A.1 Academic quality - Course content (30 % of the max. score)

A.1.1 Describe the EMMC objectives (including in socio-economic terms) in relation to the needs analysis in the field(s) concerned.

Software Engineering (SE) is a discipline that studies the creation of **high quality software** in a systematic, controlled and efficient manner. Specifically, SE aims at establishing and using sound engineering principles and methods in order to obtain economically viable software that is reliable and works on real machines[1]. According to the most used definition, SE is an engineering discipline that applies the principles of computer science and mathematics to achieving cost-effective solutions to software problems in the development, operation, and maintenance of software[2][3]. Nowadays, SE discipline is even much more that this simple statement, it comprises foundations from a wide variety of disciplines and provides knowledge and skills for integrating theory and practice. By the end of 1960's, when the term Software Engineering was coined[4], it was clear that high quality software requires the development of specific practices and methods based on both the analytical and descriptive tools developed within computer science and the rigor that the engineering disciplines bring to the reliability and trustworthiness of the artefacts they engineer. At that time, software was the privilege of universities, research centers, and few large companies, but the poor quality of software raised nevertheless a strong concern among the experts. Today software permeates any aspect of our life, has reached the mass, and supports systems sophisticated like Global Positioning System (GPS), life critical like Nuclear Reactor Control systems (NRC), and business essential like Enterprise Resource systems (ERP). As we entrust our lives to software, its quality and efficiency has become central in our society increasing the need of professionals and researchers in SE. The U.S. Government Accountability Office regularly issues reports recounting the challenges the U.S. government faces in creating large-scale, reliable, software-intensive systems on schedule, on budget, and with expected functionality[5]. The European Union has played an important role in the promotion of research and dissemination of SE related knowledge in Europe. A lot of research projects as well as research networks in the area have been funded to improve competitiveness of European software industry (the members of this consortium have participated in some of such initiatives). As software technology has rapidly spread through every aspect of modern societies, the challenge of educating software engineers has taken on new form and become more complex and urgent[6].

Expertise, competence, and professionalism in SE comprise cross-disciplinary competences related to management and quality, novelty and creativity, standards, individual skills, and teamwork and professional practice[7]. From an educational point of view, although undergraduate SE programmes are being set up in some (mostly non-European) countries, SE in the graduate programs is usually taught as part of Computing (Informatics) programmes[18]. Such programmes tend to cover a wide variety of subjects, Programming, Artificial Intelligence, Computer Architecture, etc. As consequence, SE contents do not usually get enough coverage in academic programmes specially in the graduate ones. Therefore, it becomes necessary to develop SE graduate programmes that provide students with the knowledge they need to cope with their professional responsibilities. In 2009, a large study on the development of graduate program in twenty eight universities revealed the need of alignment in graduate studies in SE[8]. In the same year, ACM issued the first of recommendations for developing and improving curricula that provide software engineering education at the master's degree level. This framework was specifically designed for students that aim at undertaking a career in the practice of SE and who is not necessarily interested in pursuing a doctorate in

---

[1] Bauer, F.L., Software Engineering, *Information Processing*, 71, 1972

[2] Ford, G. SEI Report on Undergraduate Software Engineering Education, *CMU/SEI-90-TR-003*, 1990

[3] IEEE STD 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology, *IEEE Computer Society*, 1990

[4] To be precise in 1968 in the first NATO Conference on Software Engineering in Garmisch Partenkirchen, which later became International Conference on Software Engineering. http://homepages.cs.ncl.ac.uk/brian.randell/NATO/

[5] Information Technology: Better Informed Decision Making Needed on Navy's Next Generation Enterprise Network, GAO-11-150, March 11, 2011

[6] Mehdi Jazayeri. 2004. The Education of a Software Engineer. In *Proceedings of the 19th IEEE International Conference on Automated Software Engineering* (ASE '04). IEEE Computer Society, Washington, DC, USA, 18-xxvii

[7] SWEBOK, *Guide to the Software Engineering Body of Knowledge*, P. Bourque & R. Dupuis (Eds.), IEEE Computer Society Press, 2004.

[8] Pyster, A., Turner, R., Henry, D., Lasfer, K., Bernstein, L. Master's Degrees in Software Engineering: An Analysis of 28 University Programs, *IEEE Software*, September-October 2009, 94-101.

SE.[9] No other reference is available for graduate studies in SE[10]. In Europe, our European Masters Course in Software Engineering (EMCSE), funded as Erasmus Mundus in 2006[11] is the first implementation of a well established graduate international program in SE that aims at forming students not only as SE practitioners but also as SE researchers. In addition, from the experience with EMCSE and with the strong will to extend this experience to further degree of education, **two out of four current EMCSE partners[12] have also established a joint doctorate, the International Doctorate in Software Engineering (IDSE)[13].** The doctorate includes other four partners: Simula School of research and Innovation, Lysaker, Norvegia, The Aristotele University of Thessaloniki, Thessaloniki Grecia, Institute of Software Academy of Science (ISCAS), Beijing, China. At our knowledge, EMCSE and IDSE represent the first example in Europe of a consolidate partnership among universities running a full educational graduate path that aims at the excellence in SE. We also believe that this initiative is unique in the world. With the lesson learned and the experience accumulated, partners joining this renewal aim at keeping the best and consolidated practices and tools and integrate new instruments that have been discussed and identified in running the first five years edition of the EMCSE program[14].

The primary objective of EMCSE is to train software engineers that meet the requirements of international software practice today, offering them an inter-European study programme based on the coordination of the strengths of each of the partner institutions. The partners' institutions have strong relationships with European software companies that guarantee that students of this programme will be able to deal with current problems in industry, research, and innovation. The EMCSE consortium is set up by universities and professors excelling in the SE area, which places the EMCSE in an unbeatable position for contributing to European competitiveness and excellence in the SE field. Staff of EMCSE includes some of the most able and active members of the European SE scene (prof. Dieter Rombach, prof. Claes Wohlin, and prof. Giancarlo Succi) who have made a name for themselves among the most influential SE professors in Europe. They are also full professors at some of the best universities in their respective countries. As teachers, they have been teaching SE for over fifteen years and have been responsible for organising and developing undergraduate and graduate programmes at the universities for which they work or have worked. Additionally, they play an active role in international initiatives designed to define SE teaching. As researchers, they have published their works in the internationally highest ranked SE journals and conferences, as can be seen from the abridged curricula attached in the Curricula Vitae Appendix or at their respective web pages. Some particulars that attest this are outlined in the following:

- **FUB**: FUB is a newly established university (1997) with a strong European calling owing to the strategic position in which it is located (a region in Italy where three languages have been spoken for 30 years) and whose scholars come from different international universities or research centres (Helsenki unversity, Calgary University, University of Trento, University of Genoa, ETH Zurich). **Giancarlo Succi** is the director of the Center for Applied Software Engineering and dean of the Faculty of Computer Science. He is Rector's delegate for Information and Communication Technology. He has been an active SE professor at different international universities, for example, at the University of Alberta (Canada), at the State University of New York (USA), at Virginia Tech (USA), and from 2001 at FUB. He is the author of more than 150 papers published in international conferences and journals. He has been one of the founders of the International Symposium in Open Source Software and systems in 2005.
- **BTH**: In 1990 BTH started the first Swedish undergraduate programme in SE. In 1995 the university launched the first Master of Science in SE programme in Sweden. BTH is currently running the largest software engineering research group in Sweden and was ranked as number five among the top institutions in systems and software engineering in the world by the Journal of Systems and Software based on publications in 2003-2007, and as number six for the period 2004-2008. **Claes Wohlin** is

[9] GSwE2009 (2009) Graduate Software Engineering 2009, ACM http://www.acm.org/education/curricula-recommendations

[10] Bagert D.J. and Mu X.(2005) Current State of Software Engineering Master's Degree Programs In the United States, 35th ASEE/IEEE Frontiers in Education Conference

[11] Consortium's number: 2007 - 0065

[12] Two out the three academic partner of this proposed new consortium

[13] www.idse.se

[14] The current EMCSE consortium includes Politecnica de Madrid as coordinator. Due to lack of adequate internal administrative support to EMCSE and a change of mission of the university the coordinator had to withdrawn its partnership in this renewal

heading the Software Engineering Research Lab and has served as the Pro Vice Chancellor at BTH in the recent past. He has been included in the list of the top 15 scholars ranked by the Journal of Systems and Software four times, and is editor in chief of the Information and Software Technology journal. In addition to his affiliation at BTH, ClaesWohlin is a professor of software engineering at Blekinge Institute of Technology, Sweden. He has previously held professor chairs at the universities in Lund and Linköping. His research interests include empirical methods, software management and software process improvement. Claes Wohlin received a PhD in communication systems from Lund University. He is Editor-in-Chief of Information and Software Technology and member of three other journal editorial boards. ClaesWohlin was the recipient of Telenor´s Nordic Research Prize in 2004 for his research in software engineering for telecommunication systems. He has been a visiting professor at Chalmers University of Technology and since January 1, 2009 he is a professorial visiting fellow at University of New South Wales, Sydney in Australia. Since 2011, he is a member of the Royal Swedish Academy of Engineering Sciences.

- **UKL**: UKL has strong ties to the Fraunhofer Institute for Experimental Software Engineering (IESE) Technology (one of the best European SE research centres). **Dieter Rombach** is the director of the IESE, 2006 Program Chair of the International Conference on Software Engineering (ICSE) (the internationally highest ranked SE conference), and member of the board of editors of Transactions on Software Engineering (the internationally highest ranked SE research journal).

Additionally, the EMCSE partners have been cooperating for many years through the International SE Research Network (**ISERN**[15]), which is the most influential research network in the SE community. In this network, they have exchanged staff among research groups and have been members of PhD commissions at other EMCSE institutions, as well as have used other means of regular collaboration between universities. They are part of the steering committee of the annual International Symposium on Empirical Software Engineering and Measurement (in 2008 the symposium was hosted by UKL an in 2010 by FUB).

During the first three years of EMCSE, the collaboration among the partners have consolidated and extended at the educational level with several co-supervision of master theses and the definition of a coherent shared model of graduate studies in SE.

A.1.2 Explain the EMMC's added value compared with existing masters courses in the same field at national, European and international level.

SE is one of the CS strengths of European universities. The need for such skills and knowledge is constantly growing, and Europe is playing a very active role in generating and offering such knowledge. The generation of such knowledge has been supported by EU and national agencies by means of research projects in which the EMCSE partners have participated. Notice also that European researchers hold a very good position in high quality SE journals publications[16]. In 2004-2008 of the top fifteen scholars in the field of SE, eight are European (in 2003 -2007 one of whom is a proponent of this proposal) as compared, for example, with two from the USA. In particular, Europe is also a leader on Software Engineering issues, like Requirements or Empirical Research Methods (particularly addressed in our integrated programme). For example, in this period, of the top-five scholars publishing in requirements, 4 are European and 1 from the USA and of the top 5 institutions 2 are European, 2 American and 1 Canadian [17]. Similarly, in Empirical SE at the international research network ISERN about 51% of organizations are European –including our consortium members, while only 9% are from USA[15].

Despite the great potential workforce in Europe, SE as discipline is taught at the master level as part of the Computer Science curriculum in Europe as well as in the rest of the world[18]. In 2004, in Europe only United kindom high educational systems have a relevant number of curriculum in SE (about a quarter of computer science curricula). Italy has none, Germany has two and Sweden has five curricula. These numbers are extremely low with respect to the number of the computer science curricula [18].

A survey in 2009 on twenty eight American universities reports that SE is considered a specialization of Computer Science and is taught by few dedicated professors of a faculty in Computer Science. The same happens in Europe where initial studies on the development of SE curricula describe the SE discipline as subfield of Computer Science and as such too distant from the field of application of SE (industry)[19]. In 2003, a three years initiative to explore a joint MSc curriculum in SE according to the Bologna framework,

---

[15] http://isern.iese.de/Portal/

[16] W. Eric Wonga,∗, T.H. Tseb, Robert L. Glassc, Victor R. Basili d, T.Y. Chene. *An assessment of systems and software engineering scholars and institutions(2003–2007 and 2004–2008)* The Journal of Systems and Software. 2011

[17] Data gathered from the requirements bibliography recorded at: http://www.reqbib.com/index.htm of Alan M. Davis

[18] O. Dieste, N. Juristo and A. M. Moreno. How higher-education systems influence software engineering degree programs. IEEE Software. 21(4), July 2004, pp. 78–85

[19] http://perun.pmf.uns.ac.rs/msc-se/

has been made by four European universities led by the Faculty on Mathematical Sciences of the University of Humboldt, Berlin, Germany under the EU funded programme TEMPUS (nr. 392,407.00). The result of the three years project (with about 350,000 euros funding) was the identification of a joint MSc in SE and its adoption as part of the regular curriculum in one of the partner universities. The project did not follow up in a joint curriculum in all the partners, as this was not the goal of the project, but the experience it reports is significant for the validation of the EMCSE initiative. The project reported four major recommendations:

1. A strict project organization is necessary to coordinate the cooperation between different project partners. EMCSE *has a well established schedule in which partners meets twice a year for the organization and management of the EMMC, and a Task Force will work to enhance and calibrate the curriculum content and organization according with new research and industry demand (see section A2.1 Figure2).*

2. Common principles of quality assurance are crucial in such a multi-lateral project: curriculum validation by a validation panel covering academics and industrialists; quality assurance of teaching materials by defining a strict process of their development and assessment; common principles of students' selection and students' assessment; coordination of the activities by a quality team.

