Introduction to CITS4401

Software Requirements and Design CITS4401

Lecture 1-Part1

Outline of this Lecture

What you'll be learning in CITS4401

- Software Engineering
- Requirements
- Design
- Software Methodologies
- Software Processes

How you'll be learning in CITS4401

Introductions

- **Unit Coordinator**:

  Dr Tingting Bi Tingting.Bi@uwa.edu.au

- **Teaching Team**:

  Jiachuan Liu <jiachuan.liu@uwa.edu.au>

  Jichunyang Li <jichunyang.li@uwa.edu.au>

  Aline De Souza Andrade <aline.desouzaandrade@uwa.edu.au>

CITS4401 consultation: Mondays 1:15 – 2 pm (before the lecture), starts week 2

Location: CSSE-G.14

Please use Teams to ask questions.

What you'll be learning in CITS4401

CITS4401 unit overview

Requirements and design are important phases of software development because errors or misunderstandings of software requirements or designs are expensive to correct during later stages and may lead to project failure.

This unit introduces the theory and practice of software requirements and design.

The content comprises

(1) requirements engineering

(2) software design

(3) software architectures; and

(4) design patterns.

CITS4401 Schedule (see LMS)

Lecture time: Mondays 15:00 - 17:00

Lecture venue: PHYS: [243] Clews Lecture Theatre

Week #	Week starting	Reading and Videos	Topic and Lectures Notes (PDF)	Assessments	Workshops Materials	Comments
1	Monday 26 February		Intro to Requirements			
2	Monday 4 March		Requirement Elicitation			In Week 2, no Monday lecture session due to Labour Day.
3	Monday 11 March		Writing Use cases			
4	Monday 18 March		Writing and Verifying Requirements			
5	Monday 25 March		UML Class Diagrams	Take home test 1		In Week 5, no Friday workshop sessions due to Good Friday.
	Monday 1 April	STUDY BREAK				
6	Monday 8 April		UML Dynamic Models	Project Part 1 - Phase 1		
7	Monday 15 April		Intro to System Design	Project Part 1 - Phase 2		
8	Monday 22 April		Software Architecture			In Week 8, no Thursday workshop sessions due to ANZAC Day.
9	Monday 29 April		Software Interfaces	Take home test 2		
10	Monday 6 May		Design Patterns			
11	Monday 13 May		Non-OO Design	Project Part 2		
12	Monday 20 May		Review			
	Monday 27 May	STUDY BREAK				
	Saturday 1 June to Saturday 15 June	JUNE Exam Period				

Software Engineering

- Two words: *Software* and *Engineering*
  - **Software** is a set of instructions, data or programs used to operate computers and execute specific tasks
  - **Engineering** applies systematic, scientific and well-defined processes to produce a good quality product.

- SE is a *creative* process in which
  - there are few *right/wrong* answers
  - but nonetheless some requirements and designs are (much) better than others.
- **Choices must be evaluated and justified.**

Software Engineering Definition

software engineering 3.3810 [ISO standard]

1. systematic application of scientific and technological knowledge, methods, and experience to the design, implementation, testing, and documentation of software

ISO/IEC 2382:2015, Information technology — Vocabulary

2. application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software

ISO/IEC TR 19759:2016, Software Engineering — Guide to the Software Engineering Body of Knowledge (SWEBOK)

Requirements

software requirement 3.3847

1. **software capability needed** by a user to solve a problem or to achieve an objective

2. software capability **that must be met or possessed** by a system or system component to satisfy a contract, standard, specification, or other formally imposed document

software requirements analysis 3.3848

1. **process** of studying user needs to arrive at a definition of system, hardware, or software requirements

Design

design 3.1125

1. [**process**] to define the **architecture, system elements, interfaces**, and other characteristics of a system or system element

2. **result** of the process in (1)

software requirements analysis 3.3848

1. **process** of studying user needs to arrive at a definition of system, hardware, or software requirements

Software Methodologies

methodology 3.2438

1. a system of practices, techniques, procedures, and rules used by those who work in a discipline

[A Guide to the Project Management Body of Knowledge (PMBOK® Guide) — Fifth Edition]

2. specification of the process to follow together with the work products to be used and generated, plus the consideration of the people and tools involved, during an information-based domain development effort

[ISO/IEC 24744:2014 Software Engineering — Metamodel for development methodologies, 3.2]

A series of related methods (ie systematic procedures) or techniques

[Websters Dictionary quoted by Cockburn]

Software Methodologies

"All organisations have a methodology – it is simply how they do business."

