SE Degree Curriculum Review
Working Group Draft Report (Draft 7 October 2021)

Introduction

The working group's purpose is to look at the area of software engineering and make recommendations to the School of Computer Science and Engineering (CSE) regarding reviewing SE's existing structure and offerings in the light of the changes to the core courses. The working group should as much as possible make recommendations based on evidence from difference perspectives as well as constraints in the university/school.

The composition of the Working Group is:

- Chair: Fethi Rabhi
- CSE members: Jake Renzella
- External members: George Joukhadar (SISTM UNSW), Sherry Xu (Data61)

A number of sub-groups were created to look at particular teaching areas.

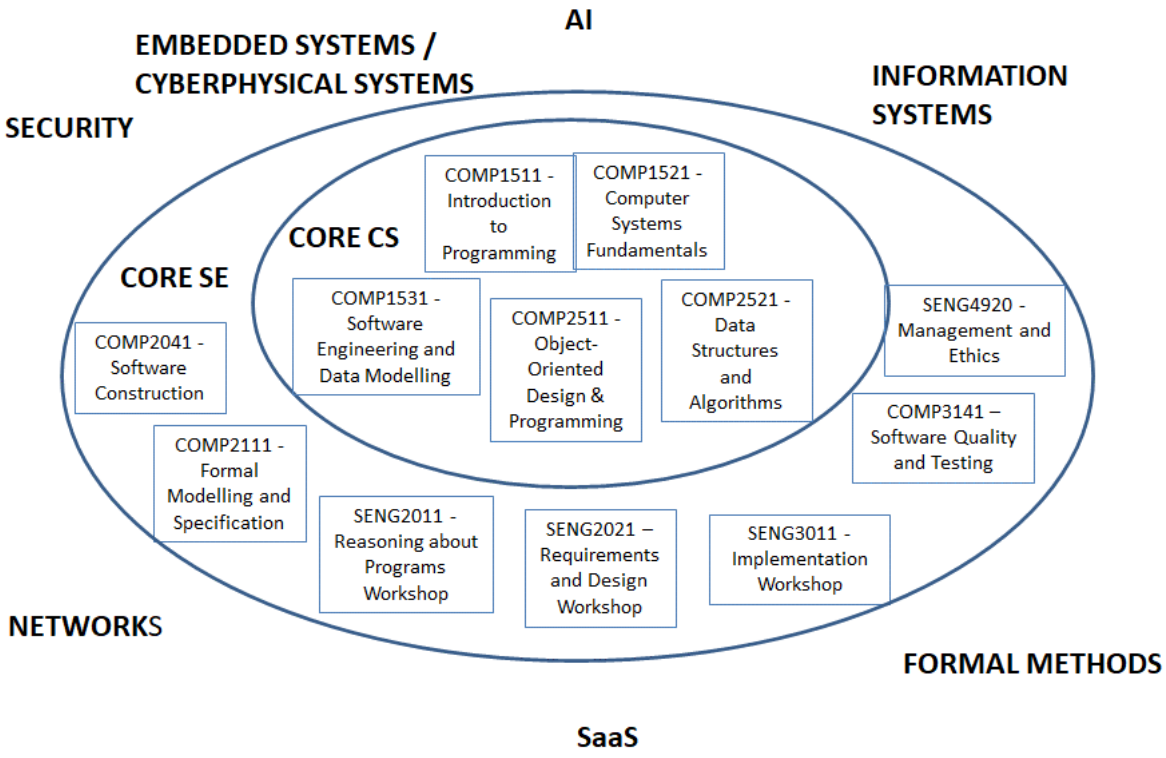
Process followed

The process was similar to the process in previous review, get feedback from different perspectives:

- Summarize the recommendations of the different working groups considering software engineering body of knowledge
- Redefining learning objectives: does existing structure supports these objectives ?
- Industry perspective: are we meeting the needs of industry ?
- Research priorities perspective: are we leveraging our research strengths ?
- Alumni/new students perspectives: are we meeting needs of existing students and aspirations of new ones ?
- New developments: are we in line with new developments in the field ?

