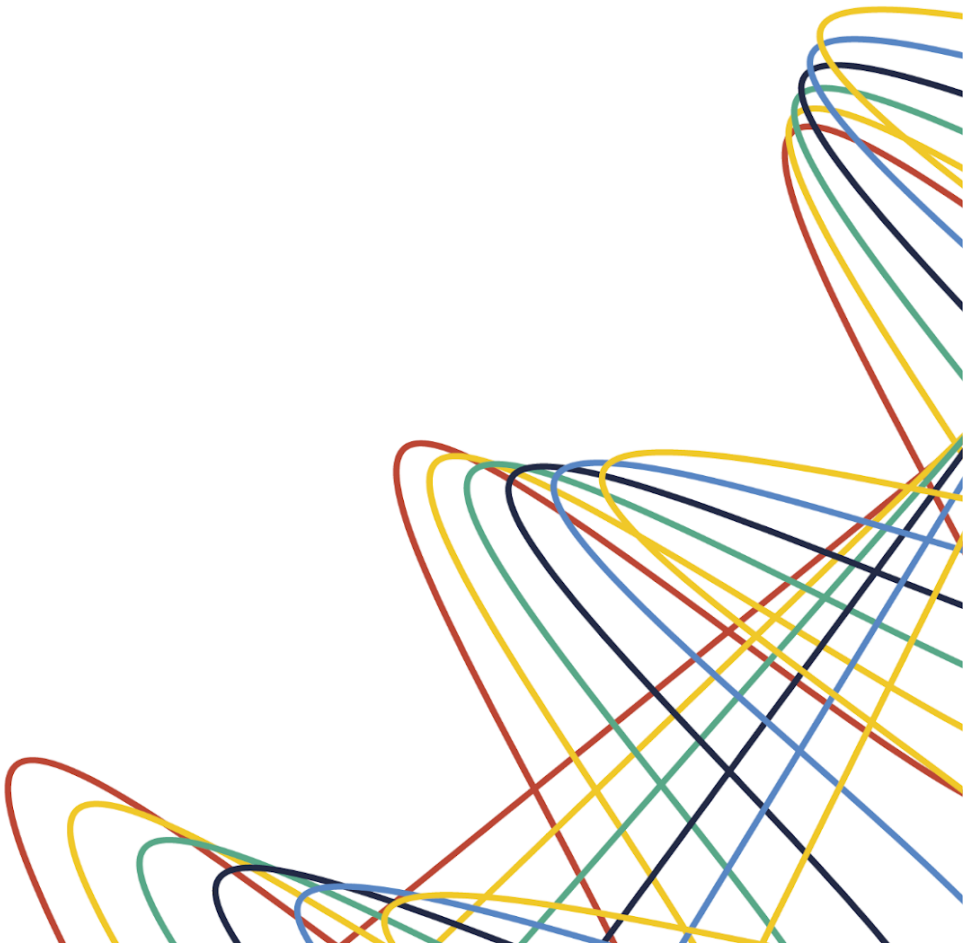
CONSTRUCTOR INSTITUTE

MSc in Quantum Software Engineering and Computer Science (QSECS) handbook



Identification: Constructor Institute Report CI-2023-012-QEC

Title: *MSc in Quantum Software Engineering and Computer Science (QSECS) handbook*

Status:

☑ Public
☐Available to Constructor Institute members only (on Intranet)
☐ Restricted availability, subject to approval

Related documents (if applicable):

- International Student handbook, Constructor Institute Report CI-2023-004-ISH
- Code of academic ethics, Constructor Institute Report CI-2023-008-CAE

Date	Authors	Comments about this version
2023-01-07	Manuel Oriol	• Approved by President 03.08.2023 • Approved by Academic Senate 09.08.2023 • Initial version

Table of Contents

1 Program overview

1.1 Concept

The future of software is quantum. This program is the first-of-its-kind program that addresses quantum computing, software engineering, and the leadership skills needed for this future.
The complex, ambitious, and innovative software systems of the future, developed by successful technology companies, require a unique type of leader who possesses both in-depth technical knowledge and excellent leadership abilities. Traditional master's programs often focus on training software developers or classical managers, but they frequently fail to produce the digital-technology-savvy leaders that leading-edge companies require for positions such as Product Manager, Chief Information Officer, Chief Security Officer, Chief Technology Officer, and others that demand competence in both domains.

The Master of Science in Quantum Software Engineering and Computer Science (QSECS) at Constructor Institute Schaffhausen addresses this urgent need. This research-oriented master's program offers a wide range of courses and projects covering various quantum technologies, computer science, and software engineering topics, along with essential management and leadership skills.

To prepare students for technology leadership roles in research and industry, the QSECS master's program provides comprehensive software engineering training, encompassing both development and management. Additionally, it includes core technical courses in two areas that are central to modern IT systems:

- Software engineering
- Quantum technologies

Courses are taught by field experts, including permanent faculty members from Constructor and guest lecturers from companies.

Upon completing the master's program, students will have acquired the core expertise of digital leaders, with a strong technological foundation, as well as essential management and leadership skills, making them ready to assume active leadership roles in the digital technology field. Constructor University's multi-country setup equips students to play a significant role in today's global and multiethnic societies.

While many students join the industry in high-profile positions, the QSECS program also serves as excellent preparation for Ph.D. research in software engineering, quantum technologies, AI, and other advanced topics.

As a consecutive Master's program, the MSc in Quantum Software Engineering and Computer Science (QSECS) is designed for graduates of undergraduate programs in computer science or physics with a strong background in programming. Upon graduation, students will have acquired a portfolio of skills in highly relevant areas of computer science and quantum technologies. They will develop their creative and problem-solving abilities to produce, develop, and evaluate solutions for technical challenges. Additionally, they will gain knowledge about the state of the art in a selected subject area and learn the skills necessary to approach, develop, and document small independent projects related to the latest research, industrial applications, and even startups.

Many modules in the software engineering subject area will be taught by distinguished experts in the field at Constructor Institute Schaffhausen. Students at Constructor Institute Schaffhausen have seamless access to courses from Constructor University Bremen. This collaboration enables quick access to real-world applications and the IT job market through Constructor's excellent international network.

To enhance the educational concept, the program utilizes contemporary blended e-learning techniques. The program is primarily in-person, but all lectures are recorded and can be recovered offline. In addition, flipped classroom teaching, where applicable, allows for a student-centric and hands-on experience. Team-based work on software projects and other activities also benefits from agile development concepts. With state-of-the-art software and hardware equipment, seamless collaboration among students and instructors from different institutions is facilitated, and it naturally adapts to conditions that may arise from pandemic emergencies.

1.2 Qualification aims

1.2.1 Educational aims

Digitalization is the backbone of industry and society, with software and digital companies playing a crucial role. Leading companies have a pressing need for a new generation of digital experts. The complexity of software and digitization requires experts with deep technological knowledge and crosscutting technical management and leadership skills.

The Quantum Software Engineering and Computer Science program aims to provide an in-depth understanding of designing, maintaining, and analyzing digital systems, as well as quantum systems. Students will acquire the skills necessary to successfully and responsibly engineer software and quantum-based systems using appropriate methods and tools. The program seeks to enhance participants' competencies and capabilities in the fields of Software Engineering and Quantum Technologies, which play a dominant role in industries and research. The curriculum also complements this Quantum Software Engineering and Computer Science education by teaching modern cross-disciplinary leadership and management competencies to prepare students for future digital leadership roles.

Practical and research-oriented work is introduced through a Capstone project, an elective research project, and a thesis, supported by frequent individual feedback sessions and personal guidance. This facilitates students' career development and enables them to become valuable assets in industries and research within a short period of time.

Constructor Institute Schaffhausen programs are offered in a highly intercultural environment. Students acquire intercultural competence as part of their education through everyday group work, class participation, and extracurricular activities. This helps students gain practical intercultural skills and build confidence in an English-speaking work and study environment. Presenting a strong, confident appearance and effectively communicating in various cultural contexts are essential abilities for internationally successful executives in any business area.

In summary, graduates of Quantum Software Engineering and Computer Science will possess the following competencies and skills:

- **Subject-matter competence in a Computer Science specialization**: Graduates have in-depth knowledge of software engineering. They can define and interpret the principles of the field and have developed a detailed and critical understanding at the forefront of knowledge in the field.
- **Quantum Software Engineering and Computer Science:** Graduates have broadened and deepened knowledge in formal, algorithmic, and applied competencies in Computer Science and Quantum Technologies. This enables them to develop independent ideas as digital experts. Responding to the high demand in industry and the increasing interest in research software, graduates have also acquired broader knowledge in software engineering and quantum computing, enabling them to solve practical and scientific problems in the field.
- **Learning, transfer, and research skills**: Graduates are capable of learning new methodologies using theoretically grounded approaches, which allows them to apply problem-solving techniques in new and unfamiliar situations. They integrate learned skills in complex and

multidisciplinary contexts, as is increasingly necessary in industry and research. Graduates can design research questions, select appropriate methods, and document and interpret research results.

- **Management and leadership skills**: Recognizing the growing need for management and leadership skills in business, industry, and research, graduates have a broad and integrated knowledge and understanding of the fundamentals of management and leadership. Their knowledge aligns with standard literature in the field, and they can solve related problems in the field of quantum technologies, computer science and software engineering with professional plausibility.
- **Teamwork and communication skills**: Graduates are proficient in exchanging specialized ideas within a group setting, with the goal of collaboratively developing digital software or hardware systems. They practice effective and reflective communication and collaboration on both academic and non-academic topics.
- **Personal and professional competence**: Graduates are capable of developing a professional profile both within and outside academia. They can make, justify, and reflect on decisions based on theoretical and professional knowledge. They critically examine their own behavior and assess social consequences. They adapt appropriately to various situations, including international environments, and continually develop their professional actions.