"Your methodology is everything you regularly do to get your software out … the conventions your group agrees to."

[Cockburn: Agile SW Development]

Methodology Concepts [Cockburn]

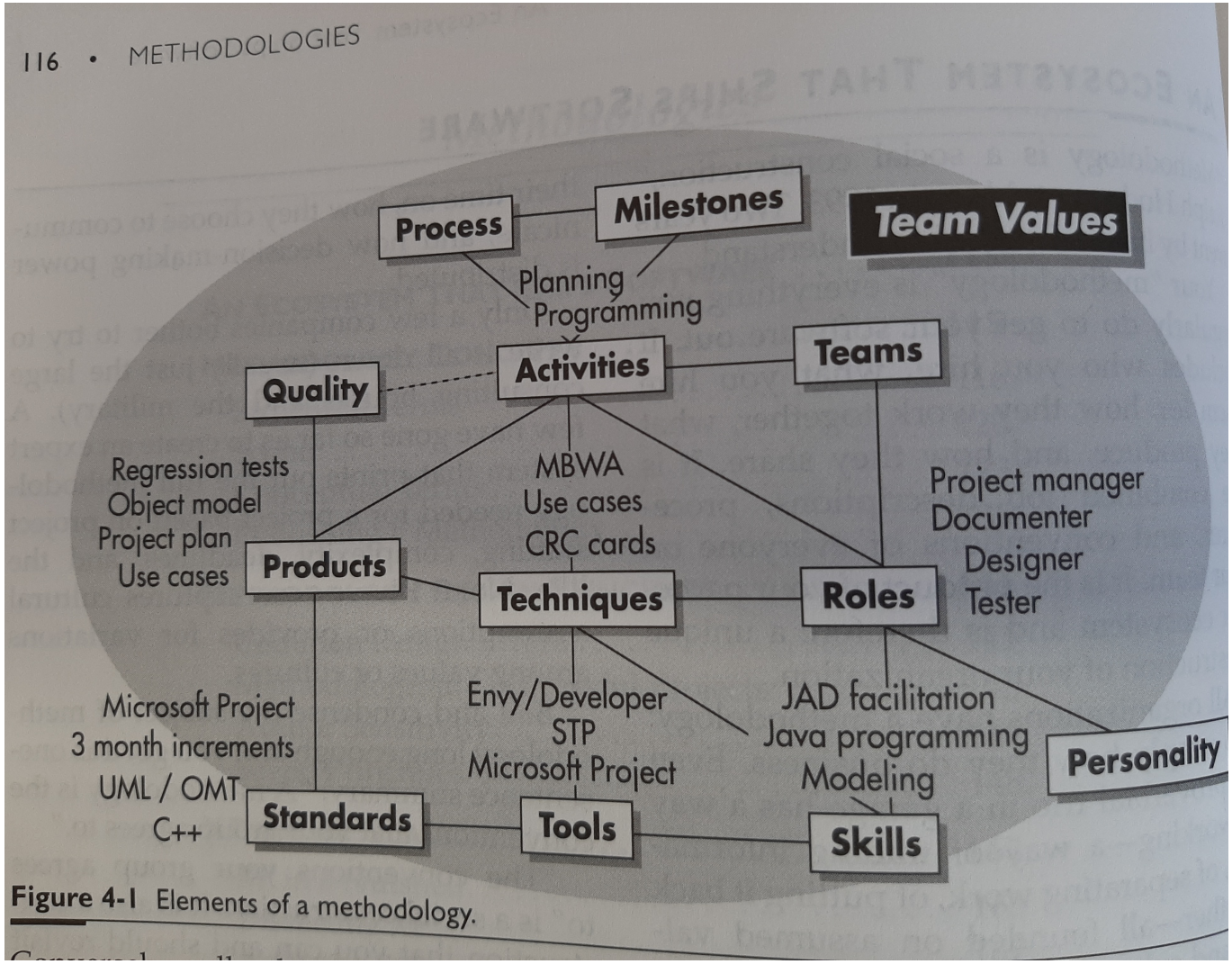


Figure 4-1 Elements of a methodology.