EMCSE joint board consists of experts in the field of SE that have been meeting to identify a common curriculum not based on the aggregation local courses but 1)on common core areas identified by reading and discussing content of each single course taught at the partner universities and 2) that reflect consortium expertise and fundamental knowledge in SE. Quality is monitored locally and at consortium level with various forms of feedback(*see A5). Students are selected on common principle in respect of the local regulations of the partners. Students are assessed in ECTS systems and rules for conversions of local systems are defined to avoid any subjective deviation. A well defined and public procedure of co-supervision has been in place for all the students of EMCSE.*

3. It is rather advantageous that experts of certain special fields are being responsible to work out the teaching materials in that field. In that way, high quality of the technical contents could be guaranteed. EMCSE *teachers are a leading group of researchers and practitioners in Europe (e.g. teachers in UKL are associated to the Fraunhofer Institute in SE of Kaiserslautern and all the partners are member of ISERN). A Task Force will ensure the calibration of the content with actual needs and demand of the technological and social world.*

4. The involvement of all partner institutions in the curriculum development process is necessary for the success of the project, in particular for the acceptance of the project results. It proved to be not so easy to adopt the new master curriculum, especially when partners were in the middle of their university reforms and with unstable legislatives.

EMCSE partners have been able to react to changes fast and efficiently. The curriculum has evolved according to the regional regulations without any effect on the students' studies.

In this sense, EMCSE represents a unique opportunity for graduate students as it joins top researchers in the field as teachers with a dedicated and calibrated joint curriculum in SE that has been tested and reviewed for some years now.

EMCSE is also unique among the one hundred forty Erasmus Mundus masters running in 2011/2012. Only nine EMMCs are in subjects related to computer science and all of them of complementary themes. If we look, for example, at the ones involving Italian universities only four out of fifty six are in the area of computer science including first five years edition of EMCSE.

The EMCSE surpasses existing SE postgraduate educational offers in different ways. From a strategic point of view:

- Enabling students to train in SE with some of the most outstanding European researchers.
- Enabling students to get a broad view of software development in different European countries (note that the EMCSE consortium is composed of countries from northern and southern Europe with a view to encouraging the integration of and acquaintance with their different cultures)
- Enabling students to working on solving real industry problems in close collaboration with the research groups and their industry partners.

From a tactical point of view:

- Providing students with broader educational opportunities than are offered by national masters programmes in the field.
- Allowing students to gather expertise on different working strategies, providing them with two tutors from two different institutions of the consortium and a master thesis co-supervised by teachers of the two different institutions.
- Consolidate further a coordinated SE programme that have been discussed, experimented, and

established since three years by all partners of the consortium and that aims at defining a "de facto" standard for graduate SE education in Europe, as well to shape SE as a discipline on its own

- Providing students with a graduate program that is constantly monitored by the EMCSE Joint Board and Task Force in terms of its quality, its evolution, and the employability of the EMCSE graduate students in the industry and in the academic world.
- Provide a curriculum that enables EMCSE students to access to international coordinated doctoral studies (e.g. IDSE)
- Train a future class of international researchers and practitioners that will foster research and innovation in software for the economic development and sustainability of the European technological market. For example, some of the partners are highly qualified subjects in the field of Open Source SE and development and Agile and distributed SE.

Through the integrated programme, the expertise of the teaching staff involved, and the reputation of the consortium institutions, EMCSE will aim at contributing in driving Europe into an international leading position in the area of SE and its applications. In our opinion, the EMCSE is set up in accordance with European SE needs and it is likely to help Europe to gain a position from which it compete with the USA in the SE field. This should be used to attract more CS students from third-countries and to strengthen ties with their institutions of origin. As we mentioned, this renewal has a novelty in this respect. The EMCSE Tasks Force. One of the major objectives of the Task Force is to explore new ways to attract students in accordance with IT and economic market fluctuations (see A.1.3).

A.1.3 Present the structure and content of the EMMC and justify the added value and relevance of the mandatory mobility component.

EMCSE program has been developed having in mind:

- Recommendations about SE education provided by international standards like the ACM Curriculum in SE GSwE[9],
- Recommendations about the SWEBOK[7] developed jointly by IEEE and ACM (to which some of the promoters of this programme have contributed),
- Practical discussion held at international software engineering educational forums ICSE[20],
- Publications in relevant international journals about SE education[21],
- Research and teaching strengths of the different partners,
- The lesson learned and the experience gather during the first three years,
- The evolution of the Information and Technology market including new frontiers in the private and public sector inspired by the "Digital Agenda for Europe" of the European Commission[22].

Accordingly, the EMCSE course makes up a jointly developed curriculum fully recognized by the consortium members. Although different courses are delivered separately by each institution, they make up a common SE programme in which students can specialize in particular disciplines that are the strengths of each institution and according to innovation demand of research and industry of the European unique market.

EMCSE structure is built upon four pillars: mobility tracks, mobility rules, mobility monitoring, and joint curriculum.

Mobility tracks. The EMCSE consolidate schema for students mobility is based on annual exchange. A student of EMCSE visits one university the first year and a second university the second year of the course moving between the two universities in summer. This exchange schema has three strengths: 1) students have one year time to get familiar with local structures, culture, and languages, 2) students have enough time to interact with teachers and colleagues and to do their exams with a regular pace 3) administration offices can manage incoming and outgoing students with needed accuracy and professionalism. This schema also

---

[20] Panel at the International Conference on Software Engineering (ICSE) 2010, Panel on Master's Programs in Software Engineering. Panel chair: Mehdi Jazayeri. Panel members: John Brackett, Ivica Crnkovic, Bertrand Meyer, Art Pyster, Tony Wasserman, and Mansour Zand. .

[21] Bach J. SE Education: We're on Our Own. IEEE Software, November/December, 1997, pp: 26, 28.
Dawson R, Introducing Software Engineers to the Real World. IEEE Software, November/December 1997, pp. 37-43.
Dart P. et al. Developing an Accredited Software Engineering Program. IEEE Software, November/December 1997, pp. 66-70.
Finlestein A, Kramer J. Software Engineering: A Roadmap. In The Future of Software Engineering. ACM Press, 2000, pp: 3-24.
Parnas DL. Software Engineering Programs Are Not Computer Science Programs. IEEE Software, November/December 1999, pp.19-30.
Pour G, Griss ML, Lutz M. The Push to Make Software Engineering Respectable. Computer, May 2000, vol 33 (5), pp. 35-43.

