Cand. It. in Software Design

Marco Carbone, Søren Debois, Yvonne Dittrich, Peter Sestoft, Andrzej Wasowski

March 16, 2017 draft.

Overview

This document describes the design and implementation of the ITU Software Design programme (SD). The programme is a two-year Master of Science (kandidat) education in software design intended for students holding an undergraduate degree in a subject not related to computing.

The programme is an “Infusion master”: it takes students with a solid prior grounding in a non-computing discipline and *infuses* foundations of applied computer science. Upon graduation from the ITU, the successful student will be able to apply computational thinking and tools to the field of his prior education.

The SD programme is a re-design of the existing ITU Software Development programme, Design Track (SDT/DT). This document assumes a reader with a working knowledge of that programme.

Differences from SDT/DT

The major operational and philosophical difference between SDT/DT and SD is in the aim and structure of specialisations. An SD specialisation is targeted at a particular set of BSc students, infusing particular computational skills and thinking. SD features three specialisations, aimed at Social Science BSc’s[1], Arts and Humanities BSc’s, respectively Science BSc’s.

SD improves on progression in SDT/DT in small, but significant ways, chiefly by posing formal requirements from later courses to earlier ones, for both specialisation and mandatory courses.

Re-design Considerations

The programme is a redesign of the SDT/DT. This redesign has two major objectives:

1. To address comments in the recent institutional accreditation [?] that SDT/DT has insufficient progression.

[1] We are interested primarily in social science BSc’s with a quantitative grounding, e.g., Economics, Business, potentially Political Science.

2. To address comments from employers that SDT/DT and SDT/AC candidates are difficult to distinguish formally.

Ad (1). The redesign address these comments by re-designing specialisations to cater to specific backgrounds. For instance, the Business Analytics specialisation is targeted at economics, political science and social science BSc's; courses in the specialisation combine with this background to provide a unique new skill set and academic perspective (e.g., combining an Economics BSc with Data Mining courses).

Besides ensuring progression through the infusion of computational thinking into existing minors, the re-design further ensures progression via (a) making specialisation courses build explicitly on mandatory ones, and (b) making specialisation courses build explicitly on each other.

Ad (2). Candidates on both SDT/DT and SDT/AC are labelled simply SDT upon graduation, as a consequence of both custom and law on university degrees. By splitting the SDT programme into Computer Science (CS) and Software Development, the differences between the two types of candidates are made formally explicit. CS becomes a new programme, whereas SD is a re-design of SDT/DT.

Note that SD is a re-design and continuation of the "Design Track" of the extant SDT programme.

Competences and Job Profiles

The successful SD graduate, like previous SDT/DT candidates, will typically find his first employment in one of the following roles:

1. As a software developer or software engineer.
2. As an analyst.
3. As a junior project manager.
4. As a junior product owner or -manager, or deputy to a product owner or manager.
5. As an IT consultant.

The exact nature of the graduate's role and the expected career trajectory of the candidate is strongly correlated with the graduate's BSc.

For BSc's in social sciences, we expect an overrepresentation of analysts, consultants and specialised software developers: The graduate is applying his original BSc, especially its data-driven, analytical component, infused with computational tools and thinking to his employment.

For BSc's in arts and humanities, we expect an overrepresentation of consultants, project managers and product owners and -managers: The graduate is applying his original BSc, especially its communication- and human-centric components, infused with a solid skills in the technical foundations, realities and tools of contemporary software development.

For BSc's in science, we expect an overrepresentation of software developers and engineers: The graduate is applying his original BSc, especially its analytical and logical componts, infused with software development tools and methodologies.

With very diverse starting points, a significant minority of SD graduates (with no obvious correlation to BSc topic), is expected to find alternative employment, in particular in organisation infrastructure and entrepreneurial activities.

Programme Structure

The SD overall programme structure is mostly unchanged from SDT/DT. SD comprises

1. 6*7.5 CTS Mandatory courses
2. 3*7.5 ECTS Specialisation courses and/or projects
3. 2*7.5 ECTS Elective courses and/or projects
4. 1*7.5 ECTS Research project (see below)
5. 30 ECTS Thesis

The programme enrolls student only once a year. Students are expected to follow the schedule below.

1st Semester	**Introductory Programming** (7.5 ECTS)	**Programming Workshop** (7.5 ECTS)	**Discrete Mathematics** (7.5 ECTS)	**Software Engineering and Software Qualities** (7.5 ECTS)
2nd Semester	**Introduction to Databases** (7.5 ECTS)	**Algorithms and Data Structures** (7.5 ECTS)	**Specialisation Course 1** (7.5 ECTS)	**Specialisation Course 2** (7.5 ECTS)
3rd Semester	**Elective Course or Project** (7.5 ECTS)	**Elective Course or Project** (7.5 ECTS)	**Research Project** (7.5 ECTS)	**Specialisation Project** (7.5 ECTS)
4th Semester	**Master Thesis** (30 ECTS)			

Note that the SD programme **does not** participate in the cross-disciplinary project on 3rd semester. As an infusion master, SD already emphasises cross-disciplinary activities.