1.2.2 Intended learning outcomes

By the end of this program, students will be able to:

1. Critically assess and creatively apply technological possibilities, solutions and innovations in real-life situations, organizations, and industries in the fields of quantum technology, software engineering and computer science.
2. Use, adapt, and improve modern quantum technology.
3. Write code quantum applications.
4. Design, implement, and exploit methods in cryptography and security-related fields.
5. Plan, conduct, document, and present small research projects in the context of quantum technologies, software engineering and computer science with appropriate language skills.
6. Use scientific methods as appropriate in the field of Quantum Software Engineering and Computer Science, such as defining research questions, justifying methods, collecting, assessing, and interpreting relevant information, and drawing scientifically founded conclusions that consider social, scientific, and ethical insights.
7. Apply cross-disciplinary management methodologies to solve academic and professional problems.
8. Take on responsibility and communicate effectively in a diverse team.
9. Take responsibility for their role in society, evaluating critical feedback and self-analysis.
10. Engage ethically and actively contribute to a sustainable future, reflecting and respecting different views.
11. Critically assess and integrate a consistent toolset of leadership abilities.
12. Design quality for various systems (quantum and non-quantum)

Educational aim	ILOs	Notes
Subject-matter competence in a Computer Science specialization	Design quality for various systems (quantum and non-quantum)	
Quantum Software Engineering and Computer Science	Critically assess and creatively apply technological possibilities and innovations in the fields of	

	quantum technology, software engineering and computer science. - Critically assess and apply software engineering methodologies considering real-life situations, organizations, and industries. - Use, adapt, and improve modern quantum technology. - Code quantum applications - Design, implement, and exploit methods in cryptography and security-related fields. - Develop and advance solutions to problems and arguments in their subject area and defend these in discussions with specialists and non-specialists.	
Learning, transfer, and research skills	- Plan, conduct, and document small research projects in the context of quantum technologies, software engineering and computer science. - Independently research, document, and present a scientific topic with appropriate language skills. - Use scientific methods as appropriate in the field of Quantum Software Engineering and Computer Science, such	

	as defining research questions, justifying methods, collecting, assessing, and interpreting relevant information, and drawing scientifically-founded conclusions that consider social, scientific, and ethical insights. Perform research on quantum software engineering and computer science	
Management and leadership skills:	- Apply cross-disciplinary management methodologies to solve academic and professional problems. Take on responsibility in a diverse team	
Teamwork and communication skills		There are no ILOs on these skills, I will add communication to one of the skills (see below)
Personal and professional competence:	Apply their knowledge and understanding to a professional context Take responsibility for their own learning, personal and professional development, and role in society, evaluating critical feedback and self-analysis Adhere to and defend ethical, scientific, and professional standards Engage ethically with academic, professional, and wider communities and actively contribute to a sustainable future, reflecting and respecting different views Critically assess and integrate a consistent toolset of leadership	

	abilities into a professional work environment.	

1.3 Target audience

The program is designed for students from different geographical and cultural backgrounds. It caters to graduates of computer science and related undergraduate programs who wish to specialize or deepen their knowledge in Quantum Software Engineering. It also welcomes physics graduates or those from closely related fields who want to focus on quantum and software engineering. The program particularly targets candidates dedicated to gaining theoretical and application-oriented knowledge in Software Engineering, Quantum Computing, Quantum Cryptography, and Quantum Communication.

Applicants are required to have completed their first degree in Computer Science or a closely related subject, or a first degree in physics, and possess strong coding skills prior to admission.

The program prepares students for key roles in the IT and Quantum industries, as well as for research in these fields. To support this, the program offers additional education in management and leadership courses, enabling students to develop their own start-ups. The educational approach of the program emphasizes exchange and discussion within the student community. Therefore, a willingness to interact, appreciate different teaching and learning formats, accept challenges, and engage in professional development during their studies are important prerequisites for successful participation in the program.

1.4 Career options and support services

Computers are indispensable in our society, playing a vital role in various domains. The complexity associated with quantum computing presents further challenges. Computer Science researchers contribute new insights across a wide spectrum of disciplines, while IT practitioners work in diverse industries such as business, government, finance, energy, education, healthcare, and aerospace. Having trained in the three aspects of software engineering, quantum and leadership, graduates are prime candidates for becoming digital-technology-savvy leaders that leading-edge companies require for positions such as Product Manager, Chief Information Officer, Chief Security Officer, Chief Technology Officer, and others that demand competence in both domains.

Software Engineering graduates are in high demand worldwide adding quantum technology, and leadership makes our graduates one step beyond others. Job opportunities in software engineering abound, allowing graduates to choose from a wide range of highly paid positions. Constructor's Alumni Association helps students build a global network that can aid in exploring career opportunities in start-ups, industry, and academia. Moreover, the Constructor Group's extensive industry network provides excellent access to leading technology enterprises.

1.5 Admission requirements

The Quantum Software Engineering and Computer Science graduate program requires applicants to have completed an undergraduate program in computer science, software engineering, information technology, or a related discipline with at least 60 ECTS credits in computer science-related topics (such as mathematics, programming, design, and software architecture) or physics. Conditional admission may be granted to students who do not meet these requirements, subject to additional coursework.

Applicants should demonstrate a strong interest in the program's content through a motivation letter. Social commitment and extracurricular activities during undergraduate studies, such as university service, clubs, sports, and social work, will be considered. Work experience is not a prerequisite.

Additionally, participants should possess advanced analytical, problem-solving, and verbal communication skills, which must be substantiated through recommendation letters.

Studying at Constructor Institute Schaffhausen involves a highly intercultural environment, so applicants must be willing to join this multicultural-international community and collaborate with students and faculty from various fields of interest.

To be considered for admission, applicants must submit the following documents:

- Letter of motivation
- Curriculum vitae (CV)
- University transcript in English or German
- Bachelor's degree certificate or equivalent in English or German (may be submitted later).
- English language proficiency test results (minimum score of 90 for TOEFL, 6.5 for IELTS, or 110 for Duolingo). Native speakers and applicants who completed their undergraduate studies in English may be exempt from this requirement.
- Copy of passport.
- Optional letter of recommendation.

2 Curriculum

2.1 The curriculum at a glance

The curriculum of the Quantum Software Engineering and Computer Science master's program is divided into four semesters and takes two years to complete. Each semester consists of a combination of core technical content, project/seminar work, management & leadership education, and academic skills work, culminating in a master's thesis that can focus on academic research, industrial applications, or the development of a start-up. The modules are grouped into several domains, as outlined in the Schematic Study Plan (see Figure 1).

To graduate, students must complete a total of 120 ECTS credits from these modules, distributed as follows:

- Technical modules: 45 ECTS credits
- Management modules: 15 ECTS credits
- Leadership / Academic skills modules: 15 ECTS credits
- Capstone project: 15 ECTS credits
- Master's thesis module: 30 ECTS credits

If desired, students can replace 5 ECTS credits of Technical CORE Modules with a research Project module.

2.2 Schematic study scheme

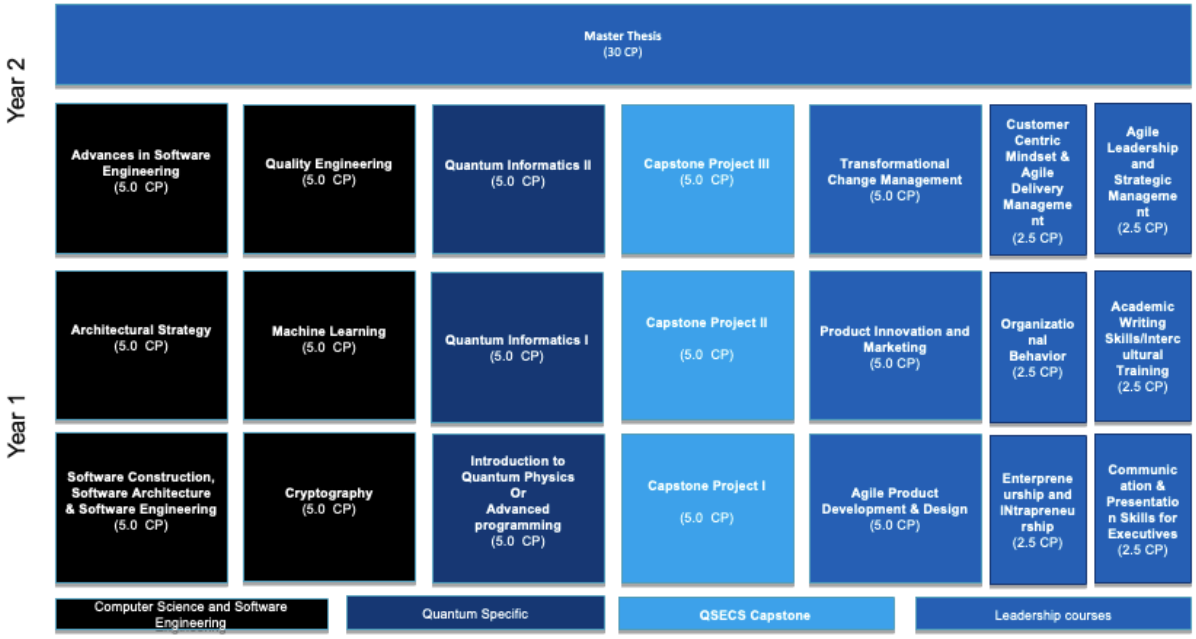


Figure 1: Overview of the Master of Science in Quantum Software Engineering and Computer Science.

2.3 Technical modules

The main subject areas covered in the modules are:

- Software Engineering
- Quantum Technologies

Additionally, there is an area that includes disciplines expected to have a significant impact in the field.

2.3.1 Software engineering modules

The software engineering area covers a wide range of methodological and systematic approaches for developing software and related applications in a professional environment. All three main content modules are mandatory. Students can choose to specialize in this area by taking at least one additional core module. Most modules in this area are taught in person at Schaffhausen.

Software Engineering modules						
Module title	**Module No.**	**Semester**	**Mandatory**	**Coordinator**	**CP**	**Location**
Main content (15 CP mandatory)						
Software Construction, Software Architecture and Software Engineering	XXX	1	Yes	B. Meyer	5	Schaffhausen
Quality Engineering	XXX	1	Yes	N.N.	5	Schaffhausen
Architectural Strategy	XXX	2	Yes	N.N.	5	Schaffhausen
Advances in Software Engineering	XXX	3	Yes	B. Meyer	5	Schaffhausen
Advanced Programming	XXX	1	Yes for physicists	N.N.	5	Schaffhausen
Machine Learning	MDE-CO-04	2	No	NN	5	Bremen or Schaffhausen

2.3.2 Quantum technologies modules

In the cybersecurity specialization, cryptography serves as the entry module into the field. This content is complemented by extended courses on security methods, tools, and technologies at both the system and network levels.

Quantum technologies modules						
Module title	**Module No.**	**Semester**	**Mandatory**	**Coordinator**	**CP**	**Location**
Main Content (5 CP mandatory)						
Introduction to Quantum informatics	XXX	1	Yes for computer scientist	NN	5	Schaffhausen
Cryptography	XXX	1	Yes	NN		
Quantum informatics I	XXX	Tbc	Yes	W. Tittel	5	Geneva
Quantum informatics II	XXX	3	Yes	N.N.	5	Schaffhausen

2.4 Management modules

To equip students with market-relevant management skills, they take modules in the fields of product development, marketing, and change management. All modules are mandatory for the program.

Management modules						
Module title	**Module No.**	**Semester**	**Mandatory**	**Coordinator**	**CP**	**Location**
Agile Product Development & Design	XXX	1	Yes	NN	5	Bremen
Product Innovation & Marketing	XXX	2	Yes	NN	5	Bremen
Transformational Change Management	XXX	3	Yes	NN	5	Bremen

2.5 Leadership / Academic skills modules

Success in industry and research is further strengthened with a set of Leadership and Academic skills modules. All modules listed below must be taken in order to graduate.