Software Processes

- A **software process** is a set of interrelated activities and tasks that **transform input work products** into **output work products**.
  [SWE page 8-2]

- **Activities** are how the people spend their days.
  e.g. planning, programming, testing, meeting. *[Cockburn]*

- A **process** is how activities fit together time, often with pre- and post-conditions for the activities.
  e.g. design review happens 2 days after designs are sent to participants and produces a list of recommendations
  [Cockburn]

Why bother with requirements and design?

Why Bother….because:

a. ***EVERY*** project has ***uncertainty***….Requirements are one way of dealing with it!

b. Engineers need ***something concrete to work toward*** (many engineers fear uncertainty).

c. The requirement that rules them all (and is implied by the Trade Practices Act). “Is the solution fit-for-purpose?”

d. A completed test program that ***demonstrates requirements are met*** is an effective tool to ***help get paid***!

How you'll be learning in CITS4401/3301

CITS4401 Learning Outcomes

1. **Classify** types of software requirements and designs
2. **Apply** requirements and design processes appropriate for a given scenario
3. **Assess** quality attributes of given requirements and designs
4. **Utilise** design patterns and idioms
5. **Document** software design rationale using discourse conventions of the discipline
6. **Select** a software architecture appropriate for a particular context

CITS4401 Lectures Mon@2pm

- Lectures will present an *overview* of *problems, theory,* and *techniques* for selected topics in SE, with a specific focus on requirements and design
- Lectures will NOT be live-streamed (ms-teams)
- Recorded lectures will be available (Echo via LMS)
- Some pre-recorded mini-lectures will be also provided in LMS
- Text books
  Pressman, Software Engineering
  Fowler, UML distilled
- The texts and other recommended reading is available from **UWA unit readings** via LMS. Online versions are available so please use them.

CITS4401 Workshops

- Workshop classes are practical sessions for students applying SE requirements and design techniques
- Workshops start from Week 1 (five workshops)
- Classwork usually in groups
- Guest presenters from industry

- Two take home tests in week 5 and 9 contribute 10% to your final mark

CITS4401 Assessment

Take home tests (10%)

- Two **individual** take home test based on lectures and workshops
- Due weeks 5 and 9
- Submit on LMS

Requirements and Design Group Project (30%)

- Group project with 2 deliverables
- Due weeks 7 and 11
- Teams of 5 students assigned by the unit coordinator
- Submit on LMS

Final home exam (60%)

- **Individual** closed book exam based on lectures and workshops
- During the standard Exam week at UWA

The small print: See the unit outline for academic conduct rules, late penalties, covid contingencies etc

Summary of this Lecture

What you'll be learning in CITS4401

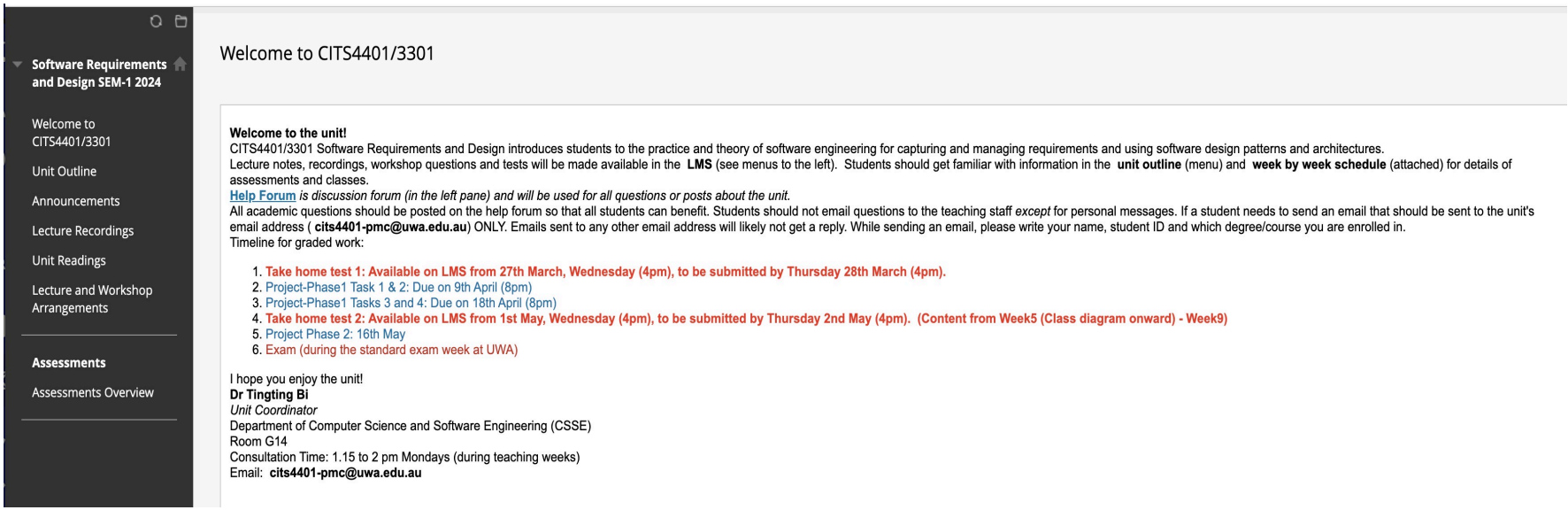
- Software Engineering
- Requirements
- Design
- Software Methodologies
- Software Processes