The 3rd semester research project takes the role of the current Thesis Preparation course. It is intended to give students a running start on their thesis. It expected but not required that the supervisor for the Research Project also supervises the Master Thesis. It is required that the Research Project gives the student a firm foundation in the research literature in his would-be thesis topic, e.g., by producing a literature review.

The Research Project solves the problem of insufficient depth and too weak a connection to thesis supervisors and topics by making that connection explicit and (partly) formalised in intended learning outcomes of the Research Project.

Progression

The SD programme progresses linearly from one semester to the next:

1. Mandatory courses on the second semester require (the skills provided in) the mandatory courses on the first.
2. Specialisation courses on the second semester require (the skills provided in) the mandatory courses of the first.
3. Specialisation courses infuse computational thinking and tools into student's existing bachelor.
4. The final specialisation course requires (the skills provided in) 2nd semester mandatory and specialisation courses.
5. The Research Project course on the third semester likely requires (the skills provided in) preceding courses.

Specialisations

A specialisation on SD is the key vehicle for ensuring progression from the student's prior bachelor to his ITU master [?]. To this end, each specialisation (a) builds on mandatory courses and (b) targets a specific class of prior bachelor's.

At the time of writing, the exact specialisations and their contents is not settled. Below is the current set of suggestions. The working group broadly agrees on "Business Analytics".

1. *Business Analytics.* For social science BSc's, in particular political science and economics. This specialisation infuses data analytics and computational skills into students with a strong background in fields that emphasise data and data analytics. Upon completing the specialisation, the student will have computational skills in his BSc field beyond what is available to Masters in that field, with a field-specific understanding beyond what is available to Computer Scientists.

   The specialisation comprises the courses "Big data management (technical)", "Data mining", and "Intelligent systems programming".

   This specialisation exists on SDT/DT.

2. *Technical Interaction Design*. For arts and humanities BSc's. This specialisation infuses interaction design from a technical point of view, allowing the student to combine SD's mandatory courses, his prior BSc—especially wrt.~communication skills—and central tenets of interaction design. Upon completing the specialisation, the student will have unique skills in and perspectives on contemporary user-facing software development, with technical skills and understanding beyond what is

available on other UX-centric educations (e.g., DDK), yet communication skills and understanding beyond what is available on traditional Computer Science educations.

The courses for this specialisation has yet to be settled.

3. *[As yet unnamed.]* For science BSc's. This specialisation infuses programming, process, and software development skills, with the aim of launching the student on a career of eventually managing software developers or other engineers. Upon completing the specialisation, the student will have a unique combination of analytical skills afforded by his existing Science BSc and understanding of at-scale software development processes and techniques.

   This specialisation is envisioned to comprise courses such as "Domain Specific Languages" or "Advanced Programming", "Enterprise Architecture", Software Architecture and a (potentially new course in) "End-user development", Software Eco-Systems, Software Product Lines.

Ad (3). This specialisation is not a traditional management education: Rather than teaching theory and practice of management in and of itself, it gives a solid grounding in the basis of software engineering that a budding manager of technical personnel *must* possess in order to be effective. This specialisation is motivated by the insight [citation needed] that SDT/DT candidates with a Science BSc *tends* to wind up as managers within a few years of their first jobs.

At the ITU, the number of courses and specialisations available to a study programme is a function of the number students enrolled in that programme. At current projections, SD likely cannot sustain three specialisations. However, anticipating short-term growth in SD enrollment, ITU management has expressed sympathy for sustaining three SD specialisations.

Short Course Descriptions

[These will *generally* not change from the extant SDT/DT programme.]

Intended Learning Outcomes

[These will *generally* not change from the extant SDT/DT programme.]

Target Group

The SD programme accepts student with a strong BSc in a topic not traditionally considered “Computer Science” or the like. The SD programme *generally* prefers students with a BSc in the social sciences, arts & humanities, or the (natural) sciences, but students with other non-Computer Science bachelors are encouraged to apply. The enrollment decision is made by the Head of Study Programme on a case-by-case basis.

Admission Requirements.

The requirements for being admitted to this programme are listed below:

- The applicant must hold a bachelor.
- The applicant must *not* hold a bachelor in Computer Science or the like.
- ?

Application Structure.

Besides the formal requirements from the IT University, the applicant must submit a cover letter (max 2 pages) containing the following fields: [...]