Leadership and academic skills modules						
Module title	**Module No.**	**Semester**	**Mandatory**	**Coordinator**	**CP**	**Location**
Entrepreneurship & Intrapreneurship	XXX	1	Yes	NN	2.5	Bremen
Communication & presentation skills for executives	MDE-CAR-01	1	Yes	S. Kettemann	2.5	Bremen
Organizational behavior	XXX	2	Yes	C. Stamov Roßnagel	2.5	Bremen
Academic writing skills / intercultural training	MDE-CAR-02	2	Yes	S. Kettemann	2.5	Bremen
Agile Leadership and Strategic Management	XXX	3	Yes	NN	2.5	Bremen
Customer-centric Mindset and Agile Delivery Management	XXX	3	Yes	NN	2.5	Bremen

2.6 Project, capstone project & master's thesis

To explore the complete development process of a software application in relation to the program's specialization areas, all students undertake three Capstone project modules. It is highly recommended to take these modules in numerical order to gain a full project experience. Students with a strong inclination towards academic research can replace one Technical Module in their third semester with the Research Project, which is carried out in one of the Faculty's research areas.

The master's studies conclude with a 6-month master's thesis, which spans the fourth and final semester.

Capstone project, research project and thesis modules						
Module title	**Module No.**	**Semester**	**Mandatory**	**Coordinator**	**CP**	**Location**
Capstone project 1	XXX	1	Yes	M. Oriol	5	Schaffhausen
Capstone project 2	XXX	2	Yes	M. Oriol	5	Schaffhausen
Capstone project 3	XXX	3	Yes	M. Oriol	5	Schaffhausen
Master's thesis	XXX	4	Yes	B. Meyer	30	Schaffhausen

3 Quantum software engineering and computer science graduate program regulations

3.1 Scope of these regulations

The regulations in this handbook are valid for all students who entered the Quantum Software Engineering and Computer Science graduate program at Constructor Institute in Fall 2023. In case of conflict between the regulations in this handbook and general policies for Master Studies, the latter shall apply.

In exceptional cases, certain necessary deviations from the regulations of this study handbook may occur during the course of study (e.g., changes in the semester sequence, assessment type, or teaching mode of courses).

In general, Constructor Institute Schaffhausen reserves the right to change or modify the regulations of the program handbook at any time and in its sole discretion, even after its publication.

3.2 Degree

Upon successful completion of the study program, students will be awarded a Master of Science (MSc) degree in Quantum Software Engineering and Computer Science.

3.3 Graduation requirements

To graduate, students need to obtain 120 credit points. In addition, the following graduation requirement applies: students must complete all mandatory components of the program as indicated in Chapter 2 of this handbook.

3.4 Other program-specific policies & practices

Close contact and cooperation between program representatives and students are crucial. Therefore, regular meetings are held to continuously evaluate the program, its modules and workshops, supervision, and opportunities. In doing so, the study program chair and involved faculty gain important insights into students' experiences, demands, and overall impressions of the program. At the module component level, students are asked to perform module component evaluations to ensure high-quality modules and provide lecturers with feedback for necessary changes.

The study program chair makes intensive use of this feedback, as well as feedback from industry partners, to improve the learning environment, the program's offerings, and its progress. The current program was shaped through input from previous experiences and discussions with several stakeholders, including students and industry practitioners.

In exceptional cases, certain necessary deviations from the regulations of this study handbook may occur during the course of study (e.g., changes in the semester sequence, assessment type, or teaching mode of courses). Constructor Institute Schaffhausen reserves the right to modify the regulations of the program handbook.

4 Module descriptions

4.1 Core modules: Software Engineering modules

4.1.1 Advanced programming

<table>
<tr><td colspan="2">Module name
Advanced programming</td><td>Module code
tbd</td><td>Level (type)
Year 1</td><td>CP
5</td></tr>
<tr><td colspan="5">Module components
Number | Name | Type | CP
tbd | Advanced programming | Lecture | 5</td></tr>
<tr><td>Module coordinator

N.N.</td><td colspan="2">Program affiliation

• MSc Quantum Software Engineering and Computer Science</td><td colspan="2">Mandatory status

Mandatory for QSECS</td></tr>
<tr><td colspan="3" rowspan="2">Entry requirements

Pre-requisites | Co-requisites | Knowledge, abilities, or skills
☒ none | ☒ none | •</td><td>Frequency

Annually (Spring)</td><td>Forms of learning and teaching

• Lectures (35 hours)
• Private study (90 hours)</td></tr>
<tr><td>Duration</td><td>Workload

125 hours</td></tr>
<tr><td colspan="5">Recommendations for preparation</td></tr>
<tr><td colspan="5">Content and educational aims

The course “Advanced Programming” focuses on expanding the basic notions of programming that physics students studied in their regular curriculum. During the course, we study various classes of standard algorithms and their complexity. The course also covers how programming languages are written, how analyses are run by static analysis tools and how to build such tools.</td></tr>
<tr><td colspan="5">Intended learning outcomes

Upon completion of this module, students will be able to:
1. Manipulate any usual data structures and sorting algorithms
2. Design domain-specific languages
3. Create and run static analyses on code written a reasonable language
4. Know what is a statement, what is a grammar, what is an abstract syntax tree, understand what is a static analysis.</td></tr>
</table>

Indicative literature Big-O cheat cheat Antlr manuals
Usability and relationship to other modules • Useful for quality courses
Examination type: module examination Assessment: Portfolio (individual assignments, group assignments) Weight: 100 % Scope: All intended learning outcomes of the module.

4.1.2 Software Construction, Software Architecture and Software Engineering

<table>
<tr><td colspan="2">Module name
Software Construction, Software Architecture and Software Engineering</td><td>Module code
tbd</td><td>Level (type)
Year 1</td><td>CP
5</td></tr>
<tr><td colspan="5">Module components
Number | Name | Type | CP
tbd | Software Construction, Software Architecture and Software Engineering | Lecture | 5</td></tr>
<tr><td>Module coordinator

Prof. Dr. Bertrand Meyer</td><td colspan="2">Program affiliation

• MSc Quantum Software Engineering and Computer Science</td><td colspan="2">Mandatory status

Mandatory for QSECS</td></tr>
<tr><td colspan="2">Entry requirements

Pre-requisites
☒ none

Co-requisites
☒ none

Knowledge, abilities, or skills
• Some programming experience</td><td>Frequency

Annually (Fall)

Duration

1 semester</td><td colspan="2">Forms of learning and teaching

• Lectures (35 hours)
• Private study (90 hours)

Workload

125 hours</td></tr>
<tr><td colspan="5">Recommendations for preparation</td></tr>
<tr><td colspan="5">Content and educational aims
Software engineering is the body of concepts and techniques that make it possible to construct industrial software systems of high quality. The size, complexity and ambition of systems being developed today requires a systematic approach based on best practices learned over the past decades. Software engineering includes many aspects, both technical (requirements, design, programming, testing and other validation techniques, maintenance) and managerial (project management, metrics, empirical studies, agile methods, lifecycle models, quality assurance). After taking the course, students will understand the issues and challenges of successful software system construction and will be ready to apply them to build high-quality software, including in management roles.

Students will know in the first session which assignments will be part of the portfolio examination.</td></tr>
<tr><td colspan="5">Intended learning outcomes
Upon completion of this module, students will be able to:
1. Use object-oriented techniques to produce high-quality programs.
2. Take advantage of mechanisms of inheritance, genericity and information hiding.
3. Take advantage of Design by Contract techniques to guarantee the reliability of their programs.
4. Apply fundamental design patterns (Observer, Visitor and others).
5. Apply basic techniques of modern software engineering such as configuration management.
6. Apply basic agile development techniques.</td></tr>
<tr><td colspan="5">Indicative literature
Pfleeger, S. and Atlee, J.M. (2010). Software Engineering: Theory and Practice (4th Edition)
Ghezzi, C., Jazayeri, M. and Mandrioli, D (2003). Fundamentals of software engineering (2th Edition), ISBN 978-0-13-305699-0</td></tr>
</table>

Usability and relationship to other modules
Examination type: module examination Assessment: Portfolio (Quizzes, Programming Assignments) Weight: 100 % Scope: All intended learning outcomes of the module.

4.1.3 Quality Engineering

Module name	*Module code*	*Level (type)*	*CP*
Quality Engineering	tbd	Year 1	5

Module components			
Number	*Name*	*Type*	*CP*
tbd	Quality Engineering	Lecture	5

Module coordinator	*Program affiliation*	*Mandatory status*
N.N.	• MSc Quantum Software Engineering and Computer Science	Mandatory for QSECS

Entry requirements			*Frequency*	*Forms of learning and teaching*
Pre-requisites	*Co-requisites*	*Knowledge, abilities, or skills*	Annually (Fall)	• Lectures (35 hours) • Private study (90 hours)
☒ none	☒ none	• Programming skills in an imperative language at CS bachelor level • Algorithms and data structure at CS bachelor level • Basic skills in software testing: structural testing, Junit • Basic knowledge of software engineering and IDEs at CS bachelor level • Discrete math at CS bachelor level	*Duration* 1 semester	*Workload* 125 hours

Recommendations for preparation

Content and educational aims Software quality can be defined as the degree of satisfaction of the requirements; it represents an essential part of the software development and cannot be guaranteed a-priori, but most be verified both during and after the development. This course introduces the main testing and analysis techniques that can be used to identify failures and verify the quality of software systems. The course introduces the general testing and analysis principles and the basic techniques, shows how to apply them to solve relevant quality problems, illustrates complementarities and differences among the different techniques, and presents the organization of a coherent quality process. The course provides the elements needed to understand principles, techniques and process that comprise the basic background of test designer, quality manager and project manager. At the end of the course, the students will be able to define and implement quality plans for complex software systems. The student will have the basic knowledge of a project and a quality manager. Students will know in the first session which assignments will be part of the portfolio examination.
Intended learning outcomes Upon completion of this module, students will be able to: 1. Manage a software quality process. 2. Select and implement a suitable set of testing and analysis activities to certify the quality of software systems. 3. Understand the core principles of software testing and program analysis. 4. Master the basic techniques underlying software testing and program analysis. 5. Choose the suitable approaches to address the different testing and analysis programs. 6. Design and monitor a suitable quality process.
Indicative literature
Usability and relationship to other modules
Examination type: module examination Assessment: Portfolio (individual assignments, group assignments) Weight: 100 % Scope: All intended learning outcomes of the module. Bonus achievement: Additional bonus homework as a voluntary task can improve the grade, but is not required to reach the best grade in the module (1.0).

4.1.4 Architectural strategy

Module name		*Module code*	*Level (type)*	*CP*
Architectural strategy		tbd	Year 1	5
Module components				
Number	*Name*		*Type*	*CP*
tbd	Architectural Strategy		Lecture	5
Module coordinator	*Program affiliation*		*Mandatory status* Mandatory for QSECS	

N.N.	• MSc Quantum Software Engineering and Computer Science	

Entry requirements			*Frequency*	*Forms of learning and teaching*
Pre-requisites	*Co-requisites*	*Knowledge, abilities, or skills*	Annually (Spring)	• Lectures (35 hours) • Private study (70 hours)
☒ none	☒ none	•		
			Duration	*Workload*
				125 hours

Recommendations for preparation

Content and educational aims

The course "Architectural Strategy" focuses on Software Architectures, the key element for systematically developing large and complex software systems. During the course, we study how to design, recover, analyze, and document Software Architectures and understand how the main design decisions comprising them influence the quality attributes of the resulting systems.