[22] http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:52010DC0245:EN:HTML

enables students 1. being in contact with the local society and industry, 2. having a better understanding of the mechanism and future chances of employment and 3. practicing local languages.
For every course edition at the opening of the enrolment session, the consortium publishes on its web site available positions for each mobility track. At their enrolment, students make their preference based on the study plan of the mobility tracks and according to any EACEA constraints for grants (see below). See section A.2.3 for details of students' selection and admission. In the appendix, we provide schemas for mobility tracks, study plan by mobility track and syllabi of the courses.
Students that have obtained their degree from countries of one of the partner university are 1) assigned to partners of the other two different countries if they have an EACEA grant or 2) assigned to partners of two universities including the one from which they get their degree, if they do not have any EACEA grant; in this case, they can apply for local grants offered to national students of the country in which they get their degree. For example, at FUB Italian students can apply for provincial grants dedicated to students that obtain their degree in Italy and want to study in Bolzano (see Table 5 for the grants already assigned to EMCSE students). To make the mobility pattern 1) fair for European students, the consortium will work to provide alternative local grants that are comparable to the one offered by EACEA so that student of a country can apply to EMCSE and attend courses at the country in which they take their bachelor degree.
Mobility Rules. EMCSE has few precise consortium rules that regulate students' mobility between the two universities in which they study. Students are instructed about these rules at the first week they arrive at the partner university and they can also retrieve them in the EMCSE web site and EMCSE alumni association on facebook (see section A.4.5)

- To move to the second university, students must complete 52/60 credits by the second exam session of the second semester of the first year (September/October). This would guarantee the second hosting university that students have enough knowledge to attend courses of the second year and helps students to get their MSc degree within two years,
- Second year students receive instruction on the academic and administrative process for preparing and defending their thesis. In particular, there are instructed on when and how to contact their supervisors and co-supervisor. Again, students can find this rule in the EMCSE web site,
- Students are informed on the graduation ceremony that is held at FUB every first week of December. All the EMCSE graduated students (also the one that never visited FUB) and the local EMCSE representatives are invited to the joint ceremony. During the ceremony, a special mention to the EMCSE students is given.

Monitoring mobility process: exchange of students is monitored by the Joint Board and in particular by the local academic representatives in each institution. The representatives inform each other on the students' performance. In particular, during summer and no later than September / October of every year, students' transcript are uploaded at the EMCSE web site private area and discussed by email case by case. During summer, a similar discussion is ongoing for the EMCSE thesis.
Curriculum. To determine the curriculum we have identified a set of common subjects from the existing syllabi of the courses taught in the partner universities. This ensured a solid agreed base on which the teaching offer could be built upon.
EMCSE is a two-year programme (120 ECTS) completely taught in English. The EMCSE course is built on six mobility patterns defined on annual exchange. Students take EMCSE at any two institutions of the consortium (50% of ECTS in each institution).
The curriculum foresees eight different SE topics according to the 2009 ACM Guideline and competence and strengths of the partners. Topics are displayed in Figure 1 and are classified by **core areas, specificities and electives**. Core areas are topics that cover foundations in SE and are taught in any mobility pattern of the programme: Software Process Management, Verification and Validation, Software Design, and Requirements Engineering. Core areas are topics that from a minimum of 4 ECTS in Requirements Engineering to a maximum of 28 ECTS in Software Design across all the possible combinations of student mobility. Specificities are topics peculiar to a student mobility pattern that reflect the local teaching offer and research. As such, they are not taught in all the universities. The only exception is Research Methods that consists in advances in research methods and applications that reflects strengths and peculiarities of the research performed at the partner universities and is taught in all the partner universities. The research methods courses prepare students to elective activities and future research work. Elective activities consist of a project (7,5-8 ECTS) and a master thesis of 30 ECTS. Both are autonomous and supervised activities. Master thesis is co-supervised by two teachers of the partners the student has visited. Students that successfully complete the program get a joint degree from the two universities that have visited in the case of FUB and BTH and get a double degree for a mobility pattern including UKL. UKL is working hard to issue a

joint degree by 2012. Regulations allow all the partners to issue a joint degree; in case of UKL is just a matter of organization of the administrative offices. Namely, at UKL there is already an EMMC funded by EACEA, the European Master's Program in Embedded Computing Systems that issues a joint degree.
Figure 1 illustrates the program structure and the distribution of the topics over the semesters. It is worth noticing that areas do not correspond to courses or modules, but rather to contents provided by the EMCSE programme. The possible learning paths according to the mobility paths and the specific courses at the universities are displayed in the Appendix. In particular, Figure 1 shows the range (minimum – maximum credits) of common knowledge provided in that topic across the curriculum patterns. In the following we briefly describe the eight topics.