A draft SE Degree handbook has been prepared, containing proposed degree structure assuming all changes have been carried out. The draft handbook is provided as an attachment to this report.

As a reference, the structure of the degree at the last review in 2016 is illustrated below.



Specializations recommended in 2016 are listed in Appendix B.

Summary of Sub-groups recommendations

The recommendations are grouped according to Software Engineering Body of Knowledge areas (table of content listed as Appendix A at the end of this document) and which are available for download at Software Engineering Body of Knowledge Version 3 | IEEE Computer Society

Each recommendation is preceded by a number indicating the sub-group it originates from.

- [1] Review of formal methods (Chaired by Carroll Morgan)
- [2] Review of early software lifecycle (Chaired by Fethi Rabhi)
- [3] Review of Software Design (Chaired by Nick Patrikeos)
- [4] Review of Management and Ethics Issues (Chaired by Wayne Wobke)

Formal Methods

Course Name	Recommendations	SWEBOK Sub-Areas
New course (referred to as X6721), based on existing COMP6721	[1] First course on Formal Methods Practice that will become core for the software engineering degree [1] Focus on informal but rigorous reasoning. Explicit teaching of propositional- and predicate calculus. Informal rigour for imperative code; informal rigour for data-structure	1 Software requirements -Requirements process - Requirements elicitation - Requirements analysis - Requirements validation

	abstraction;	
SENG2011	[1] Focus on automated reasoning using Dafny as well as introducing project-oriented tools. [1] showing how to automate some of the informal reasoning introduced in X6721	9 Software Engineering Models and Methods -Modeling -Types of Models -Analysis of Models
COMP2111	[1] Introduce the basic and general theoretical formalisation concepts needed as in introduction to more specific and focused CS theory in later years	9 Software Engineering Models and Methods -Modeling -Types of Models -Analysis of Models
COMP1511/COMP2521	[1] Integrate a small amount of X6721-style content into COMP1511 and COMP2521 in a way that would make some students better programmers, and the others no worse	13. Computing Foundations -Programming Fundamentals -Programming Language Basics -Data Structure and Representation
New (for 1st year)	[1] As an alternative to the above, introduction of an extra core course to relieve some of the pressure on COMP1511 and COMP2521 Data structures and algorithms", which could then allow better integration between informal methods and first-year students	

Early Phases in Software Cycle

Course Name	Recommendations	SWEBOK Sub-Areas
DESN2000 (new course introduced in 2020)	[2] Keep main focus of the course on User-Centred design, design thinking and Requirements Engineering (RE) and early stages of software lifecycle. DESN2000 to become a course which teaches students to see software as a product.	1 Software requirements -Requirements process - Requirements elicitation - Requirements analysis - Requirements validation 2 Software Design -UI Design 7. Software Engineering Management -Review and Evaluation
	[2] Focus on “product management” – which user stories are most important and	

	which features should be developed first – planning sprints at a high level rather than specifics of a project implementation	
	[2] Include the use of a UI design tool like Figma	
	[2] Make COMP151 (Software Engineering Fundamentals) a pre-requisite	
SENG2021	[2] Play down Requirements Engineering, User stories and UI design as these will be part of DESN2000	2 Software Design -Software structure and architecture -Software design notation 13 Computing Foundations -Abstraction -Basic Concept of a System -Database Basics
	[2] Include introduction of a Project Management tool (e.g. Jira) and architecture design, sequence diagrams and data models	
	[4] Project management to be redistributed amongst the SE workshops and DESN2000	
	[3] Make students continue a project started by someone else	
No particular course identified		

Software Design and Programming in the Large

Course Name	Recommendations	SWEBOK Sub-Areas
COMP1531	[2] Should cover testing fundamentals	1 Software requirements -Fundamentals 2 Software Design -Fundamentals 3. Software Construction -Fundamentals 4. Software Testing -Fundamentals 8. Software Engineering Process -Software Process Definition -Software Lifecycles
COMP2511	[2] should cover testing at design level like API testing	2 Software Design -Key issues