Students will know in the first session which assignments will be part of the portfolio examination.

Intended learning outcomes

Upon completion of this module, students will be able to:

5. Understand methods for designing large software systems
6. Design complex and large software systems using components and connectors
7. Use UML as modeling language to represent the main concepts of software systems
8. Document their main design decisions and motivate them in terms of quality attributes

Indicative literature

R.N. Taylor, N. Medvidovic, E.M. Dashofy, Software Architecture: Foundations, Theory, and Practice, Wiley, January (2009)

Len Bass, Paul Clements, Rick Kazman: Software Architecture in Practice. Addison Wesley 2013

C. Pautasso, Software Architecture, 2020 (Visual Lecture Notes)

Usability and relationship to other modules

•

Examination type: module examination

Assessment: Portfolio (individual assignments, group assignments) Weight: 100 %

Scope: All intended learning outcomes of the module.

Bonus achievement: Additional bonus homework as a voluntary task can improve the grade, but is not required to reach the best grade in the module (1.0).

4.1.5 Advances in Software Engineering

<table>
<tr><td colspan="3">Module name
Advances in Software Engineering</td><td>Module code
tbd</td><td>Level (type)
Year 2</td><td>CP
5</td></tr>
<tr><td colspan="6">Module components
Number | Name | Type | CP
tbd | Advances in Software Engineering | Lecture | 2.5
tbd | Advances in Software Engineering – Lab | Lab | 2.5</td></tr>
<tr><td>Module coordinator

Prof. Dr. Bertrand Meyer</td><td colspan="3">Program affiliation

• MSc Quantum Software Engineering and Computer Science</td><td colspan="2">Mandatory status

Mandatory for QSECS</td></tr>
<tr><td colspan="3">Entry requirements

Pre-requisites

☒ Software Construction, Software Architecture and Software Engineering

Co-requisites

☒ None

Knowledge, abilities, or skills

• Familiarity with basics of software engineering and software architecture
• Programming experience</td><td>Frequency

Annually (Spring)

Duration

1 semester</td><td colspan="2">Forms of learning and teaching

• Lectures (17.5 hours)
• Lab (17.5 hours)
• Private Study (90 hours)

Workload

125 hours</td></tr>
<tr><td colspan="6">Recommendations for preparation</td></tr>
<tr><td colspan="6">Content and educational aims

The course covers topics of modern software engineering beyond the basic concepts covered in the first semester SCAE course (Software Construction, Architecture and Engineering). After taking it, the students will master important techniques for high-quality software development and management, particularly in three areas: requirements engineering; formal methods and software verification; project management and agile methods.</td></tr>
<tr><td colspan="6">Intended learning outcomes

1. Apply techniques of formal software verification, particularly axiomatic semantics, to proving program correctness.
2. Use a program-proving framework.
3. Perform effective requirements.
4. Apply requirements techniques such as use cases and object-oriented requirements.
5. Use agile development techniques to manage a project.
6. Make the difference between productive and harmful agile ideas.
7. Combine agile methods with process models such as CMMI.</td></tr>
<tr><td colspan="6">Indicative literature

Bertrand Meyer, Handbook of Requirements Engineering and Business Analysis, Springer, 2022
Flemming Nielson, Hanne Riis Nielson, Chris Hankin: Principles of Program Analysis, Springer, most recent edition</td></tr>
</table>

Bertrand Meyer, Agile! The Good, the Hype and the Ugly, Springer. 2014
Usability and relationship to other modules
Examination type: Module component examinations (tbc) **Module component 1: Lecture** Assessment type: Written examination Duration/length: 90 min Weight: 50% Scope: All intended learning outcomes of this module. **Module component 2: Lab** Assessment type: Requirements project Weight: 50 % Scope: All intended learning outcomes of this module. Completion: To pass this module, the examination of each module component has to be passed with at least 45%

4.1.6 Quantum technologies modules

4.1.7 Cryptography

<table>
<tr><td colspan="2">Module name
Cryptography</td><td>Module code
tbd</td><td>Level (type)
Year 1</td><td>CP
5</td></tr>
<tr><td colspan="5">Module components
Number | Name | Type | CP
tbd | Cryptography | Lecture | 5</td></tr>
<tr><td>Module coordinator
Prof. Dr. Jürgen Schönwälder</td><td colspan="2">Program affiliation
• MSc Computer Science and Software Engineering
• MSc Quantum Software Engineering and Computer Science</td><td colspan="2">Mandatory status
Mandatory for QSECS</td></tr>
<tr><td colspan="2">Entry requirements
Pre-requisites
☒ none

Co-requisite
☒ none

Knowledge, abilities, or skills</td><td>Frequency
Annually (Fall)</td><td colspan="2">Forms of learning and teaching
• Lectures (35 hours)
• Private study (70 hours)
• Exam preparation (20 hours)</td></tr>
<tr><td colspan="2"></td><td>Duration
1 semester</td><td colspan="2">Workload
125 hours</td></tr>
<tr><td colspan="5">Recommendations for preparation

Students are expected to have a solid mathematical foundation. Students should review basic concepts of number theory, probability theory, and complexity theory as preparation for this module.</td></tr>
<tr><td colspan="5">Content and educational aims

Information security requires techniques to protect information and to secure communication. Cryptography studies the design of cryptographic algorithms that can ensure the confidentiality, the integrity, and the authenticity of data and messages exchanged in a secure communication protocol. This module focuses on the mathematical and algorithmic foundations of cryptography, and it covers the application of basic primitives to solve common information security challenges. Students familiar with the foundations of cryptographic algorithms will be able to judge the applicability and limitations of different cryptographic algorithms.</td></tr>
<tr><td colspan="5">Intended learning outcomes

Upon completion of this module, students will be able to:
1. Understand the mathematical problems on which cryptographic algorithms are built
2. Describe pseudo random number generators and pseudo random functions
3. Evaluate the strengths, weaknesses, and the applicability of cryptographic algorithms
4. Select from a set of symmetric block cipher, message integrity, and authenticated encryption algorithms
5. Contrast different asymmetric ciphers (finite field based, elliptic curve based, lattice based, hash based)
6. Explain the notion of quantum resistant cryptographic algorithms
7. Analyze the properties of cryptographic protocols such as key exchange mechanisms
8. Apply techniques to analyze cryptographic protocols and their implementations
9. Explain homomorphic encryption schemes and differential privacy</td></tr>
<tr><td colspan="5">Indicative literature

▪ Bruce Schneier: Applied Cryptography, 20th Anniversary Edition, Wiley, 2015</td></tr>
</table>

- Wm.Arthur Conklin, Gregory White: Principles of Computer Security, 5th Edition, McGraw-Hill, 2018
- Simon Singh: The Code Book: Science of Secrecy from Ancient Egypt to Quantum Cryptography, Anchor Books, 2000
- Dan Boneh, Victor Shoup: A Graduate Course in Applied Cryptography, version 0.5, online, 2020

Usability and relationship to other modules

- The module serves as the foundational module in the cyber security specialization. Other modules related to cyber security build on this module.

Examination type: module examination

Assessment: Written examination — Duration: 120 min
Weight: 100%

Scope: All intended learning outcomes of the module.

4.1.8 Machine Learning

<table>
<tr><td colspan="3">Module name
Machine Learning</td><td>Module code
MDE-CO-04</td><td>Level (type)
Year 1</td><td>CP
5</td></tr>
<tr><td colspan="6">Module components
Number | Name | Type | CP
MDE-CO-04 | Machine Learning | Lecture | 5</td></tr>
<tr><td>Module coordinator
NN</td><td colspan="3">Program affiliation
▪ MSc Data Engineering
▪ MSc Quantum Software Engineering and Computer Science</td><td colspan="2">Mandatory status
Mandatory for QSECS</td></tr>
<tr><td colspan="3">Entry requirements
Pre-requisites
☒ None
Co-requisites
☒ None
Knowledge, abilities, or skills
▪ Basic linear algebra, calculus and probability theory, as typically acquired in entry modules in BSc studies</td><td>Frequency
Annually (Spring)

Duration
1 semester</td><td colspan="2">Forms of learning and teaching
▪ Lectures (35 hours)
▪ Private Study, incl. exercises and exam preparation (90 hours)

Workload
125 hours</td></tr>
<tr><td colspan="6">Recommendations for preparation

Read the syllabus.
Highly recommended: Mitchell, Tom M.: Machine Learning (McGraw-Hill, 1997) IRC: Q325.5.M58 1997. This standard, classical textbook gives a very accessible overview of ML.</td></tr>
<tr><td colspan="6">Content and educational aims

Machine learning (ML) is a module that concerns algorithms that are fed with (large quantities of) real-world data, and which return a compressed "model" of the data. An example is the "world model" of a robot: the input data are sensor data streams, from which the robot learns a model of its environment. Another example is a spoken language model: the input data are speech recordings, from which ML methods build a model of spoken English -- useful, for instance, in automated speech recognition systems. There are many formalisms in which such models can be cast, and an equally large diversity of learning algorithms. At the same time, there is a relatively small number of fundamental challenges that are common to all of these formalisms and algorithms.

The module introduces such fundamental concepts and illustrates them with a choice of elementary model formalisms (linear classifiers and regressors, radial basis function networks, clustering, neural networks). Furthermore, the module also (re)introduces required mathematical material from probability theory and linear algebra. The main educational aims are twofold: to make students fully aware of the two main hurdles for obtaining good models from data: (i) the "curse of dimensionality" and (ii) the bias-variance dilemma and to provide standard tools to cope with these difficulties, namely (i') dimension reduction by feature extraction, for example via PCA or clustering, and (ii') cross-validation and regularization.</td></tr>
<tr><td colspan="6">Intended learning outcomes

Upon completion of this module, students will be able to:</td></tr>
</table>

1. design, implement and exploit elementary supervised ML methods for classification and regression with expert care given to dimension reduction preprocessing and regularization; 2. understand and practically use PCA and linear regression; 3. understand the core ideas behind feedforward neural networks and the backpropagation algorithm, as the basis for accessing "deep learning" methods.
Indicative literature T. M. Mitchel, Machine Learning, McGraw-Hill, 1997, IRC: Q325.5.M58.
Usability and relationship to other modules This module together with the module “Data Analytics” are favorable companion modules for students with a focus on Software Engineering or Cybersecurity that still want to gain knowledge in these relevant areas. “Deep Learning” targets a deeper understanding of the related field.
Examination Type: Module Examination Assessment Type: Written Exam Duration: 120 minutes Weight: 100% Scope: All intended learning outcomes of this module.