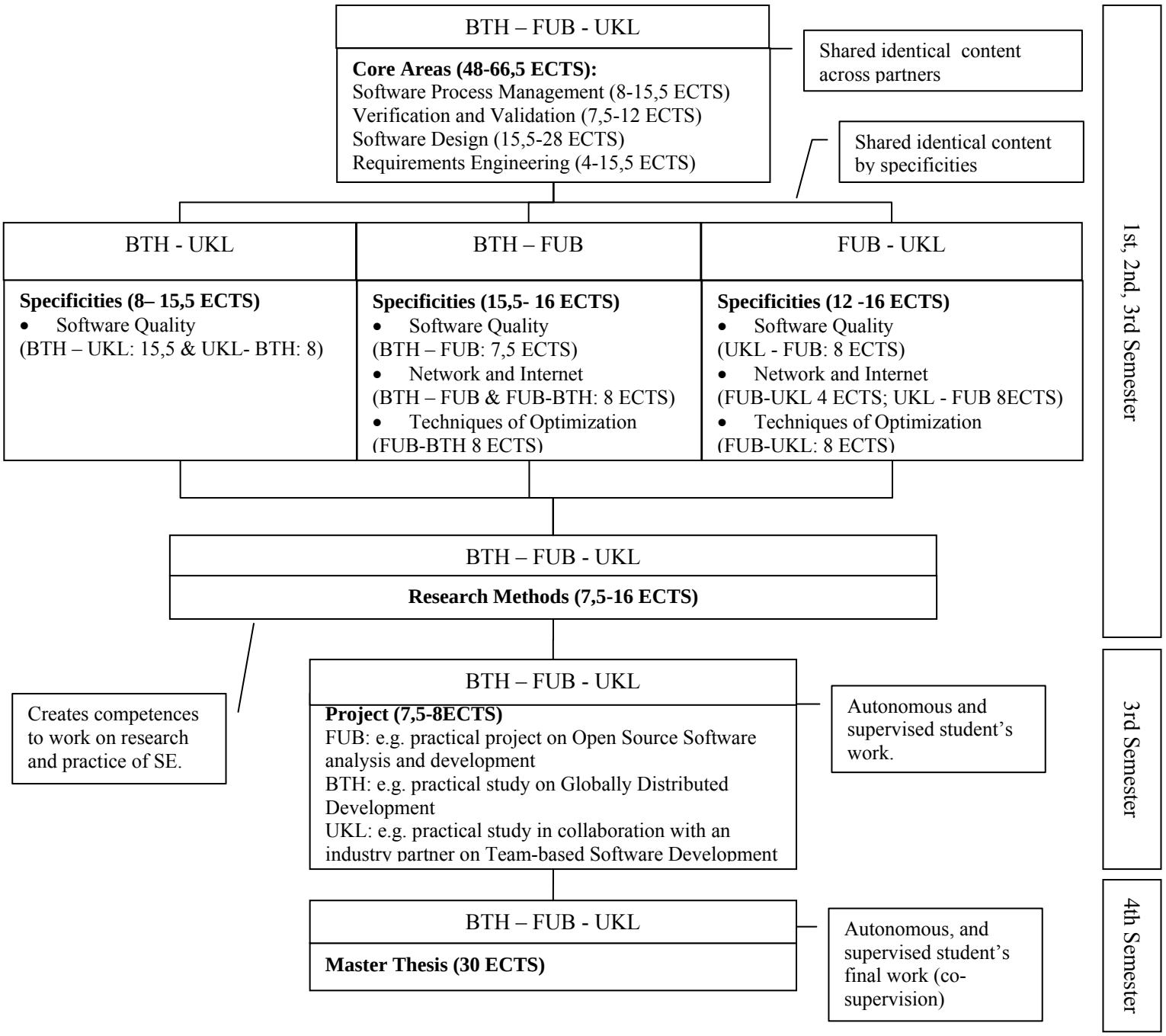

Figure 1: The program structure by topics. Learning paths by mobility track in Appendix

Software Process Management (core area): Software management is concerned with knowledge about the planning, organization and monitoring of all software life-cycle phases. Management is critical for ensuring that software development projects are what the organization is looking for, work in different organizational units is coordinated, software versions and configurations are maintained, resources are available when

necessary, project work is divided appropriately, communication is facilitated, and progress is accurately charted.

Software Quality (Specificity): Software quality is a pervasive concept that affects, and is affected by all aspects of software development, support, revision and maintenance. It encompasses the quality of work products developed and/or modified (both intermediate and deliverable work products). Quality work product attributes include functionality, usability, reliability, safety, security, maintainability, portability, efficiency, performance and availability.

Software Project Management

Verification and Validation (core area): Software verification and validation uses both static and dynamic system testing techniques to ensure that the resulting programme satisfies its specification and that the program as implemented meets stakeholders expectations. Static techniques are concerned with analyzing and checking system representations throughout all stages of the software lifecycle, whereas dynamic techniques involve only the implemented system.

Software Design (core area) Software design introduces students to the architecture, components, interfaces, and other characteristics of a system or component. Students learn to produce various models to implement and evaluate, analyze them against requirements. These models form a kind of blueprint and starting point of construction and testing

Requirements Engineering (core area) Requirements issues are essential for documenting and evaluating user needs. Requirements represent the real-world needs of users, customers and other stakeholders affected by the system. Dealing with requirements includes an analysis of system feasibility, elicitation and analysis of stakeholders needs, the creation of a precise description of what the system should and should not do along with any constraint on its operation and implementation, and the validation of this description or specification by the stakeholders.

Network and Internet (Specificity): Network and Internet module provides a comprehensive knowledge regarding Internet Technologies. It a cross cutting module that provides competence in analysing, designing, developing, and testing standard and mobile systems for the Internet.

Techniques of Optimization (Specificity): optimization methods and techniques for students in computer science provide a wide variety of real models (in economics, engineering, finance) based on advanced mathematical instruments like neural networks. With this competences, students will be able to make prediction on the evolution of software artefacts against the market demand.

Research Methods (Specificity); in both academia and industry, it is often necessary to evaluate different concepts or competing technologies. Applying a rigorous scientific method ensures that the results from an evaluation are reliable. This covers all aspects from writing a research proposal, conducting a literature survey, deciding on an appropriate research methodology for the study (choosing between a multitude of quantitative and qualitative methodologies), designing the research study, gathering and analysing results, understanding the risks and generalisability of the study and the results, and disseminating the results. It also requires an understanding of the ethical issues that govern which studies should or should not be done or should be done differently.

Project and Thesis (Electives): Project is a module in which students performs an autonomous but supervised work on an elective topic. Students integrate theory into practice often performing an accurate literature review in a specific SE area. The project can be performed as student independent work or with advanced courses. Students will develop either a research- or an application-oriented project with the aim of reinforcing some of the knowledge gathered in theory. The project is mainly empirical and covers various aspects of software development. Figure 1 illustrates one representative example of project for each partner. In the final master's thesis, candidates should demonstrate their ability to use SE methods to solve a problem independently. This problem may deal with the construction of a real software system in collaboration with a software company with which the consortium partners have relations or with a knowledge gap in the software engineering body of knowledge. The thesis is supervised by two different supervisor. Each supervisor belongs to one of the universities in which the student took courses. The defence of the MSc Thesis is carried out in the second host institution. The detailed program structure is reported in the Appendix. In each partners, topics are organized in courses distributed across three semesters. They add up to a total of 90 ECTS credit points. In addition, 30 ECTS credit points are assigned to master's thesis and its defence in the fourth semester. This results in a two-year programme of 120 ECTS credit points. Notice that this programme distribution guarantees the Erasmus Mundus programmes requirement related to a minimum of 30% of the credits being offered by each institution in a way that each institution is in a position to be a hosting institution. In fact, in EMCSE all institutions offer 100% of the programme. The distributions of the subjects have limited variations for students visiting the same two universities in opposite order. This would

guarantee homogeneity of the course fundamentals and highlight partner specificity. For example, students that visit UKL – FUB are trained more on foundations on (empirical) research methods in SE than on software design theory and practice. Some flexibility accommodates different interests of individual students or for experimental subtopics introduced from time to time in the curriculum by individual partners (see appendix A for the complete courses descriptions).

Internships are not part of the EMCSE curriculum. However, some students do spend part of the academic course in companies when carrying out their MSc thesis. As we describe later, MSc thesis at EMCSE have a strong research character and, in some cases, that research can be carried out in companies. Examples of this research may be: Identification of best practices, data acquisition and analysis, execution of case studies, etc. So far, companies that have hosted EMCSE students are: Sony Ericsson (private company, Sweden), ST-Ericsson (private company, Sweden), RUAG (formerly known as SAAB Space, private company, Sweden), SAAB Microwave (private company, weden), Fraunhofer IESE (foundation, Germany), Azienda Energetica Trading srl– Etschwerke Trading gmbh (Italy). The conditions of the student-company collaboration are defined on per-case basis, but they are not usually made explicit. The reason lies on the fact that the companies and the Consortium's partners already have higher-level agreements in place (common research projects, for instance), and the students' theses are considered to be included in those agreements. The advantage of this strategy is that all the important issues (for example, intellectual property) have been already defined and all parties (including the student) have secured its rights. Obviously, the fact that the research can be carried out in companies does not mean that companies are readily available. Historically, the relations between the universities and the private sector are complicated, and the current scenario of crisis makes collaboration even more difficult. Not being able to secure enough companies for all EMCSE students is the reason why placements/internships are not an integral part of our curriculum. However, we are aware of the importance of the participation of companies in the master and all our partners are working towards reinforcing the relations with the companies mentioned above and establishing new synergies. For this reason, we have created a new joint working group, the Task Force that includes the associate partners and university representatives that have strong contacts with industry and private research. (see).