How you'll be learning in CITS4401

Learning Resources
Get to know these now!

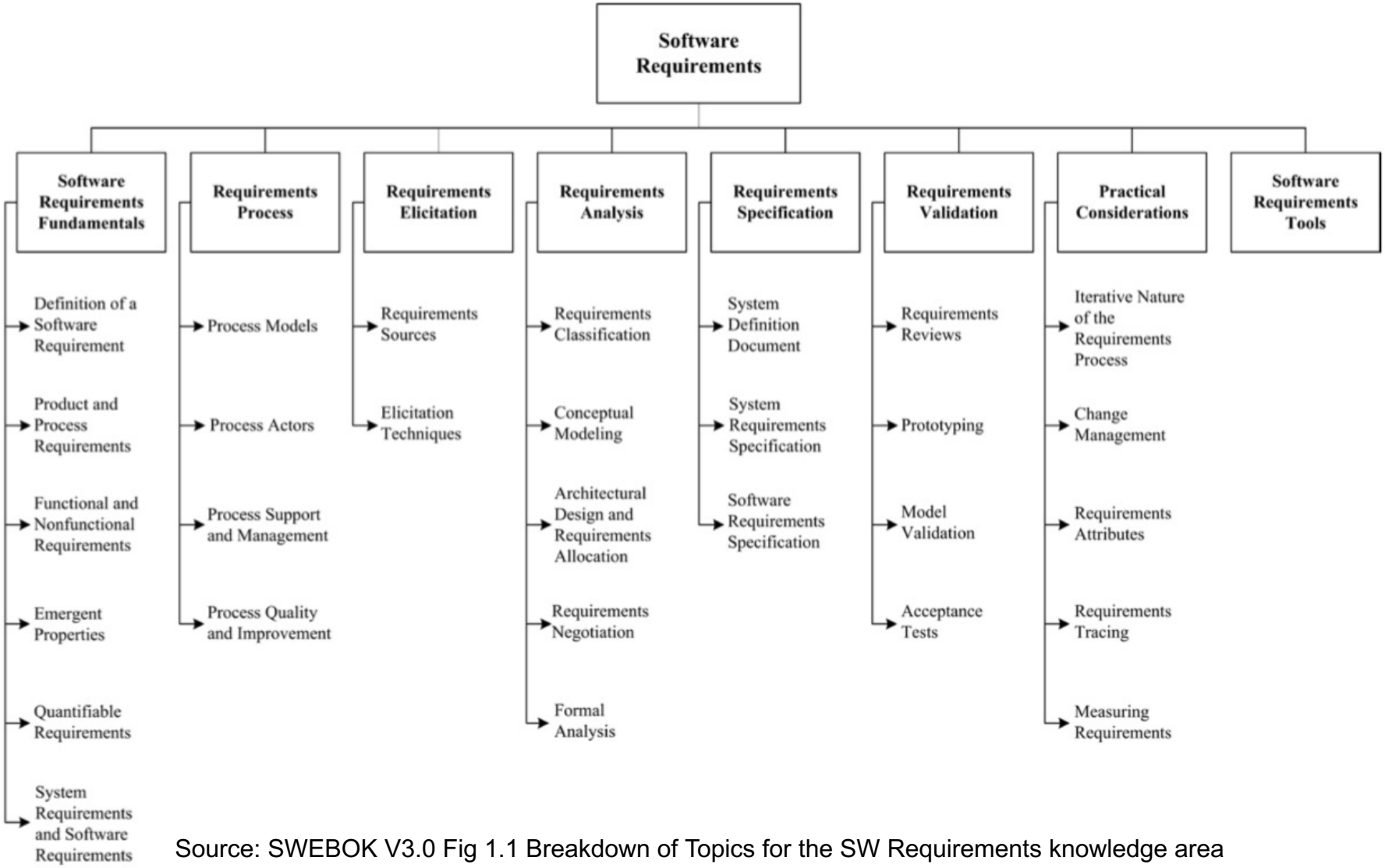
Learning resources 1: LMS