4.1.9 Quantum informatics I

Module name		***Module code***	***Level (type)***	***CP***
Quantum informatics I		tbd	Year 2	5
Module components				
Number	*Name*		*Type*	*CP*
tbd	Quantum informatics		Lecture	2.5
tbd	Quantum informatics Lab		Lab	2.5
Module coordinators Prof. Dr. Wolfgang Tittel	***Program affiliation*** • MSc Quantum Software Engineering and Computer Science		***Mandatory status*** Mandatory for QSECS	
Entry requirements *Pre-requisites* ☒ none	*Co-requisites* ☒ none	*Knowledge, abilities, or skills* Basic linear algebra	***Frequency*** Annually	***Forms of learning and teaching*** Lectures (17.5 hours) Lab/precepts (17.5 hours) Private study incl. exercises, projects, and exam preparation (90 hours)
			Duration	***Workload***

	1 semester	125 hours

Recommendations for preparation
Introductory texts on quantum mechanics, quantum information and quantum computing; review of vectors and matrices

Content and educational aims

This module features a self-contained introduction to Quantum informatics, one of the fastest growing emergent fields in science and technology, including essential elements from physics and mathematics. Topics include an overview of current quantum technology; pertinent aspects of quantum mechanics and information theory; qubits, quantum registers, quantum gates; no-cloning theorem, deferred and implicit quantum measurement; circuit model of quantum computing; quantum communication, cryptography and attacks; Grover, Shor and further quantum algorithms; post-quantum cryptography; decoherence, quantum channels, quantum error correction; physical qubits; variational and adiabatic quantum computing, quantum annealing; quantum simulation; quantum programming and quantum SDKs.

The lectures are complemented by a lab, where concepts are further deepened and practically applied. Part of the lab will be in precept-style with exercises, part will involve hands-on practical experience including mini projects.

Intended learning outcomes

Upon completion of this module, students will be able to:

1. Discuss the state of the art of quantum computing and quantum communication.
2. Apply the principles of quantum theory to analyze quantum circuits.
3. Develop quantum algorithms and quantum communication protocols.
4. Assess applications of quantum informatics.

Indicative literature

Michael A. Nielsen, Isaac L. Chuang: Quantum Computation and Quantum Information (10th Anniversary Edition), Cambridge University Press, 2010

N. David Mermin: Quantum Computer Science: An Introduction, Cambridge University Press, 2007

Usability and relationship to other modules

Module component examinations

Module component 1: Final exam
Assessment Type: Written examination Duration/length: 120 min
Weight: 50%
Scope: all ILOs (focus on theory).

Module component 2: Lab assessment
Assessment Type: Portfolio (Graded Exercises, Project Work)
Weight: 50%
Scope: all ILOs (focus on practical application).

4.1.10 Quantum informatics II

Module name	*Module code*	*Level (type)*	*CP*

Quantum informatics II	tbd	Year 2	5

Module components

Number	*Name*	*Type*	*CP*
tbd	Quantum informatics	Lecture	2.5
tbd	Quantum informatics Lab	Lab	2.5

Module coordinators	***Program affiliation***	***Mandatory status***
NN	• MSc Computer Science & Software Engineering • MSc Quantum Software Engineering and Computer Science	Mandatory for QSECS

Entry requirements

Pre-requisites	*Co-requisites*	*Knowledge, abilities, or skills*
☒ none	☒ none	Basic linear algebra

Frequency	***Forms of learning and teaching***
Annually	Lectures (17.5 hours) Lab/precepts (17.5 hours) Private study incl. exercises, projects, and exam preparation (90 hours)
Duration	***Workload***
1 semester	125 hours

Recommendations for preparation

Introductory texts on quantum mechanics, quantum information and quantum computing; review of vectors and matrices

Content and educational aims

This module goes further in quantum technologies. In particular it is a mix of research paper reading, seminars and very advanced concepts in Quantum technologies.

Intended learning outcomes

Upon completion of this module, students will be able to:

- read research articles on quantum technology
- come up with new ideas around software engineering and quantum technologies
- write the result of the research on quantum technology

Indicative literature

Michael A. Nielsen, Isaac L. Chuang: Quantum Computation and Quantum Information (10th Anniversary Edition), Cambridge University Press, 2010

N. David Mermin: Quantum Computer Science: An Introduction, Cambridge University Press, 2007
Usability and relationship to other modules This is a continuation of the quantum informatics I module.
Module component examinations **Module component 1: Final Exam** Assessment Type: Written examination Duration/length: 120 min Weight: 50% Scope: all ILOs (focus on theory). **Module component 2: Lab Assessment** Assessment Type: Portfolio (Graded Exercises, Project Work) Weight: 50% Scope: all ILOs (focus on practical application).

4.2 Management modules

4.2.1 Agile product development & design

<table>
<tr><td colspan="3">Module name
Agile product development & design</td><td>Module code
tbd</td><td>Level (type)
Year 1</td><td>CP
5</td></tr>
<tr><td colspan="6">Module components
Number | Name | Type | CP
tbd | Agile product development & design | Lecture | 5</td></tr>
<tr><td>Module coordinator
NN</td><td colspan="3">Program affiliation
▪ MSc Quantum Software Engineering and Computer Science</td><td colspan="2">Mandatory status
Mandatory for QSECS</td></tr>
<tr><td colspan="3">Entry requirements
Pre-requisites
▪ none
Co-requisites
☒ None
Knowledge, abilities, or skills</td><td>Frequency
Annually (Fall)</td><td colspan="2">Forms of learning and teaching
▪ Lecture (80 hours)
▪ Private study (45 hours)</td></tr>
<tr><td colspan="3"></td><td>Duration
1 semester</td><td colspan="2">Workload
125 hours</td></tr>
<tr><td colspan="6">Recommendations for preparation

N.A.</td></tr>
<tr><td colspan="6">Content and educational aims

This course is focused on key aspects of agile product and service development and design process.
State-of-the-art user centered design methods will be at the core of the course.

The overall goal of this module is to help managers without a business degree to learn, understand and practice agile customer- and data-driven innovation processes in the information age. This module helps students to understand today’s real-life challenges in a complex world, with wicked problems and with multiple stakeholder interests, where unpredictable is common, and where managers need to focus on achieving goals rather than repetitive tasks.
Students learn to develop and present innovative user-centered and theory-oriented solutions for real-world challenges in an IT-driven world.
This course is strongly based on the agile paradigm of user-centeredness, user-centered design and the ideas of the Service Dominant Logic. Service-dominant (S-D) logic is a meta-theoretical framework for explaining value co-creation, through exchange, among configurations of actors.
Major challenges and concerns will be reflected:
● the role of the customer and data in a transformed business world
● new theories, concepts, and approaches (such as service dominant logic, customer integration, gamification, new service models)
● new methods and management techniques in (service) innovation (Design Thinking)
● new methods in handling business processes: (agile) business process management - BPM
● ethics and security issues.

The module will enable students to collaborate across disciplines with experts from various areas.</td></tr>
<tr><td colspan="6">Intended learning outcomes</td></tr>
</table>

Upon completion of this module, students will be able to:

1. Develop practical knowledge and management skills, and mind sets to master the challenges from an agile business environment
2. Understand (routine) business processes in various context and how to adapt business processes to an agile business environment (agile Business Process Management)
3. Summarize and classify the new data- and customer-driven technologies in a business context
4. Understand the ideas of the "service dominant logic" as a business opportunity, such as user-centricity, value in use, value in interaction, business service ecosystems.
5. Apply innovative creativity methods and processes for product and software development (Design Thinking)
6. Adapt to a new working culture based on a user-centricity, empathy, and playful testing of new products and services.

Indicative literature

Service Dominant Logic

Vargo, S.L., & Lusch, R. (2004). Evolving to a New Dominant Logic for Marketing. Journal of Marketing, Vol. 68(1), 1 – 17

Vargo SL, Akaka MA, Vaughan CM. (2017). Conceptualizing Value: A Service-ecosystem View. Journal of Creating Value. 3(2):117-124. https://doi.org/10.1177%2F2394964317732861

Lusch, R.F., Nambisan, S. (2015). Service Innovation: A Service-Dominant Logic Perspective. MIS Quarterly. Vol. 39 No.1 , pp. 155-175. https://doi.org/10.25300/MISQ/2015/39.1.07

Business Process Management and agile Management

Daniel Paschek, D., Frank Rennung, F., Trusculescu, A., Draghici,A. (2016). Corporate Development with Agile Business Process Modeling as a Key Success Factor, Procedia Computer Science, Vol 100, Pages 1168-1175, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2016.09.273.

Design Thinking

Brenner, W., Uebernickel, F., Abrell, T. (2016). Design Thinking as Mindset, Process, and Toolbox, in: Brenner, W., Uebernickel, F. (Eds.), Design Thinking for Innovation. Springer International Publishing, pp. 3–21. https://doi.org/10.1007/978-3-319-26100-3_1

Brown, T. (2008). Design Thinking. Harvard Business Review. 86, 84–92. Available at: https://hbr.org/2008/06/design-thinking

Usability and relationship to other modules

Examination type: module examination

Assessment Type: Presentation Duration: 30 min
Weight: 100%

Scope: All intended learning outcomes.

4.2.2 Product Innovation & Marketing

Module name	*Module code*	*Level (type)*	*CP*
Product Innovation & Marketing	tbd	Year 1	5

Module components			
Number	*Name*	*Type*	*CP*
tbd	Product Innovation & Marketing	Lecture	5

Module coordinator	*Program affiliation*	*Mandatory status*
Prof. Dr. Tilo Halaszovich	▪ MSc Quantum Software Engineering and Computer Science	Mandatory for QSECS

Entry requirements			*Frequency*	*Forms of learning and teaching*
Pre-requisites	*Co-requisites*	*Knowledge, abilities, or skills*	Annually (Spring)	▪ Lecture (80 hours) ▪ Private study (45 hours)
▪ none	☒ None		*Duration* 1 semester	*Workload* 125 hours

Recommendations for preparation

N.A.

Content and educational aims

This course focuses on key strategic aspects of the innovation and commercialization process. The course draws on insights from a variety of fields – in particular, product management, innovation, marketing, and strategic management – in order to (i) develop a holistic, state-of-the art understanding of this process, (ii) to nurture the underlying mindset that spans technology and market elements, and (iii) to provide students with concrete tools that help them in navigating the journey from product idea to market success. The course will take both the perspective of established companies as well as of new ventures.

Intended learning outcomes

Upon completion of this module, students will be able to:

1. understand the innovation process, particularly in technology domains
2. understand the commercialization process, particularly in technology domains
3. analyze how value can be created and appropriated through innovation
4. understand and apply tools, methods and concepts to manage the commercialization process

Indicative literature

Gruber/Tal (2017). Where to Play: 3 Steps for Identifying your Most Valuable Market Opportunities, Financial Times/Pearson.
Mohr, J. et al. (2013). Marketing of high-technology products and innovations. Pearson Education.
Moore, G. A. (2014). Crossing the chasm. Harper Business.
Schilling, M.A. (2019). Strategic Management of Technological Innovation. McGraw-Hill.

Usability and relationship to other modules

Examination type: module examination

Assessment Type: Presentation Duration: 30 min

Weight: 100% Scope: All intended learning outcomes.