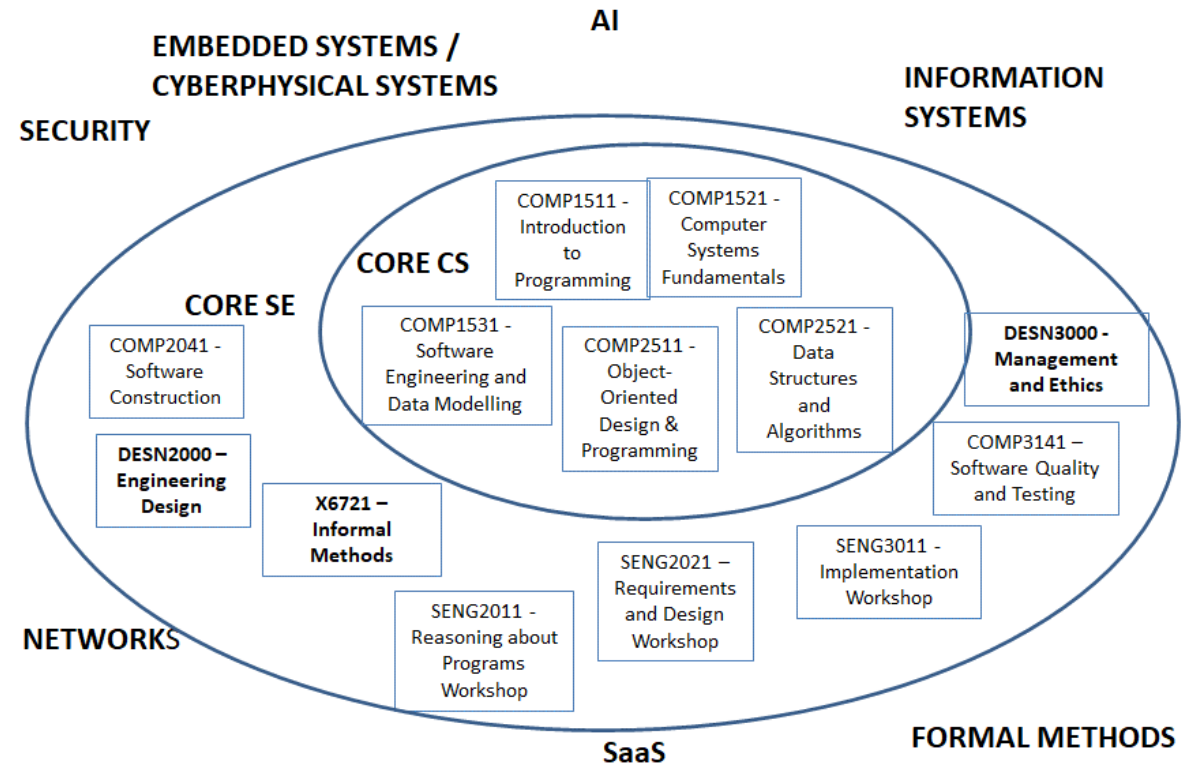
		-Structure and Architecture -Strategies and Methods 3. Software Testing -Design evaluation techniques (not listed)
SENG3011	[2] Include more Project Management	3. Software Construction -Managing Construction -Construction Technologies -Software Construction Tools 7. Software engineering management -Software Project Planning
	[2] Include the use of tools to support complete software development and deployment pipelines	
	[3] Needs more discussion on API integrations in software construction	
	[3] Make students continue a project started by someone else	
	[4] Project management to be redistributed amongst the SE workshops and DESN2000	
No particular course identified but will affect COMP6080 and COMP4511	[3] Need for a dedicated course that teaches Web-Front end programming from a technical perspective building on COMP1531	
No particular course identified	[3] Software quality and testing is not adequately covered. There is minimal coverage of benchmarking, software performance	
No particular course identified	[3] In area of DevOps, need to cover these topics -Multi-instance, lambdas -Advanced continuous integration and pipelines -Cloud distribution -Software maintenance -Managing dependencies	
	[3] In area of software design and construction, need to cover these topics: -Lack of “Good software design” practices:	

	non object-oriented software architecture is not covered in depth (e.g. asynchronous programming and concurrency), more experience and interaction with frameworks, while maintaining understanding of implementations, scalability as a design consideration for software (includes use of microservices), Hyrum's Law as a principle/consideration -Lack of team-code building: reading and working with code written by others, working with a large codebase, writing code as part of a team or organisation -Not enough discussion of data persistence (trivially covered in COMP1531 & 2511) outside of pure database courses	