https://lms.uwa.edu.au/



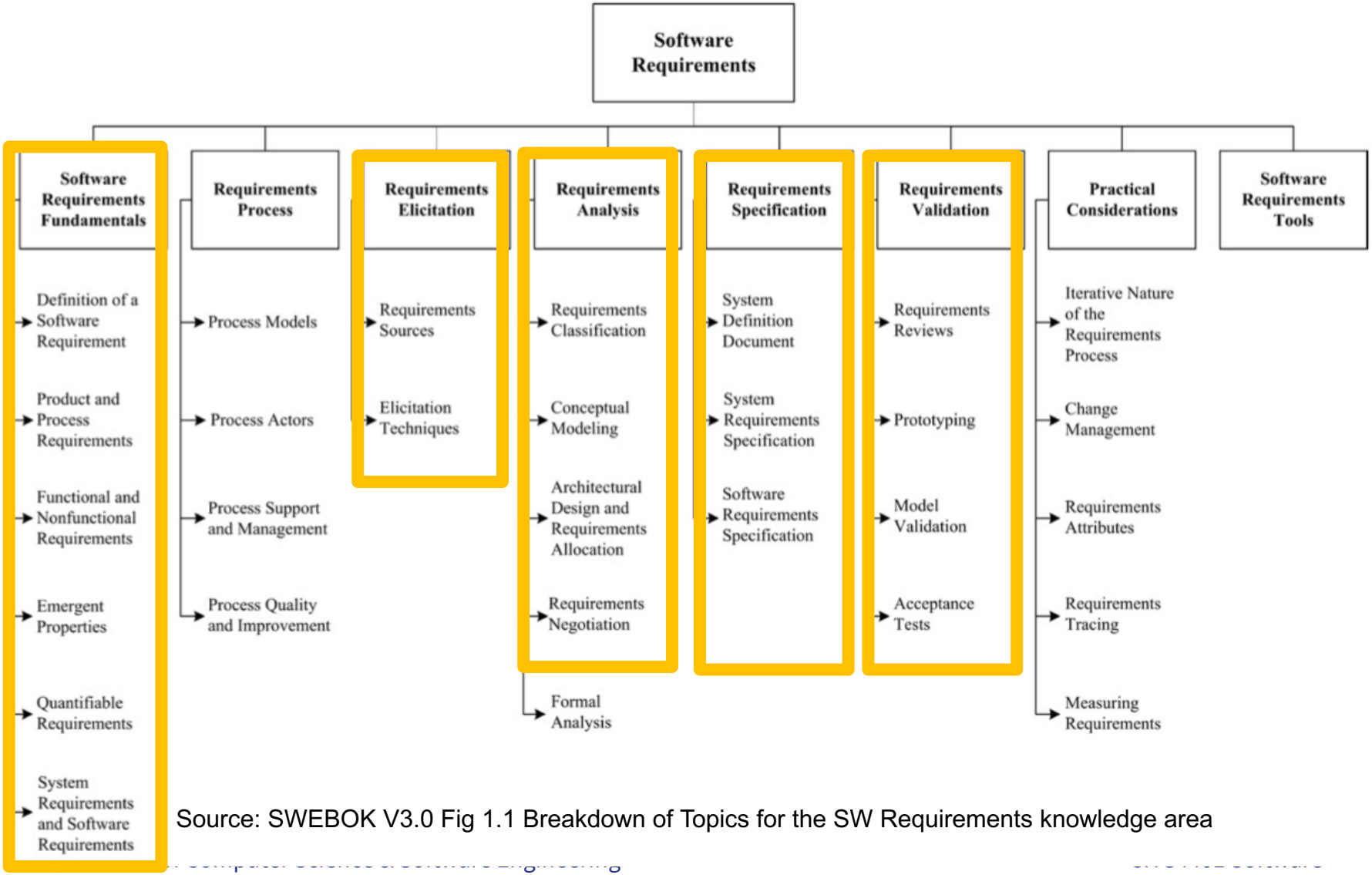
Software Engineering Body of Knowledge

SW Requirements Knowledge