A.1.4 Justify the learning outcomes relevance in view of the students' future academic opportunities (e.g. at doctorate level) and employability.

EMCSE provides students with a thorough education comprising both theoretical and practical knowledge that will enable them to tackle and develop software solutions to many problems that they will come up against in their professional or academic career. Specifically, graduated students will be able to master the core body of knowledge in SE and in particular in a specific domain they have deepened in their studies, learn and apply new model (e.g. empirical), techniques, methods, and technologies as they emerge, understand and practice team working and SE stakeholders communication and management. Studying in at least two universities in Europe - with English as the language of instruction, accompanying language classes in another European language - in multi-national groups will contribute to the student's preparation for the increasing globalisation of software commerce and industry.

Learning outcome. A successful learner form the EMCSE programme will be able to:

- Have solid knowledge on foundations, methods, and techniques of requirements analysis and design, software process development, software design, and verification and validation
- Employ the scientific methods of investigation in SE; in particular understanding and using mathematical instruments that support SE and its practice
- Understand and apply principles, structures, and use of SE
- Have basic knowledge of public, corporate and professional culture and society
- Have practice in various software applicative sectors in the socio-economic context including safety critical ones
- Be able to work in large autonomy also managing projects and infrastructures
- Have a command of the SE technical knowledge and skills necessary to practise as software developer
- Have a command of the SE management skills needed to organize and control software project team work
- Be able to reconcile conflicting project objectives providing reasonable cost and time information
- Overcome the challenges of global software development and internationalization

All these skills will prepare students to cope with professional practice that will encompass a wide range of activities, including technical, problem solving, management, ethical and legal issues, written and oral communication abilities and, very importantly, the ability to keep up in a rapidly changing discipline. In sum, students will gain a solid knowledge to be applied as managers of different roles in software companies or software departments in non-software companies.

In particular, EMCSE graduates will mainly found jobs in the "Computer Services" sector (in the NACE classification K72 sector (computer services))[23]. which cover consultancy activities for hardware and software, data processing activities, database activities and the maintenance and repair of office and information technology machinery. In 2007, the 2010 Eurostat report has indicated a significant increase of the sector NACE K72 in terms of the total workforce whose growth rate (2006-2007) of employment reached 6.9 % only second after the Renting sector in the economic area classification.
EMCSE will also establish a Task Force that - with the help of the associate partners - will exploit new frontiers for integrating theory and practice in SE. In particular the Task Force evaluate and tune the programme over the years balancing the industrial and the research content of the courses (e.g. research topics should in most cases be rooted in a real-world problem) and foster academic and professional opportunities. In this Task Force, the associated partner will contributes with the industrial perspective on the major challenges that students may face in their future professional career. Given the internationality characteristic of the associate partners, their role is crucial to gather a cross-country vision of industry in Europe and in the world.
Finally, the course also prepares students for follow-up PhD studies in SE provided by the participating partners and others. In particular, study plans for every mobility track foresees courses in research methods and a project. Research methods course provides students with the instruments to perform research studies whereas a project will help them to apply suitable methodologies to a given scenario. In this way, EMCSE prepares students to their Master thesis first and then to doctoral studies. Specifically, EMCSE prepares students to access to IDSE joint doctoral studies that guarantees an international continuation of the high quality programme provided with EMCSE and an academic opportunity for best EMCSE students. In this context, students will be in contact with the active international research community the partners belong to and will be integrated into ongoing research projects. They will develop competence in foreign languages and international relationships, thereby improving their social skills.

A.1.5 Justify the relevance of the consortium composition and the expertise of the key academic staff involved to achieve the EMMC objectives.

The consortium is compounded of three academic partners and three associate partners. FUB will be the coordinating organization.

FUB, Italy	Coordinator	member of the Joint Board / Task force
BTH, Sweden	academic partner	member of the Joint Board / Task force
UKL, Germany	academic partner	member of the Joint Board / Task force
Fondazione Universita' Bolzano, Italy	associate partner	member of the Task Force
Fraunhofer Institute in Software Engineering, Germany	associate partner	member of the Task Force
Engineering, Italy	associate partner	member of the Task Force

The institutions involved in the EMCSE consortium and the individual researchers and instructors involved in the programme are leading experts in SE worldwide with a long track record in SE, as can be seen from their CVs in the CVs Appendix. By way of an example, papers have been published in the internationally highest ranked SE journals like IEEE Transactions on Software Engineering, IEEE Software, Journal of Systems and Software, Information and Software Technology, Empirical Software Engineering or ACM Software Engineering Notes. As well as, at the internationally highest ranked SE conferences like ICSE, the European Conference on Software Engineering, the International Conference on Software Engineering Research and Practice, the International Symposium on Software Metrics, the International Symposium on Empirical Software Engineering or the International Conference on Requirements Engineering.
The publication by consortium researchers of international books (edited by Kluwer, Springer and MIT press) on various aspects of SE is also noteworthy, as is their participation in several SE-related projects funded by the European Community. The above indicators show that the different professors involved in EMCSE have a sound international reputation as researchers in SE.
The associate partners are well-established organizations with the mission to facilitate the interaction between research and applications in the socio-industrial sector worldwide. See their presentation in the letters of intent attached.

A.1.6 Explain the EMMC interaction with the professional socio-economic/scientific/cultural sectors concerned.

[23] NACE K72. Eurostat report 2010 http://epp.eurostat.ec.europa.eu/cache/ITY_OFFPUB/KS-HA-10-001-06/EN/KS-HA-10-001-06-EN.PDF

As we mentioned all the partners are member of ISERN, the international network that fosters and ground knowledge in empirical research in Software Engineering and scientific member of the ESEM conference series committee. As such, they strongly positioned in the SE scientific area. Each single partner has established long-term collaborations with various actors of the local, national, and international SE sectors
Partners are various on going project with enterprises and public offices (see list of project in the eForm an din the CVs)
Each partner institution has a Career Advisory Service
(FUB, http://www.unibz.it/en/students/internships/default.html) (BTH, http://www.bth.se/eng/current-students.nsf/pages/bth-careercenter) (UKL, http://www.uni-kl.de/wcms/career.html?&L=1)
that will help students in building their own career plans. Career coaching are organized via individual meetings to empower students skills and their vision for their own future. Students are helped to prepare their CV. In addition, in cooperation with the marketing offices, partners organize faculty alumni –industry days where graduated students present their MSc thesis and companies illustrate their profile and meet students. Every partner institution organizes open days, industry days or special events to interact with the socio-economic partners on the territory. For example, FUB participates to the Researchers' night (http://ec.europa.eu/research/researchersnight/) promoted by the Directorate-General for Education and Culture is run in parallel in more than two hundreds research European institutions and where the research artefacts and innovation is exposed to the public (FUB http://www.lunganottedellaricerca.it/).
Besides the traditional channels that the single partner organizations have, EMCSE will establish new channels and synergy through its Task Force (see section 2.1). The Task Force includes the Fraunhofer institute on Software Engineering, and Engineering spa. The first as well known research center and the second as well-established international software company have clear understanding of the dynamics of the socio economic market. As organizations that pursue innovation and research development, they represent the added value of this programme and the strong link with the real world in which students will find their first job. The long experience of the associate partners and the single members of the Task Force (see CVs in appendix) will ensure the quality of the work and achievement of the objectives.