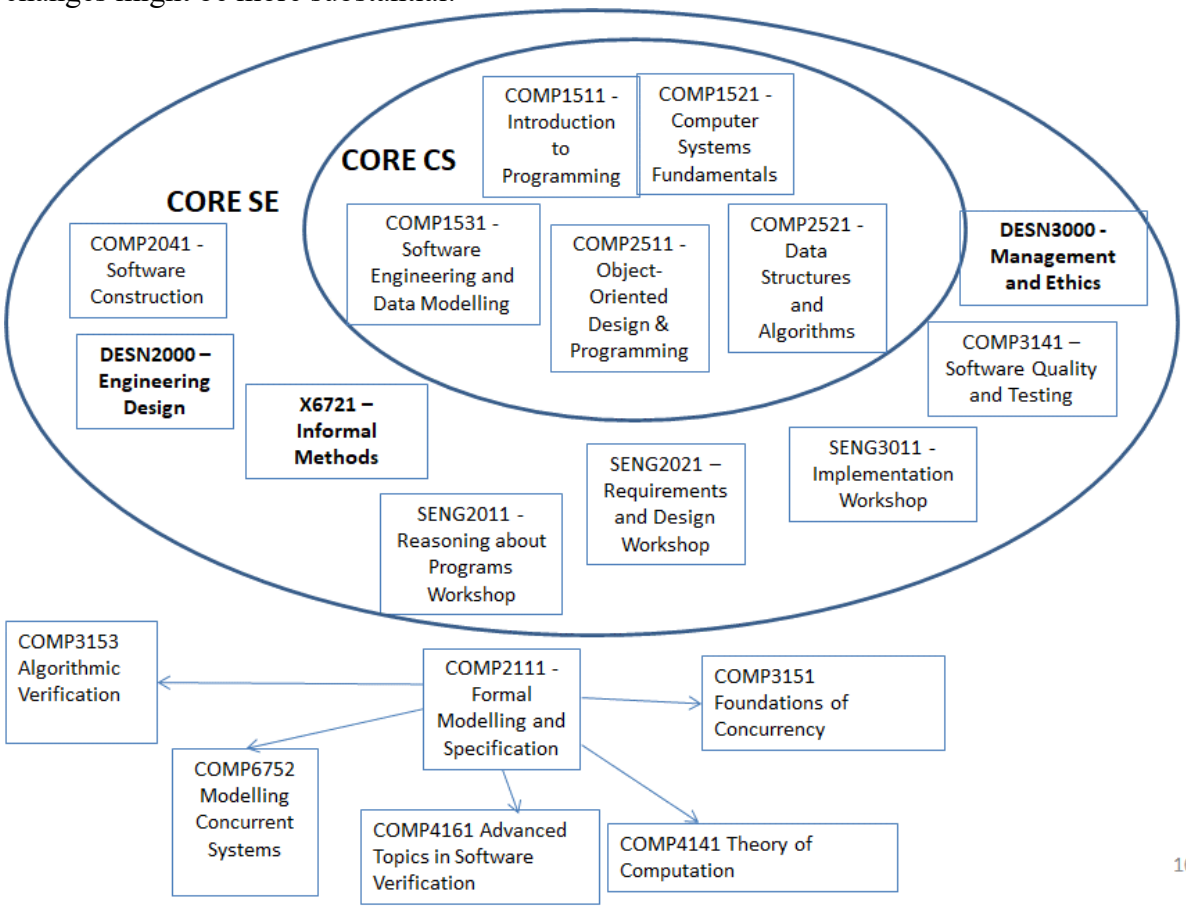
Management and Ethics

Course Name	Recommendations	SWEBOK Sub-Areas
DESN3000 (new course to be introduced in 2022)	[2] can be about lean canvas and creating business plans	11. Software Engineering Professional Practice 12. SE Economics
	[4] DESN3000 to replace SENG4920 as core to the SE programme	
	[4] Work with DesignNext to ensure sufficient coverage of ethics in a Software Engineering context to meet accreditation requirements	
	[4] Include ethical design considerations (such as data privacy, algorithmic decision making bias, etc.)	