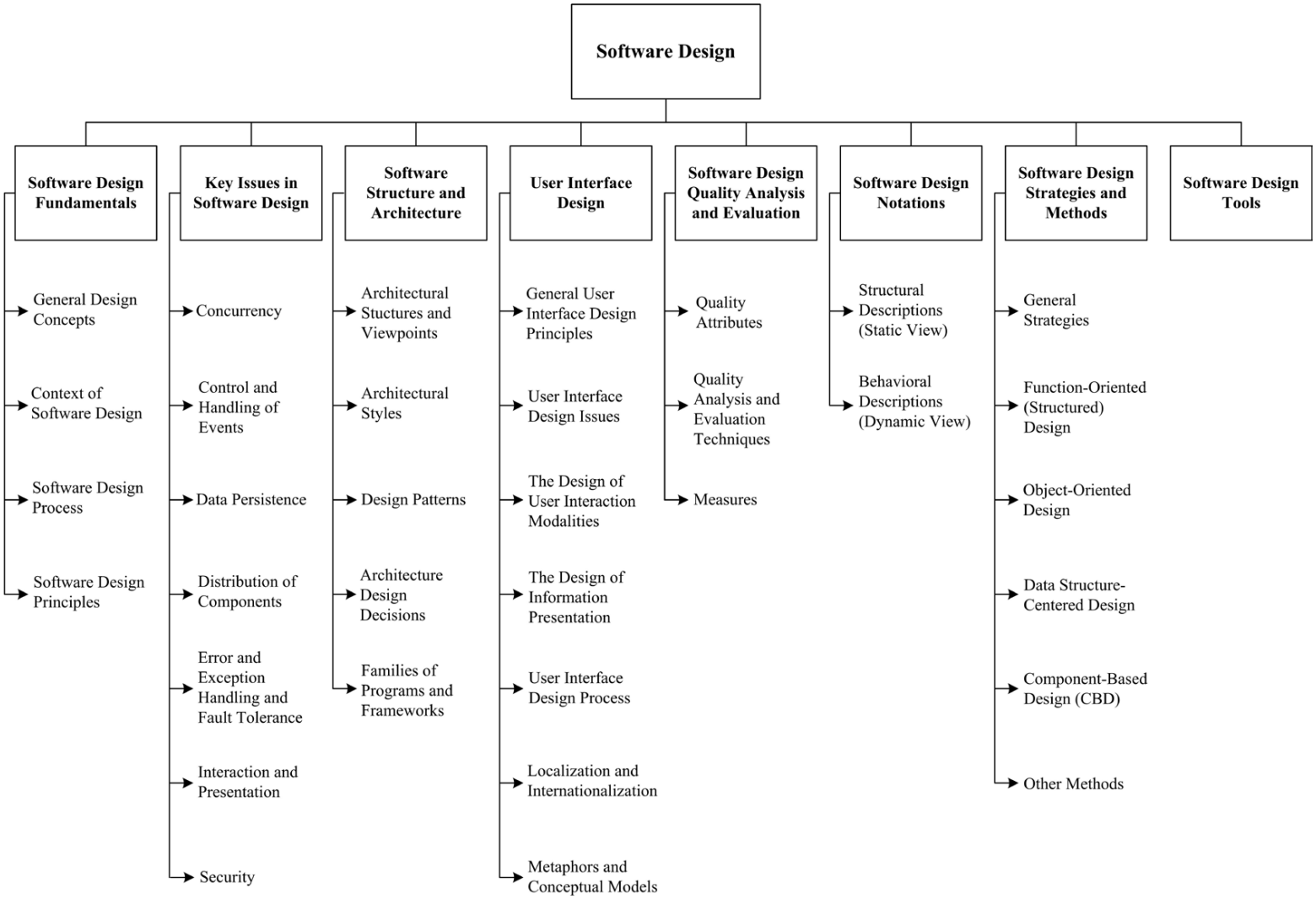
Source: SWEBOK V3.0 Fig 1.1 Breakdown of Topics for the SW Requirements knowledge area

