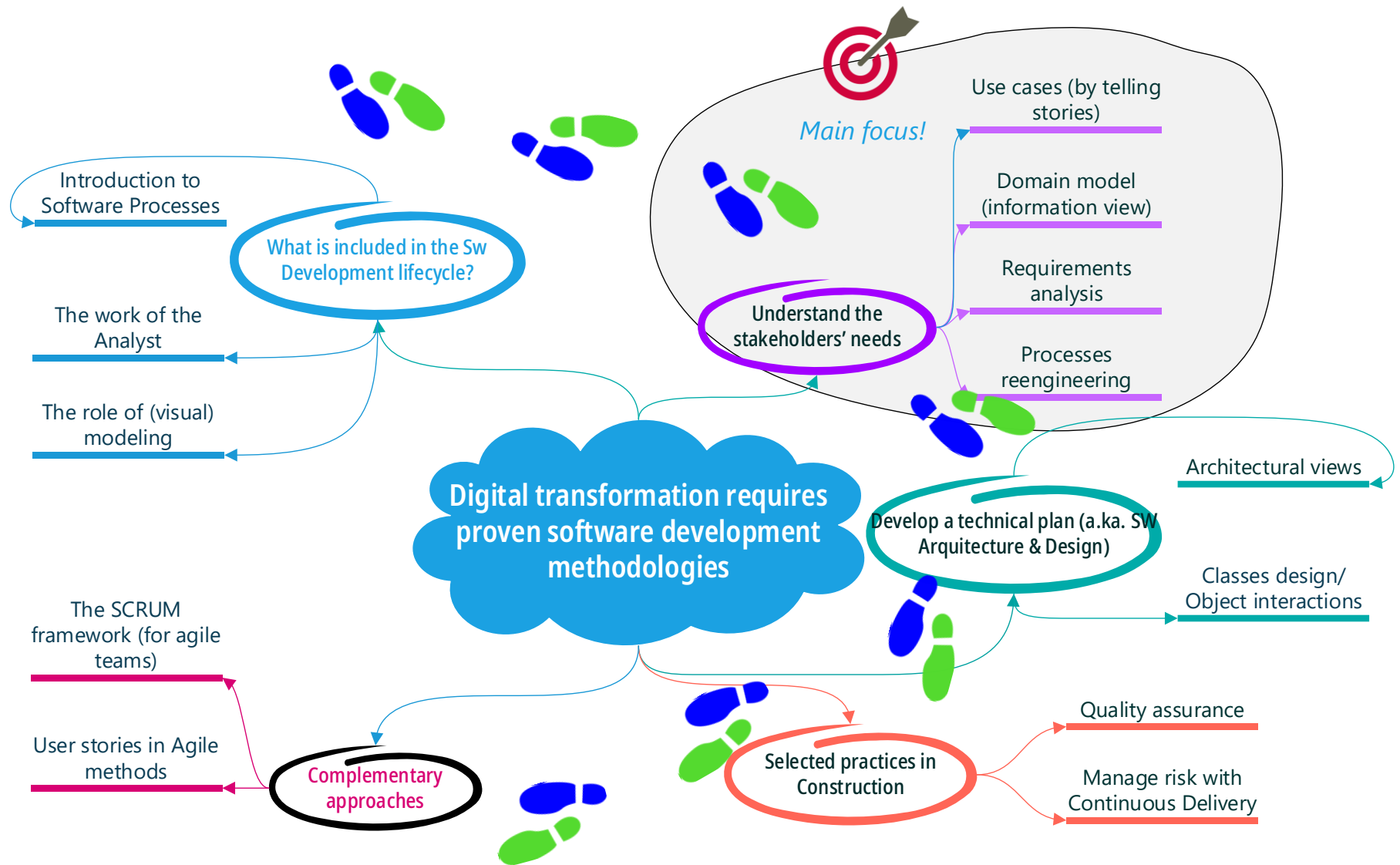
Usability

Tools

Management

© 2013 by Marcus Sprunck, www.me-engineers.com/, v1.0

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



Note on cooperative learning

Team effort

Labs

Project assignment

Individual

Written exam

More on [Cooperative Learning](#).

[Resource] Turning groups into teams @eLearning

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 AMS?

Attend the classes ;)

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

Books

See references cited at the end of each presentation to find the relevant Chapters (from selected references)

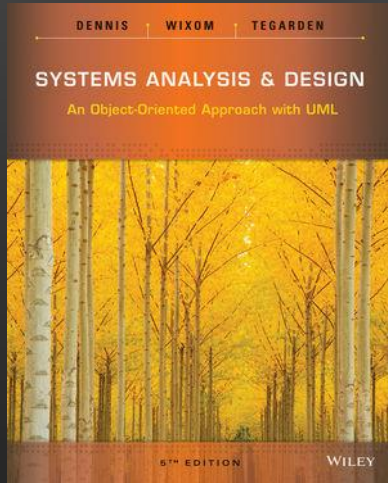
Labs & project

Actively participate in every assignment.

Pitfalls

- ✗ distribute the tasks and cut the discussion in lab assignment... everybody should go through the work.
- ✗ skip rotative "roles" in the group
- ✗ let the "smart volunteer" take all the responsibilities

Main references



Adopted text book.