As a result of the recommendations, the structure of the degree is as shown below (new courses shown in bold):



Most specializations will have only minor changes except for Formal Methods where the changes might be more substantial:



10

Revisiting Learning Objectives

Comments are invited for possible changes

SLO 1: demonstrate a solid understanding of the software engineering knowledge and skills, necessary to begin practice as a software engineer

SLO 2: ability to appropriately define and apply relevant abstractions from algorithmics, computer science, and mathematics to complex software system development

SLO 3: ability to design and build a system, component, or process to meet desired needs within realistic constraints such as technical, economic, and ethical constraints

SLO 4: ability to think at multiple levels of detail and abstraction encompassing an appreciation for the structure of computer systems and the processes involved in their construction and analysis

SLO 5: ability to think and design software systems from the perspective of the end user and to communicate clearly and effectively with business stakeholders

SLO 6: have the understanding that software interacts with many different domains and the ability to be able to communicate with, and learn from, practitioners from different domains

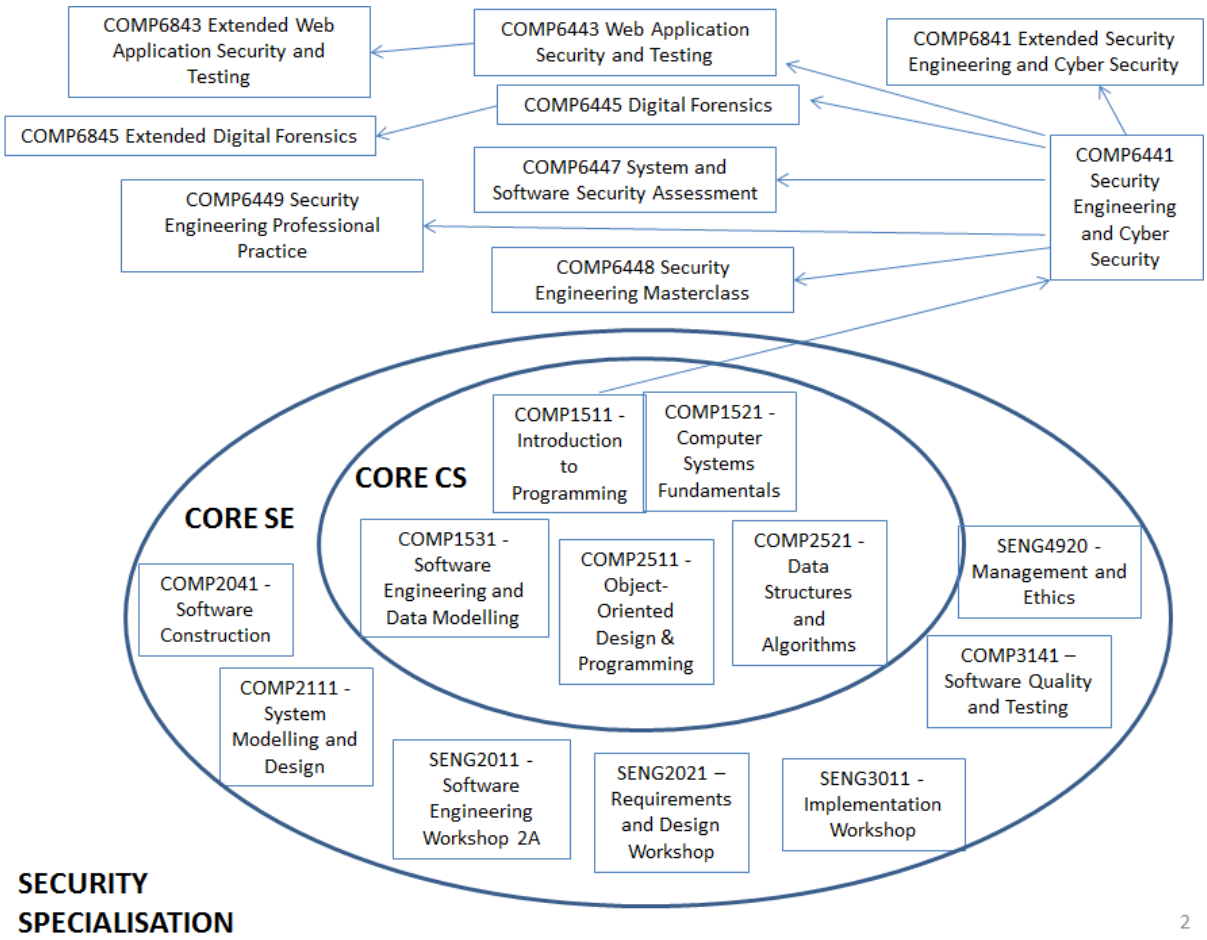
SLO 7: be knowledgeable about current software engineering practices in the workplace, collaborative software development and management processes and their role in the development of quality software systems