CITS4401 Requirements

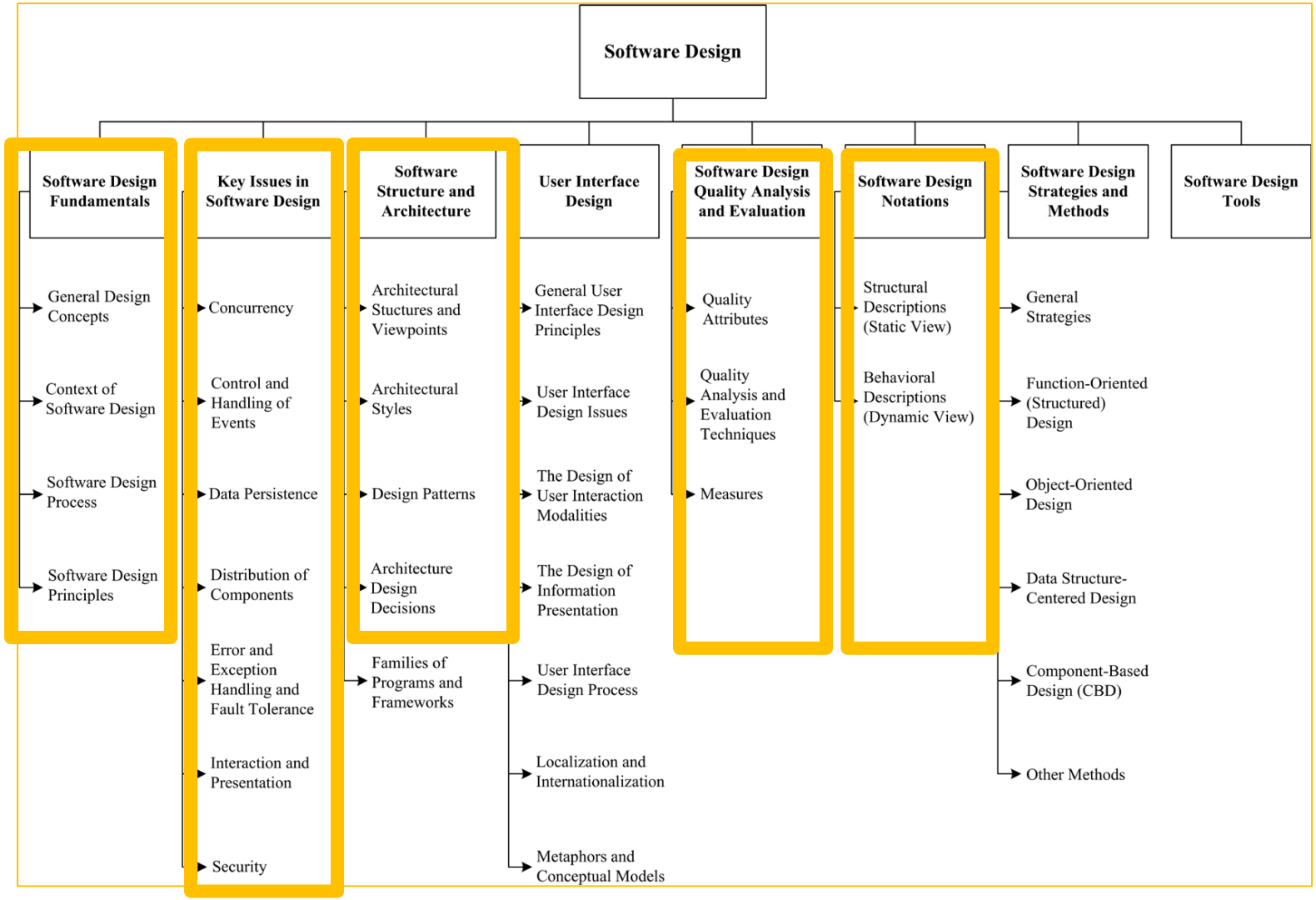


Source: SWEBOK V3.0 Fig 1.1 Breakdown of Topics for the SW Requirements knowledge area

Software Design Knowledge



CITS4401 (Design)



Learning SW engineering

In this unit you will be learning a number of new software engineering methods and techniques. All have strengths and weaknesses.

In Agile SW Development Cockburn discusses 3 levels of understanding new methods and skills

1. **Following:** you know a detailed procedure that works and can follow it exactly
2. **Detaching:** locate the limits of your single procedures; look for rules that explain when it works well and when it does not; adapt your use of the procedure as needed
3. **Fluent:** understand the desired end effect and work towards it; understand trade-offs and selection

In this unit you will mostly be working at the following level with some detaching. Fluency takes years of experience. But keep it in mind as the long term goal.