A.2 Course integration (25% of the max. score)

A.2.1 Justify the extent to which the EMMC is organised in a truly integrated way.

As we mentioned, the EMCSE curriculum has been refined during the first three years of the current edition. The proposed new curriculum is structured into five core topics and three curriculum specificities. For each topic and specificity, the Joint Board has identified courses in each partner university that cover the SE foundations according to the expertise of the partners and recent guidelines for a SE MSc curriculum[9]. The courses' syllabus at the partner university have been reviewed and agreed upon the partners. The eight topics included in the curriculum reflect the major strengths of the partners without creating an extra structure as with the modules. If needed, partners can synchronize the local master courses according with the homogeneity and balance structure of Figure 1. In particular, partners have included foundations on Research Method among the curriculum specificities. Students will be espoused and prepared to practices in SE research such as empirical measurement and design, grounded theory, and statistical analysis. Authors as part of ISERN have a large experience in this. With the project, EMCSE students can implement some of the practice as their initial research investigation that will be consolidated in the master thesis. Specificities, project, and thesis are the major factors of specialization among the partners, but students can also find some local additions in the courses related to the core topics. For example, FUB that is strongly oriented to practice includes in its courses the development of Open Source software projects whereas in UKL, practices of SE are specific for embedded systems. Finally, BTH focuses on product lines, architectures, and global development. During the first years of the EMCSE edition, partners have been collaborating for the establishing and consolidation of few but well-defined processes that govern the EMCSE course. This is the major commitment of the EMCSE Joint Board. The board consists of one representative for each of academic partner. The activities of the commission are defined in the Memorandum of Understanding attached to this proposal. Specifically, theses' co-supervision is one of the curriculum related tasks of the board. The process to determine the two supervisors has been defined, agreed, and published at the EMCSE web site. In particular, partners modified local regulations to let the second university supervisor to join the local MSc defence commission. The EMCSE renewal foresees a mixed committee for the thesis defence when issued the joint degree. This ensured the fairness of the evaluation and the accuracy of the grade conversions across the partners. The EMCSE renewal also foresees a new instrument: the Task Force.The Task Force is coordinated by a representative person of all the partners including associated partners (Pekka Abrahamsson and Barbara Russo (FUB), Darja Smite (BTH), Marcus Ciowlkovski (UKL), Stefano de Panfilis (Engineering), Dieter Rombach (Fraunhofer Institute), Moinul Islam (FUB)). Names representative

of the partners have been selected their experience with EMCSE and with industrial environment. The major activities of the Task Force are
1) fund rising 2) EMCSE marketing 3) curriculum synchronization with SE industry, economic, and research demand. The Task Force meets once a year. The Task Force work in strict collaboration with the join commission and commits to explore new opportunities for the development of EMCSE in the three directions we have mentioned. The Joint Board and the Task Force collaborate for sustaining and developing the EMCSE curriculum. In particular, the Task Force suggests to the Joint Board trends and directions for the employability of the EMCSE students whereas the Joint Board report to the Task Force actual learning outcome and statistics on EMCSE students. For the curriculum sustainability, the Task Force's major commitment is to raise funds for the future sustainability of the course. In addition, the Task Force evaluate the curriculum and the students learning outcome in terms of the employability of the students in industry, university, or in suitable areas of the socio economic market. Finally, the organization of the course is founded on the mutual collaboration of the administrative offices. During this first three years this collaboration has evolved defining informal channels among representative clerical servant of the offices. With this renewal, we elicitate the existing administrative collaboration into formal networks of International Relation Offices (IRO), Students Secretariat (SS), and Faculty Secretariat (FS) collaborating with each other. Each network has a representative in the academic partner. FUB will coordinate the administrative works in the consortium, an extra resource will be dedicated to the course. This person will serve as **help desk** and front desk for students, academic, and administrative personnel and coordinate the activities of the administrative offices.
The EMCSE web portal included all the up to date information public and private for students and partners (see section A.3.1). http://emse.fi.upm.es/.

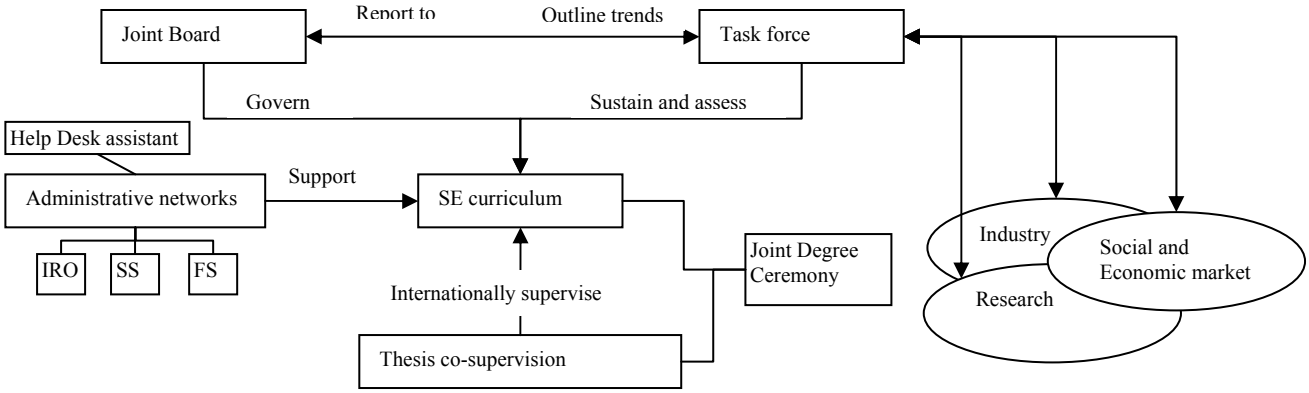

Figure 2: Forms of joint activities in EMCSE

A.2.2 Justify the extent to which the EMMC is recognised in participating countries and leads to the award of an official degree by each of the partner institutions concerned. Describe the type of degree(s) that will be awarded to successful students.