APPENDIX A: SWEBOK Areas

APPENDIX B: Specialisations in 2016 review

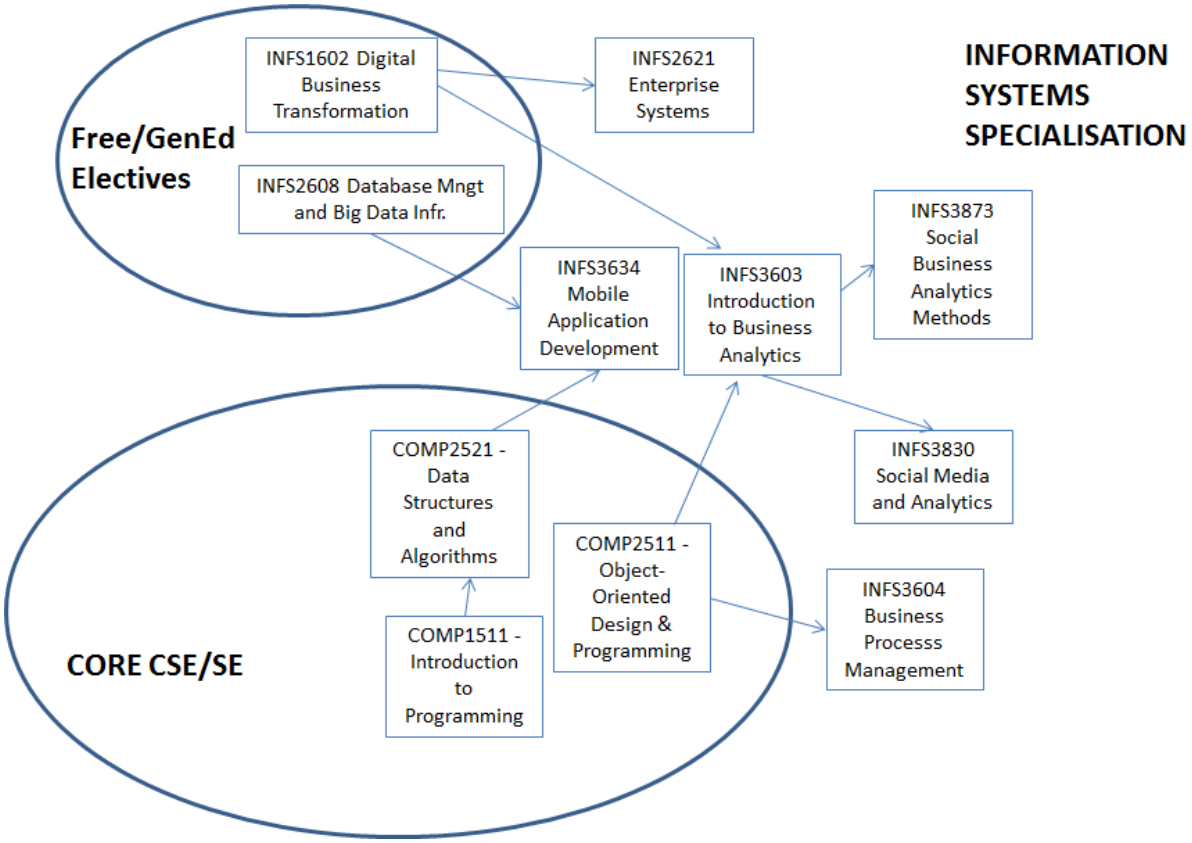
Security specialisation

Defined around new courses funded by CBA initiative.



