

# Software Design & Construction

## Working Group Discussion

Software Engineering Degree Revision – 2021

Members:

- Nick Patrikeos (Leader)
- Boualem Benatallah
- Braedon Wooding
- Christopher Di Bella
- Hayden Smith
- Fethi Rabhi (SE Director)

Introduction

1. This working group discussed educational objectives around design construction of software from an industry and academic point of view in the Software Engineering degree, with a lens to reviewing existing CSE courses, changes in the past few years and deficiencies in the curriculum.
2. Courses within the general scope of discussion included:
  - COMP1531 (Software Engineering Fundamentals – Core)
  - COMP2511 (Object-Oriented Design & Programming – Core)
  - COMP6080 (Web Front-end Programming)
  - SENG2021 (Software Engineering Workshop – Core)
  - SENG3011 (Software Engineering Workshop – Core)
3. Areas of the Software Engineering Body of Knowledge (SWEBOK) within the scope of discussion included:
  - Chapter 2 – Software Design
  - Chapter 3 – Software Construction
  - Chapter 4 – Software Testing
  - Chapter 5 – Software Maintenance
  - Chapter 6 – Software Configuration Management
  - Chapter 8 – Software Engineering Process
  - Chapter 10 – Software Quality
4. All ideas listed under ‘possible solutions’ are thoughts as part of the discussions that took place, and in some instances may lack context and are by no means a definitive plan for the future of any course; they are simply suggestions at this point in time. Some of the suggestions listed are mutually exclusive of one another.

## Front-end Programming

5. **Current state:**
  - COMP6080 was introduced in 20T3 – non-core
  - Teaches HTML, CSS, Vanilla JS and React JS
  - Only course in CSE that explicitly focus on teaching web front-end programming
  - SENG2021 has required knowledge of frontend to develop a product
  - COMP4511 also teaches React JS
  - COMP2511 no longer teaches GUI Programming
  
6. **Identified areas for change:**
  - COMP6080 and COMP4511 have overlapping content which could be reduced and made better use of
  - COMP6080 has capacity to have better learning outcomes in frontend if the work is distributed across multiple courses
  - Web development skills should be taught earlier in the degree – required for some courses and useful to know early
  
7. **Possible solutions:**
  - Teach web programming in an earlier core course
    - Was in COMP1531, moved to extension topic due to not enough time to cover, moving back in would mean needing to make space and remove other topics
    - Teach web fundamentals as part of a revised programming fundamentals course – ‘introduction to programming’ at a higher level of abstraction (Python/web, similar to COMP1010) and move C/1511 back:
    - Change the structure of COMP1531 to teach JavaScript instead of Python, allowing for a more web-centric approach and the capacity to build very basic frontends
  - Make COMP6080 a core course
  - Make COMP6080 a prerequisite to COMP4511, and replace web development content currently in COMP4511 with more advanced practical applications of JS, and/or move some content from COMP6080 into COMP4511

## Software Deployment, Maintenance and DevOps

8. **Current state:**
  - COMP1531 introduces the basics of continuous integration and basic deployment of a web server
  - SENG2021 involves deploying a web application
  
9. **Identified areas for change** – areas not covered:
  - Multi-instance, lambdas
  - Advanced continuous integration and pipelines
  - Cloud distribution
  - Software maintenance
  - Managing dependencies (covered briefly in COMP1531)

10. **Possible solutions:**
- Teach above areas as part of SENG2021 in deploying, maintaining and developing a codebase
  - Teach some of the above areas as part of COMP2511 and adjust the focus of the course to be on building and managing good software
    - Difficult without removing content around OO & Software Design

## **Software Design and Construction**

11. **Current state:**
- COMP1531 involves build a Python web server
  - COMP2511 involves building software Java using OOP
  - COMP2511 teaches software engineering design principles, patterns in software architecture design and more generally good software development practices using OO as a vehicle
  - COMP2511 in 21T3 is briefly touching on concurrency as a concept/challenge in software design
12. **Identified areas for change** – areas that are covered, but need more depth:
- “Good software design” is hard to teach in a single course – only enough time to cover the foundations and principles, rather than having students write ‘genuine’ code akin to design and development in industry
  - Writing code as part of a team or organisation is only briefly mentioned – there should be more emphasis on ‘longevity code’
  - More experience and interaction with frameworks, while maintaining understanding of implementations
  - Very little discussion of asynchronous programming, concurrency and parallel patterns (note COMP3151 doesn’t appear to cover parallel computing)
  - Not enough discussion of data persistence (trivially covered in COMP1531 & 2511) outside of pure database courses
  - Minimal discussion of API integrations in software construction in SENG3011
13. **Identified areas for change** - topics not covered:
- Students and graduating students moving into industry struggle with reading and working with code written by others, working with a large codebase as a starting point is not taught in courses currently
  - Should teach Hyrum’s Law as a principle/consideration
  - Missing knowledge around scalability as a design consideration for software (includes use of microservices)
  - Non object-oriented software architecture is not covered
  - Little discussion of full-stack architecture and ‘bringing everything together’

14. **Possible solutions:**
- Continue discussions around good software design in a course after COMP2511, possibly applying in SENG3011
    - Incorporate as part of discussion on software quality
  - Focus SENG2021/SENG3011 around a project that has been started by someone else and 'inherited' by the team to understand and build on
  - Discuss full-stack architecture design considerations and microservices (somewhere, potentially SENG2021 and/or 3011)
  - Teach 'using abstractions while understanding implementations' and work with particular frameworks as case studies/examples
  - Discuss persistence and databases as part of SENG2021
  - Discuss asynchronous programming and concurrency more (somewhere)
  - Replace Java in COMP2511 with Kotlin and spend more time discussing structured concurrency
  - Discuss 3<sup>rd</sup> party API integrations as part of SENG2021/SENG3011

## Software Testing & Quality

15. **Current state:**
- COMP1531 (in 21T3) discusses the fundamentals of testing, mainly focused on API testing with pytest, test-driven development, an overview of verification and validation and basics of test design
  - COMP2511 (in 21T3) discusses testing theory and the design of tests, and briefly the evaluation of design
  - COMP2511 discusses software quality in terms of good software design
  - COMP1531 discusses acceptance testing briefly as part of requirements engineering
16. **Identified areas for change:**
- Not enough emphasis on testing and test-driven development throughout software construction courses
  - More advanced types of testing (volume, regression, smoke, fuzzing) not taught
  - No course that focuses on / discusses in depth 'Software Quality' as a whole
  - Minimal coverage of benchmarking, software performance
17. **Possible solutions:**
- Discuss testing throughout the software construction courses and reinforce its importance
  - COMP2511 could cover more on testing (one lecture on it currently) and tests as a means for designing and prototyping
  - DESN2000 to cover acceptance/usability testing and only discuss at a high level in COMP1531
  - Discuss Software Quality and benchmarking, software performance and advanced types of testing as part of SENG2021 and/or SENG3011, or as its own new course

## **Conclusion**

18. The points above list identify requirements and suggestions for possible changes to CSE courses for review by the broader degree review working group, industry and the CSE Education Committee.

3 October 2021

### Courses Map

The following 'map' suggests ways some of the above recommendations could be integrated into the SE curriculum. It highlights suggested changes to some existing courses and in other places links to current curricula.