4.2.3 Transformational change management

<table>
<tr><td colspan="2">Module name
Transformational change management</td><td>Module code
tbd</td><td>Level (type)
Year 2</td><td>CP
5</td></tr>
<tr><td colspan="5">Module components
Number | Name | Type | CP
tbd | Transformational Change Management | Lecture | 5</td></tr>
<tr><td>Module coordinator
NN</td><td colspan="2">Program affiliation
▪ MSc Quantum Software Engineering and Computer Science</td><td colspan="2">Mandatory status
Mandatory for QSECS</td></tr>
<tr><td colspan="2">Entry requirements
Pre-requisites
▪ none
Co-requisites
☒ None
Knowledge, abilities, or skills</td><td>Frequency
Annually (Fall)

Duration
1 semester</td><td colspan="2">Forms of learning and teaching
▪ Lecture (80 hours)
▪ Private study (45 hours)

Workload
125 hours</td></tr>
<tr><td colspan="5">Recommendations for preparation

N.A.</td></tr>
<tr><td colspan="5">Content and educational aims

Change is part of every successful manager’s and organization’s life. Thus, learning to lead change and/or be part of a successful change effort, is essential for anyone who hopes to rise from being an individual contributor. Some change efforts have no impact whatsoever; the organization is neither better nor worse afterwards. This is a waste of human capital (and probably financial capital as well). Some change efforts work for a while, but then gravity takes over and the organization returns to where it was beforehand; again, a waste. And there are other change projects that get us to a new level, and we stay there, which is not bad; a vast improvement on the previous two situations. But what we all want, and what this course will focus on, is to change an organization in some way, and put it on a continuous upward trajectory. That is transformation. To build this understanding, the courses deals with the following topics:
● Change management models
● Influencing styles and tactics
● Communicating well in a group
● Understanding your biases
● Seeing and understanding different leadership styles in company transformations
● Stakeholder management</td></tr>
<tr><td colspan="5">Intended learning outcomes

Upon completion of this module, students will be able to:

1. Understand, evaluate, and apply different leadership styles
2. Understand and evaluate the change process in organizations
3. Understand and apply communications and influencing
4. Evaluate their role in a change situation
5. Assess the stakeholders in any change context
6. Lead or be part of an organizational change effort</td></tr>
</table>

Indicative literature Daniel Goleman, HBR, 2002, Leadership that gets results.
Usability and relationship to other modules
Examination type: module examination Assessment Type: Presentation Duration: 30 min Weight: 100% Scope: All intended learning outcomes.

4.3 Leadership / Academic skills modules

4.3.1 Entrepreneurship and intrapreneurship

<table>
<tr><td colspan="2">Module name
Entrepreneurship and intrapreneurship</td><td>Module code
tbd</td><td>Level (type)
Year 1</td><td>CP
2.5</td></tr>
<tr><td colspan="5">Module components
Number | Name | Type | CP
tbd | Entrepreneurship and intrapreneurship | Lecture | 2.5</td></tr>
<tr><td>Module coordinator
NN</td><td colspan="2">Program affiliation
• MSc Quantum Software Engineering and Computer Science</td><td colspan="2">Mandatory status
Mandatory for QSECS</td></tr>
<tr><td colspan="2">Entry requirements
Pre-requisites
▪ none
Co-requisites
☒ None
Knowledge, abilities, or skills</td><td>Frequency
Annually (Fall)
Duration
1 semester</td><td colspan="2">Forms of learning and teaching
▪ Lecture (17.5 hours)
▪ Private study (45 hours)
Workload
62.5 hours</td></tr>
<tr><td colspan="5">Recommendations for preparation

N.A.</td></tr>
<tr><td colspan="5">Content and educational aims

The module introduces students to the themes which are relevant to clearly develop corporate innovation and entrepreneurship as an activity. It introduces entrepreneurial thinking styles that are important to develop radical forms of innovation in companies. This is about a way of thinking, reasoning and acting that is opportunity obsessed and holistic in approach. It is first and foremost a process that has an intention to create, enhance, realize, and renew value, not just for owners, but for all participants and stakeholders in either a new or existing organization. Today, entrepreneurship has evolved beyond the classic start-up notion to include companies and organizations of all types, old and new; small and large; fast and slow growing; private, not-for-profit, and public.
This focus on “entrepreneurship as a process” has become a fundamental part for three main reasons. The first is the growing recognition of the critical importance of entrepreneurial activities in the economy and the society at large. As such, having an insight in the specific challenges and solutions that characterize entrepreneurship has broader implications for any 21st century graduate. The second reason is that many graduates eventually find themselves occupying a position as entrepreneur, or are associated with one as their financier, partner, supplier or customer. This requires an action-oriented approach and approaching the phenomenon from multiple angles. Finally, given the specific challenges entrepreneurs often face in terms of uncertainty and resource scarcity, solutions applied by expert entrepreneurs can be of value to any professional that finds him/herself in similar situations in organizations seeking growth, renewal or even survival.
The module focuses on the tasks and skills that entrepreneurs typically complete/use in their journey towards success. With this in mind, this module aims to provide students with insight into the approach entrepreneurs use to identify opportunities and build new ventures; the analytical skills that are needed to implement this approach; and the background knowledge and managerial skills that are needed for dealing with issues involved in starting, growing, and harnessing the value of new ventures. First and foremost, however, entrepreneurship is about action. Hence our approach is based on the primary objective of having students experience entrepreneurship.</td></tr>
</table>

The module assessment will consist of three presentations. Students will know in the first session which topics need to be covered in their presentations.

Intended learning outcomes

Upon completion of this module, students will be able to:

1. Understand the essence of entrepreneurship
2. Assess and develop a business case
3. Analyse and identify new venture opportunities in a more systematic way
4. Understand the importance of a business model for new venture creation
5. Evaluate the viability of a new venture idea
6. Understand how to finance a new venture
7. Create and present a business case for a new venture

Indicative literature

Clarysse, B., Kiefer, S. The Smart Entrepreneur. Elliott & Thompson, 2011.

Usability and relationship to other modules

Examination type: module examination

Assessment type: Presentations Duration: 30 min
Weight: 100%

Scope: All intended learning outcomes.

4.3.2 Communication & presentation skills for executives

<table>
<tr><td colspan="3">Module name
Communication & Presentation Skills for Executives</td><td>Module code
MDE-CAR-01</td><td>Level (type)
Year 1</td><td>CP
2.5</td></tr>
<tr><td colspan="6">Module components
Number | Name | Type | CP
MDE-CAR-01 | Communication & presentation skills for executives | Seminar | 2.5</td></tr>
<tr><td>Module coordinator
Prof. Dr. Stefan Kettemann</td><td colspan="3">Program affiliation
▪ MSc Quantum Software Engineering and Computer Science</td><td colspan="2">Mandatory status
Mandatory for QSECS</td></tr>
<tr><td colspan="2">Entry requirements
Pre-requisites
☒ None</td><td>Co-requisites
☒ None</td><td>Knowledge, abilities, or skills
▪ Analysis, Basic Calculus, and Linear Algebra</td><td>Frequency
Annually (Fall)

Duration
1 semester</td><td>Forms of learning and teaching
▪ Seminar (17.5 hours)
▪ Private study (45 hours)

Workload
62.5 hours</td></tr>
<tr><td colspan="6">Recommendations for preparation

Read the Syllabus</td></tr>
<tr><td colspan="6">Content and educational aims

An executive career in an international business environment requires excellent communication and presentation skills. Managers have to communicate effectively with a large variety of target audiences, often in different languages and with different cultural backgrounds. This is true for employees and/or direct reports, business partners as well as customers. The ability to present and communicate succinctly and confidently while being culturally aware and building rapport and trust with different audiences is crucial. In this interactive module, students are introduced to the basics of effective presentation and communication techniques. They learn how to present themselves, their business project, or academic work, with impact, tailoring both the content and their delivery style to different types of audiences.</td></tr>
<tr><td colspan="6">Intended learning outcomes

Upon completion of the module, students will be able to

1. act as effective communicators – in both group and individual situations;
2. understand interpersonal communication models and group dynamics in presentations;
3. understand the importance of building rapport and trust with audiences;
4. use presentation software (PowerPoint, Prezi) confidently and in a visually pleasant way;
5. learn how to structure presentations in a coherent manner and develop captivating narratives;
6. work with different presentation formats (Ignite, Pecha Kucha, Pitching etc.);
7. understand and apply the basics of logical reasoning in oratory (deductive/inductive);
8. develop oratory and rhetorical skills drawing on Aristotle's teaching of logos, ethos and pathos;
9. understand and apply the basics of interpersonal communication (Johari Window, 4-Ears model etc.);
10. present themselves in different business situations;
11. collaborate effective in intercultural teams.</td></tr>
</table>

Indicative literature
Usability and relationship to other modules
Examination type: module examination Assessment Type: Oral Presentation — Duration: 15 minutes Weight: 100% Scope: All intended learning outcomes of this module.

4.3.3 Organizational behavior

<table>
<tr><td colspan="2">Module name
Organizational behavior</td><td>Module code
tbd</td><td>Level (type)
Year 1</td><td>CP
2.5</td></tr>
<tr><td colspan="5">Module components
Number | Name | Type | CP
tbd | Organizational behavior | Lecture | 2.5</td></tr>
<tr><td>Module coordinator

Prof. Dr. Christian Stamov Roßnagel</td><td colspan="2">Program affiliation

▪ MSc Quantum Software Engineering and Computer Science</td><td colspan="2">Mandatory status

Mandatory for QSECS</td></tr>
<tr><td colspan="2">Entry requirements

Pre-requisites
▪ none

Co-requisites
☒ None

Knowledge, abilities, or skills</td><td>Frequency

Annually (Spring)

Duration

1 semester</td><td colspan="2">Forms of learning and teaching

▪ Lecture (17.5 hours)
▪ Private study (45 hours)

Workload

62.5 hours</td></tr>
<tr><td colspan="5">Recommendations for preparation

N.A.</td></tr>
<tr><td colspan="5">Content and educational aims

Geared towards improving an organization’s effectiveness, organizational behavior (OB) focuses on the impact of people, groups, and organizational structures on work-related behavior within organizations. OB research findings help align personal and organizational needs in selecting, placing, and developing people in organizations. In the face of the current ‘3D’ megatrends of digitalization, diversity, and demographic change, companies’ demand for OB solutions is greater than ever. For a thorough understanding of the principles governing OB, you will build a generic model of the multilevel interactions between parameters on the individual, group, and organizational levels, and how those relate to individual and organizational productivity. From this comprehensive model, you will derive actionable guidelines for personnel selection, performance management, and leadership and apply them to addressing leadership and management challenges in selected business case examples. This module is intended to help you acquire the background to analyses and structure organizations in an evidence-based 21st -century manner.

The module assessment will consist of three presentations. Students will know in the first session which topics need to be covered in their presentations.</td></tr>
<tr><td colspan="5">Intended learning outcomes

Upon completion of this module, you will be able to:
1. Explain basic principles of individuals’ and groups’ behaviours in organisations
2. Apply established theories to assessing and predicting behaviour
3. Describe core techniques of influencing and modifying behaviour
4. Critically discuss selected approaches to effectively lead employees, teams, and groups</td></tr>
<tr><td colspan="5">Indicative literature

King, D., & Lawley, S. (2019). Organizational Behaviour (3rd ed.). Oxford University Press.</td></tr>
</table>

Usability and relationship to other modules
Examination type: module examination Assessment type: Presentations Duration: 30 min Weight: 100% Scope: All intended learning outcomes. Completion: To pass this module, the examination of each module component has to be passed with at least 45%.