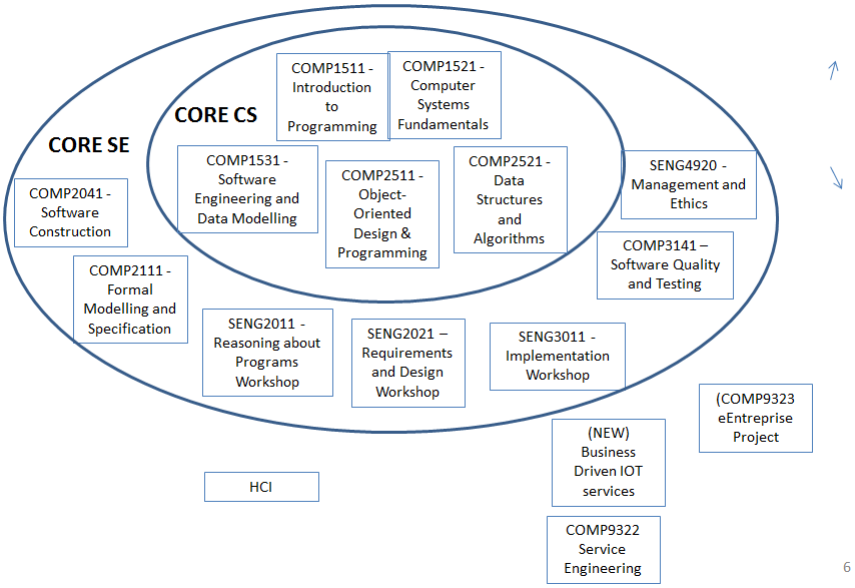
Information Systems Specialisation

Defined around courses offered by SISTM (www.sistm.unsw.edu.au)

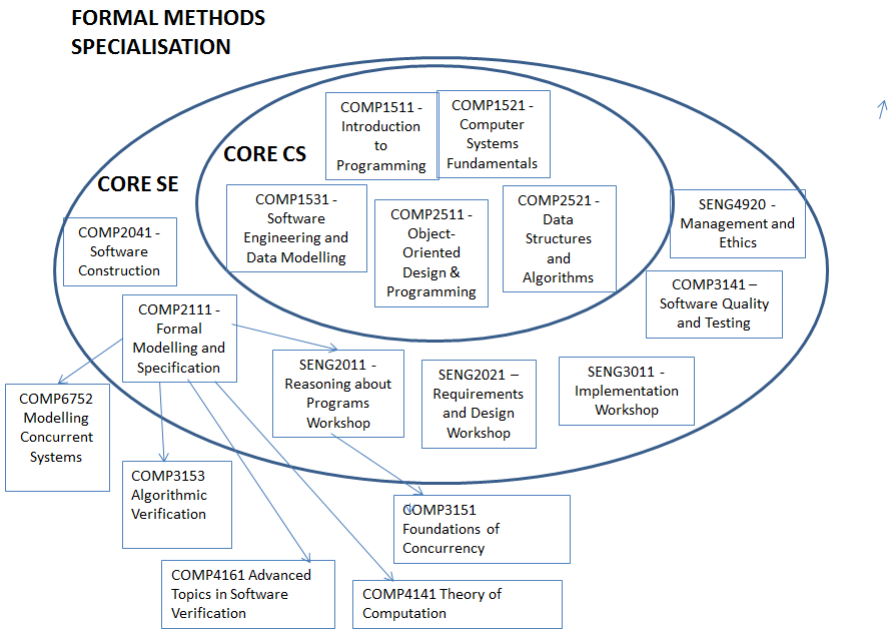


Software as a Service Specialisation

Defined around the work of Service Oriented Computing Research Group
https://sites.google.com/site/unswsoc/