4.3.4 Academic writing skills / intercultural training

<table>
<tr><td colspan="3">Module name
Academic writing skills/Intercultural training</td><td>Module code
MDE-CAR-02</td><td>Level (type)
Year 1</td><td>CP
2.5</td></tr>
<tr><td colspan="6">Module components
Number | Name | Type | CP
MDE-CAR-02 | Academic Writing Skills/Intercultural Training | Seminar | 2.5</td></tr>
<tr><td>Module coordinator
Prof. Dr. Stefan Kettemann</td><td colspan="3">Program affiliation
• MSc Quantum Software Engineering and Computer Science</td><td colspan="2">MandatorysStatus
Mandatory for QSECS</td></tr>
<tr><td colspan="3" rowspan="2">Entry requirements
Pre-requisites
☒ None
Co-requisites
☒ None
Knowledge, abilities, or skills
☒ None</td><td>Frequency
Annually (Spring)</td><td colspan="2">Forms of learning and teaching
• Lectures (17.5 hours)
• Private Study (45 hours)</td></tr>
<tr><td>Duration
1 semester</td><td colspan="2">Workload
62.5 hours</td></tr>
<tr><td colspan="6">Recommendations for preparation

Read the Syllabus.
Fraedrich, J. & Ferrell, O.C. (2014): Business Ethics: Ethical Decision Making & Cases. Cengage Learning.</td></tr>
<tr><td colspan="6">Content and educational aims

The academically rigorous nature of graduate studies requires students to master academic writing skills and techniques. In this introductory course, students in DE master's program will learn the foundations of academic writing at a graduate level, with special focus on writing academic essays, identifying organizational patterns of academic texts, and formulating arguments to produce cohesive and coherent academic papers. Through the process of drafting, continuous feedback and editing, students will improve their writing skills. This course will also help students develop their research skills by highlighting techniques of finding and evaluating sources, and utilizing citation and referencing styles. As graduate students, adhering to The Code of Academic Integrity is a requirement. Hence, this course will incorporate a session on scholarly and intellectual standards set by Constructor. The second part of this course is a training seminar. It will give answers to frequently asked questions by students on the topics of working and living in Germany or Switzerland. Here the students will find information on employment and how to get access to the German and Swiss labor market. The seminar also provides an overview of labor conditions in Germany and Switzerland, the multifaceted forms of employment, business cultures and useful tips and information for the job entry in a German or Swiss company.</td></tr>
<tr><td colspan="6">Intended learning outcomes

Upon completion of this module, students will be able to:
1. structure their ideas to write clear summaries, coherent paragraphs and cohesive literature reviews;
2. write different segments of an academic paper employing writing styles that display advanced grammar and precise and concise language use;
3. successfully find and evaluate sources for research;</td></tr>
</table>

4. use citation and referencing styles applicable for their discipline; 5. Avoid unintentional plagiarism and adhere to the code of academic integrity. 6. understand labor conditions in Germany and Switzerland. 7. understand the typical business cultures in German or Swiss companies.
Indicative literature
Usability and relationship to other modules
Examination Type: Module Examination Assessment Type: Term Paper (Report) Length: 10 pages Weight: 100% Scope: All intended learning outcomes of this module.

4.3.5 Agile leadership and strategic management

<table>
<tr><td colspan="2">Module name
Agile Leadership and Strategic Management</td><td>Module code
tbd</td><td>Level (type)
Year 2</td><td>CP
2.5</td></tr>
<tr><td colspan="5">Module components
Number | Name | Type | CP
tbd | Agile Leadership and Strategic Management | Lecture | 2.5</td></tr>
<tr><td>Module coordinator
NN</td><td colspan="2">Program affiliation
▪ MSc Quantum Software Engineering and Computer Science</td><td colspan="2">Mandatory status
Mandatory for QSECS</td></tr>
<tr><td colspan="2">Entry requirements
Pre-requisites
▪ none

Co-requisites
☒ None

Knowledge, abilities, or skills</td><td>Frequency
Annually (Fall)

Duration
1 semester</td><td colspan="2">Forms of learning and teaching
▪ Lecture (17.5 hours)
▪ Private study (45 hours)

Workload
62.5 hours</td></tr>
<tr><td colspan="5">Recommendations for preparation

N.A.</td></tr>
<tr><td colspan="5">Content and educational aims

This module focuses on key strategic aspects of the leadership and strategy development processes, specifically strategic problems solving, alignment, engagement and copying with black swans and paradigm shifts. The module draws on insights from a variety of fields such as business strategy, problem solving, strategic communication, strategic planning, and strategic resilience. To build a holistic understanding, the module deals with the following topics:
● The strategic process: from analysis, definition, planning and evaluation
● Hypothesis driven problem solving
● Pyramid principle strategic communication
● Antifragile strategies

The module assessment will consist of three presentations. Students will know in the first session which topics need to be covered in their presentations.</td></tr>
<tr><td colspan="5">Intended learning outcomes

Upon completion of this module, students will be able to:

1. Understand and analyse business strategies
2. Understand and analyse strategic statements and levels of ambition
3. Understand opportunities and threats on the external environment
4. Evaluate sources of competitive advantage as well as strategic strengths and weaknesses
5. Analyse core challenges of agile leadership and strategy development
6. Develop and communicate strategic initiatives
7. Apply this knowledge to real-world strategic planning processes</td></tr>
<tr><td colspan="5">Indicative literature

Sola, D. & Couturier, J, 2013, How To Think Strategically, FT Publishing International.</td></tr>
<tr><td colspan="5">Usability and relationship to other modules</td></tr>
</table>

Examination type: module examination Assessment type: Presentations Duration: 30 min Weight: 100% Scope: All intended learning outcomes.

4.3.6 Customer-centric mindset and agile delivery management

<table>
<tr><td colspan="2">Module name
Customer-centric mindset and agile delivery management</td><td>Module code
tbd</td><td>Level (type)
Year 2</td><td>CP
2.5</td></tr>
<tr><td colspan="5">Module components
Number | Name | Type | CP
tbd | Customer-centric Mindset and Agile Delivery Management | Lecture | 2.5</td></tr>
<tr><td>Module coordinator
NN</td><td colspan="2">Program affiliation
▪ MSc Quantum Software Engineering and Computer Science</td><td colspan="2">Mandatory status
Mandatory for QSECS</td></tr>
<tr><td colspan="2" rowspan="2">Entry requirements
Pre-requisites
▪ none
Co-requisites
☒ None
Knowledge, abilities, or skills</td><td>Frequency
Annually (Fall)</td><td colspan="2">Forms of learning and teaching
▪ Lecture (17.5 hours)
▪ Private study (45 hours)</td></tr>
<tr><td>Duration
1 semester</td><td colspan="2">Workload
62.5 hours</td></tr>
<tr><td colspan="5">Recommendations for preparation

N.A.</td></tr>
<tr><td colspan="5">Content and educational aims

Successful firms are forced to walk a tightwire between meeting market needs and creating organizational efficiencies. Just how they do this requires, organization, insights, management understanding and determination. The modern manufacturing or service firm is simultaneously engaged in three core processes. 1) The design and development of products and services (BUILD), 2) The efficient and effective delivery of those products and services to the market (DELIVER), and 3) The process of gaining customers that wish to purchase those products and services or enter into transactions with the firm (CAPTURE). How it organizes and the processes it adopts are key to a firm´s ability to optimize these often divergent but highly interdependent activities.

While these three processes are often at odds with each other, this module will inform, challenge, and enlighten the participants on a) The best practices in each of these areas, b) The ways to improve their understanding and implementation of course concepts, and c) The trends that they will invariably deal with in the near future. In this module, students touch upon the design of innovative R&D, operations, and marketing strategies that provide firms with a strategic and sustainable competitive advantage that is capable of utilizing global resources and capturing markets. These strategies will constantly be viewed in a competitive, resource constrained, and capital efficient marketplace.

The module assessment will consist of three presentations. Students will know in the first session which topics need to be covered in their presentations.</td></tr>
<tr><td colspan="5">Intended learning outcomes

Upon completion of this module, students will be able to:</td></tr>
</table>

1. Analyze critically the task of going to market under contemporary conditions and to examine the major functions that comprise the marketing servicing task 2. Evaluate various types of policies that can be employed in guiding market centric activities 3. Develop an awareness of the major types of market problems faced by organizations, with emphasis on sound analytical approaches to effective problem-solving decisions 4. Analyze different business models and understand how the marketing function can be employed to enhance them
Indicative literature Chernev, A., 2018, Strategic Marketing Management.
Usability and relationship to other modules
Examination type: module examination Assessment type: Presentations Duration: 30 min Weight: 100% Scope: All intended learning outcomes.

4.4 Research project, capstone project, and master's thesis

4.4.1 Capstone project 1

<table>
<tr><td colspan="2">Module name
Capstone project 1</td><td>Module code
tbd</td><td>Level (type)
Year 1</td><td>CP
5</td></tr>
<tr><td colspan="5">Module components
Number | Name | Type | CP
tbd | Capstone project 1 | Project | 5</td></tr>
<tr><td>Module coordinator

Prof. Dr. Manuel Oriol</td><td colspan="2">Program affiliation

• MSc Quantum Software Engineering and Computer Science</td><td colspan="2">Mandatory status

Mandatory for QSECS</td></tr>
<tr><td colspan="3">Entry requirements

Pre-requisites
☒ None

Co-requisites
☒ None

Knowledge, abilities, or skills
• Programming skills in an imperative language at CS bachelor level
• Algorithms and data structure at CS bachelor level</td><td>Frequency

Annually (Fall)

Duration

1 semester</td><td>Forms of learning and teaching

• Project group meetings (42 hours)
• Group-based and independent project work (83 hours)

Workload

125 hours</td></tr>
<tr><td colspan="5">Recommendations for preparation

Train and advance programming, read about agile development, watch videos on ideation processes and read books on team and teamwork.</td></tr>
<tr><td colspan="5">Content and educational aims

This series of Capstone modules gives the possibility of experiencing knowledge and expertise learned in the master by a posteriori analysis, transformational adaptation and coherent planning hands-on practice. The series spans over three modules during which students develop a complete product from scratch. The project starts with an ideation process, creation of clickable demos and initial requirements. It continues with the practical creation of a software architecture and development of the solution. It then finishes with application of artificial intelligence and cybersecurity. During the project, students are going through various steps during which they are encouraged to talk directly to potential real-world customers and users, thus gathering an understanding of what real users and customers for their project might want.
The project is organized in tribes (20-30 people) in charge of exactly one project. The tribes are then further split in agile teams working with the advice of the instructors and the assistants (impersonating the business owners and product owners). The teams can be geographically distributed and work with an up-to-date environment supported with open source IDEs and engineering tools. Few lectures indicate the best practices to follow and the interim goals. Periodic meetings with instructor and teaching assistants steer the process towards the overall goal.

This instance is the first semester of the Capstone project that focuses on ideation and requirements elicitation.</td></tr>
<tr><td colspan="5">Intended learning outcomes</td></tr>
</table>

1. Create and propose mocks
2. Perform requirements elicitation
3. Prototype
4. Approach customers and users
5. Specify user stories
6. Organize themselves through collaborative tools
7. Understand team dynamics and resolve most interpersonal issues

Indicative literature

Agile the good the hype and the ugly. Book by Bertrand Meyer

The Five Dysfunctions of a Team. Book by Patrick Lencioni

Group dynamics and Teams interventions. Book by Timothy M. Franz

Online resources on team dynamics:

- *https://www.challengeapplications.com/stages-of-team-development*
- *https://agilescrumguide.com/blog/files/tag-5-stages-of-team-development.html*

Usability and relationship to other modules

It is highly recommended to take the three Capstone project modules in their numerical order to gain the full experience of the project.

Examination Type: Module component examinations

Assessment: Project Weight: 100%

Scope: All intended learning outcomes of the module.

4.4.2 Capstone project 2

<table>
<tr><td colspan="3">Module name
Capstone project 2</td><td>Module code
tbd</td><td>Level (type)
Year 1</td><td>CP
15</td></tr>
<tr><td colspan="6">Module components
Number | Name | Type | CP
tbd | Capstone project 2 | Project | 15</td></tr>
<tr><td>Module coordinator

Prof. Dr. Manuel Oriol</td><td colspan="3">Program affiliation

• MSc Quantum Software Engineering and Computer Science</td><td colspan="2">Mandatory status

Mandatory for QSECS</td></tr>
<tr><td colspan="3">Entry requirements

Pre-requisites

☒ None

Co-requisites

☒ None

Knowledge, abilities, or skills

• Programming skills in an imperative language at CS bachelor level
• Algorithms and data structure at CS bachelor level</td><td>Frequency

Annually (Spring)

Duration

1 semester</td><td colspan="2">Forms of learning and teaching

• Project group meetings (42 hours)
• Group-based and independent project work (83 hours)

Workload

125 hours</td></tr>
<tr><td colspan="6">Recommendations for preparation

Train and advance programming, read about agile development, watch videos on ideation processes and read books on team and teamwork.</td></tr>
<tr><td colspan="6">Content and educational aims

This series of courses courses gives the possibility of experiencing knowledge and expertise learned in the master by aposteriori analysis, transformational adaptation and coherent planning hands-on practice. The course series spans over three courses during which students develop a complete product from scratch. The project starts with an ideation process, creation of clickable demos and initial requirements. It continues with the practical creation of a software architecture and development of the solution. It then finishes with application of artificial intelligence and cybersecurity. During the project students are going through various steps during which they are encouraged to talk directly to potential real-world customers and users, thus gathering an understanding of what real users and customers for their project might want.
The project is organized in tribes (20-30 people) in charge of exactly one project. The tribes are then further split in agile teams working with the advice of the instructors and the assistants (impersonating the business owners and product owners). The teams can be geographically distributed and work with an up-to-date environment supported with open source IDEs and engineering tools. Few lectures indicate the best practices to follow and the interim goals. Periodic meetings with instructor and teaching assistants steer the process towards the overall goal.

This instance is the second semester of the capstone project that focuses on architecture and base implementation.</td></tr>
<tr><td colspan="6">Intended learning outcomes

1. Describe and defend a software architecture</td></tr>
</table>

2. Code in groups
3. Code as a large team
4. Integrate independent works
5. Use a source code versioning system
6. Specify user stories
7. Hold practical discussions with stakeholders
8. Organize themselves through collaborative tools
9. Understand team dynamics and resolve most interpersonal issues

Indicative literature

Agile the good the hype and the ugly. Book by Bertrand Meyer

The Five Dysfunctions of a Team. Book by Patrick Lencioni

Group dynamics and Teams interventions. Book by Timothy M. Franz

Online resources on team dynamics:

- *https://www.challengeapplications.com/stages-of-team-development*
- *https://agilescrumguide.com/blog/files/tag-5-stages-of-team-development.html*

Usability and relationship to other modules

It is highly recommended to take the three Capstone project modules in their numerical order to gain the full experience of the project.

Examination type: module component examinations

Assessment: Project Weight: 100%

Scope: All intended learning outcomes of the module.

4.5.4 Capstone project 3

<table>
<tr><td colspan="3">Module name
Capstone project 3</td><td>Module code
tbd</td><td>Level (type)
Year 1 and 2</td><td>CP
15</td></tr>
<tr><td colspan="6">Module components
Number | Name | Type | CP
tbd | Capstone project | Project | 15</td></tr>
<tr><td>Module coordinator
Prof. Dr. Manuel Oriol</td><td colspan="3">Program affiliation
• MSc Quantum Software Engineering and Computer Science</td><td colspan="2">Mandatory status
Mandatory for QSECS</td></tr>
<tr><td colspan="2">Entry requirements
Pre-requisites
☒ None</td><td>Co-requisites
☒ None</td><td>Knowledge, abilities, or skills
• Programming skills in an imperative language at CS bachelor level
• Algorithms and data structure at CS bachelor level</td><td>Frequency
Annually (Fall)

Duration
1 semester</td><td>Forms of learning and teaching
• Project group meetings (42 hours)
• Group-based and independent project work (83 hours)

Workload
125 hours</td></tr>
<tr><td colspan="6">Recommendations for preparation
Train and advance programming, read about agile development, watch videos on ideation processes and read books on team and teamwork.</td></tr>
<tr><td colspan="6">Content and educational aims

This series of courses gives the possibility of experiencing knowledge and expertise learned in the master by aposteriori analysis, transformational adaptation and coherent planning hands-on practice. The course series spans over three courses during which students develop a complete product from scratch. The project starts with an ideation process, creation of clickable demos and initial requirements. It continues with the practical creation of a software architecture and development of the solution. It then finishes with application of artificial intelligence and cybersecurity. During the project students are going through various steps during which they are encouraged to talk directly to potential real-world customers and users, thus gathering an understanding of what real users and customers for their project might want.
The project is organized in tribes (20-30 people) in charge of exactly one project. The tribes are then further split in agile teams working with the advice of the instructors and the assistants (impersonating the business owners and product owners). The teams can be geographically distributed and work with an up-to-date environment supported with open-source IDEs and engineering tools. Few lectures indicate the best practices to follow and the interim goals. Periodic meetings with instructor and teaching assistants steer the process towards the overall goal.

This instance is the third semester of the Capstone project that focuses on integrating artificial intelligence, cybersecurity, and develops practices.</td></tr>
<tr><td colspan="6">Intended learning outcomes

1. Know practical cybersecurity</td></tr>
</table>

2. Hold practical discussions with stakeholders
3. Practice of machine learning
4. Work with continuous improvements tools
5. Organize themselves through collaborative tools
6. Understand team dynamics and resolve most interpersonal issues

Indicative literature

Agile the good the hype and the ugly. Book by Bertrand Meyer

The Five Dysfunctions of a Team. Book by Patrick Lencioni

Group dynamics and Teams interventions. Book by Timothy M. Franz

Online resources on team dynamics:

- *https://www.challengeapplications.com/stages-of-team-development*
- *https://agilescrumguide.com/blog/files/tag-5-stages-of-team-development.html*

Usability and relationship to other modules

It is highly recommended to take the three Capstone project modules in their numerical order to gain the full experience of the project.

Examination type: module component examinations

Assessment: Project Weight: 100%

Scope: All intended learning outcomes of the module.

4.5.5 Master's thesis

<table>
<tr><td colspan="3">Module name
Master's thesis</td><td>Module code
tbd</td><td>Level (type)
Year 2</td><td>CP
30</td></tr>
<tr><td colspan="6">Module components
Number | Name | Type | CP
tbd | Master's thesis | N.A. | 30
tbd | Colloquium | | </td></tr>
<tr><td colspan="2">Module coordinator
Prof. Dr. Bertrand Meyer</td><td colspan="2">Program affiliation
• MSc Quantum Software Engineering and Computer Science</td><td colspan="2">Mandatory status
Mandatory for QSECS</td></tr>
<tr><td colspan="3">Entry requirements
Pre-requisites
• None

Co-requisites
☒ None

Knowledge, abilities, or skills
• Proficiency in the area of the chosen thesis topic.</td><td>Frequency
Annually (Spring)

Duration
1 semester</td><td colspan="2">Forms of learning and teaching
• Private Study (725 hours)
• Colloquium (25 hours)

Workload
750 hours</td></tr>
<tr><td colspan="6">Recommendations for preparation

Read the Syllabus.</td></tr>
<tr><td colspan="6">Content and educational aims

The aim of this module is to train students to motivate, design, carry out and document a 6-month project. The thesis topic is determined in mutual agreement with the module instructor. Among others, it may arise
• from research in the instructor's research area (research thesis),
• from a collaboration with a company (industry thesis), or
• from a student-driven product development idea for a start-up (start-up thesis)

In all cases, the instructor needs to agree to supervise the thesis.

The thesis work comprises the full cycle of a scientific project, starting from the identification of an open research question or focus of the work with a survey on the state of the art in research / industry / business, over the formulation of a concrete objective to the design, implementation and evaluation of an object of interest by scientific measures and with respect to the state of the art. All results are documented in the thesis report. document all of this in a thesis report. Depending on the type of thesis (research / industry / start-up), additional components, like a research / business plan, might be a necessary part of the thesis. Irrespective of the thesis type, it is a mandatory part of each thesis to develop a digital system as known from the various branches of Computer Science and Software Engineering.

All above outlined work should be done with as much self-guidance as can be reasonably expected. The instructor will likely give substantial guidance for the first steps, whereas the other aspects will be addressed with larger degrees of self-guidance. The project consists of the thesis report (target size: 30–60 pages, and an oral presentation at the end of the course.</td></tr>
<tr><td colspan="6">Intended learning outcomes</td></tr>
</table>

Discipline-Specific Skills (subject area depending on individual project):

1. understanding, at a professional level, of a circumscribed segment of the project in its environment (research, industry, startup);
2. ability to apply specific and selected QSECS techniques, as required for the project, at a professional level;
3. general professional skills;
4. designing and carrying out the full cycle of a project by scientific means in a professional manner;
5. writing a thesis such that it could be submitted to a scientific publication venue, as a project report to a funding agency / industrial client, or as a proposal for start-up funding;
6. presentation of project results for specialists and non-specialists.

Indicative literature

N.A.

Usability and relationship to other modules

Examination Type: Module Examination

Assessment Component 1: Thesis — Length: 30 – 60 pages, Weight: 80%

Scope: All intended learning outcomes of this module.

Assessment Component 2: Oral Examination (Defense) — Duration: 20 minutes, Weight: 20%

Scope: Mainly presentation of project results but the presentation touches all intended learning outcomes

Completion: This module is passed with an assessment-component weighted average grade of 45% or higher.

C›ONSTRUCTOR
INSTITUTE

Constructor Institute Schaffhausen
Rheinweg 9 · 8200 Schaffhausen · Switzerland
+ 41 52 630 28 38
institute.constructor.org