Formal Methods Specialisation

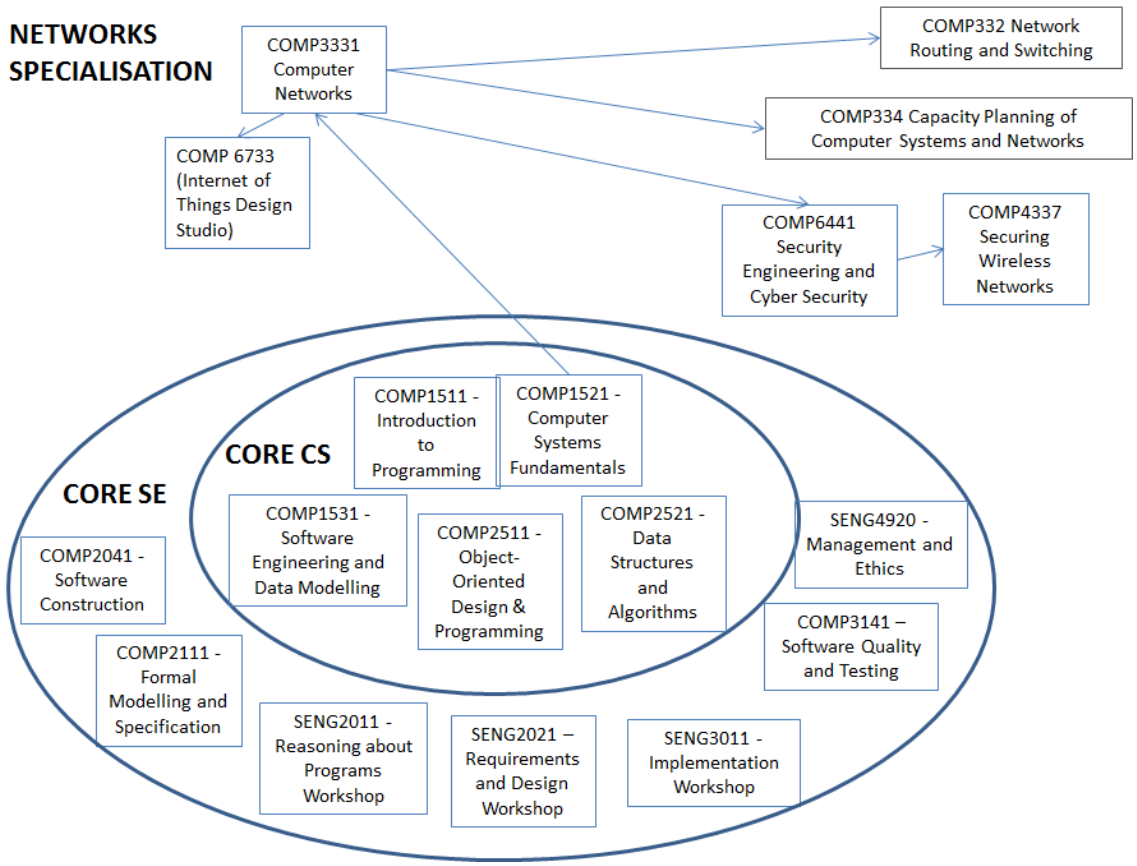


Data science Specialisation (awaiting information)

Defined around the work of Databases Research Group
https://www.engineering.unsw.edu.au/computer-science-engineering/research/research-activities/database-research-group

Networks Specialisation

Defined around the work of Networked Systems and Security Research Group
https://www.engineering.unsw.edu.au/computer-science-engineering/research/research-activities/networked-systems-and-security-group-netsys



Embedded Systems Specialisation (awaiting information)

Defined around the work of Trustworthy (http://ts.data61.csiro.au/projects/TS/) and Embedded Systems Research Groups

AI Specialisation (awaiting information)

Defined around the work of AI Research Group