The consortium will issue a joint degree by the end of the first course edition. National regulations allow the establishment of a joint degree. In 2012 the Erasmus Mundus European Master in Computational Logic (EMCL) (in which FUB is partner) will issue the first joint degree at FUB and the Erasmus Mundus European Master's Program in Embedded Computing Systems already issue such a degree at UKL. At the time of this submission UKL cannot officially claim to be able to issue a joint degree for EMCSE and can only officially agree on issuing a double degree, but it is a matter of administrative steps to perform. The joint degree certificate will replace the existing local ones. The allowance to issue a joint degree for unlimited duration according to the local laws[24] is contained in the following documents
(FUB: see annex Proof of recognition_FUB, University Statute art. 26, 2 and ministry decree art. 26, 2 Statute at http://www.unibz.it/en/organisation/organisation/bodies/Documents/st2006-02-20_aus%20Intranet.pdf, see rector approval "Provvedimento Rettore", and President approval "Provvedimento Presidente" in attachment;

[24] In accordance with national laws a HEI that has the right to award a national qualification within a certain subject area and cycle, also have the right to award a joint degree in the same subject area and cycle.

BTH: see annex Proof of recognition_BTH, and http://www.regeringen.se/sb/d/108/a/122185/dictionary/true in Swedish or The Higher Education Act
http://www.hsv.se/lawsandregulations/theswedishhighereducationact.4.5161b99123700c42b07ffe3956.html
UKL: see annex Proof of recognition_UKL)
The joint degree is equivalent to a Master of Science degree in Germany, to a Laurea Magistrale in Informatica in Italy and to a Master i Programvaruteknik in Sweden.
The degree awarded to the students of the EMCSE has been defined based on a multilateral Memoranda of Understanding (MoU) (see Memorandum of Understanding Appendix). The degrees offered by each institution are officially nationally recognised as shown in the proofs attached to this proposal (see Official Proof of Recognition Appendix). All successful students (European and third-country) receive the same final degrees. There is a joint diploma supplement issued by the consortium (see Diploma Supplement Appendix).

Name of institution	Title of degree awarded	Type of degree awarded	Date and reference of formal approval of degree
FUB	Laurea Specialistica in Informatica	joint	5 May 2009 "Decreto MIUR di approvazione della laurea specialistica. Circolare ministeriale dd. 05.05.2009
BTH	Master i Programvaruteknik	joint	2010-11 List of Ordinances by the National Agency of Higher Eduaction and Act on Amendment of the Higher Education Act (2010:701)
UKL	Master of Science	double (joint)	"Hochschulgesetz" (HochSchG), 21.07.2003, §30, Abs. 4, and §19, Abs. 4

Legal approval procedures have been defined based on the MoUs: recognition of a partner institution's programme components and of a partner's study programme as a whole is based on mutually recognizing study achievements on the basis of the learning paths, the project and the master thesis. The following procedure is established in order to guarantee this recognition:

- Each student develops his/her individual study plan jointly with the local tutors and according with the learning path in the Appendix
- The study plans are approved by the Joint Board
- The project and the master thesis of a student are supervised by two professors from two partner institutions
- Each institution provides official proof of examination for each module passed by the student so that these components can be automatically recognised by the other institutions.
- The final defence is performed with a mixed commission (also with a virtual meeting) of the two universities that issue the final degree.

A.2.3 Describe the consortium joint student application, selection and admission procedure.

EMCSE follows formal joint criteria and qualitative joint criteria for admission.

Formal criteria: 1. respect to the application deadline, 2.provide proof of proficiency in English language, 3. have been awarded a BSc on Software Engineering, Computer Science or Computer Engineering, 4. provide certificate of the subjects taken during their degrees, along with the marks obtained and the number of hours/credits devoted to each subject.

Qualitative criteria: 1. language skills, 2. type of entrance qualification (BSc on Software Engineering, Computer Science or Computer Engineering), 3. study results, work experience and professional qualifications, 4. motivation and willingness.

Criteria and document requested are publicly available at
http://emse.grise.upm.es/private/index.php?title=EMSE_private_area:Selection_procedures.

Applicants apply centrally to the Consortium through the consortium web portal
http://emse.grise.upm.es/GEMA/ indicating that they would like to take EMCSE and select they preferred learning path and mobility track. For each learning path, an equal number of available grants is advertised in the web portal. Top ranked students will be offered the learning path according with their rank position, their choice, and the available places. If no more places are available for a given path a top ranked student eligible for a grant will be offered with one of the remaining places in different paths. This guarantees transparency

and fairness of the students' selection and allocation among the partners. Students enrolled locally at the first year at one partner university can opt for the EMCSE program and undertake the below process selection in accordance with local and consortium deadlines. Applicants applying centrally must submit the application by two different deadlines 31st December of every n-1 year and 15th July of year n. Third country students and applicants that applies for an Erasmus Mundus grant must apply by the first deadline. The second is for other European students that do not need long embassy procedure for their visa.
All qualitative criteria are equally weighted except the English proficiency criterion, which is applied in case of equal ranking. Universities ranking (www.webometrics.info) is used to ensure the quality of the selection process. A pre-assessment on applicants' eligibility is done at FUB by a local selection committee, then FUB makes the list and the documents available to the joint board at http://emse.grise.upm.es/GEMA. The Joint Board assesses students also in respect of the local regulations. Finally, it agrees on the final assessment through electronic discussion and vote. Diverging notations are discussed in the selection committee at FUB and the Joint Board. The absence of conflict of interest is guaranteed by making FUB and Joint Board aware of this concern and by multiple assessments. The Joint Board promotes gender balance as preference in case of equal ranking. In case of disadvantaged applicants, the commission use the same principle.

A.2.4 Describe the joint examination methods and mechanisms in place between the consortium partners to assess the students' achievements.

The consortium will follow the current approach. In particular, it will follow the local regulations at the partners. The MSc thesis is the only exception (see below). An equivalence grading scheme has also been defined as shown in the Grading Scheme Appendix. This also means that exams passed at one consortium institution are fully and automatically recognized by the other institutions in the consortium. The overall recognition procedure starts at the end of the first academic year (in Spring semester) at which students exam's marks and transcript are uploaded in the Consortium archives (http://emse.grise.upm.es/private/) and checked (by the joint board) against the mobility rules and the quality of the students' performance. Before the start of the second academic year, transcripts are then transmitted to the second university through the students' secretariat network. Exams marks and credits are automatically inserted in the second university' students database. Second year exams and thesis' assessment are then transmitted to the first year university that automatically recognizes marks and credits and final thesis evaluation. The final essay and the students' documents are uploaded in the private area of Consortium archives. The Joint Board keeps a dossier for each student for Consortium regulation enforcement and reporting purposes.
Thesis' assessment is performed in two different steps: (1) by the two thesis co-supervisors, who write a report and agree on a grade; and (2) by the local defence commission, who finally decides the grade on the basis of the co-supervisors report, the presentation of the thesis and the subsequent discussion with the student. Partners' local regulations allow the second co-supervisor to join the local defence commission.
All the students of EMCSE (also the ones that never visited FUB) participate to the Degree Defence at FUB.

A.2.5 Explain how the students' participation costs to the EMMC have been calculated and agreed upon by the consortium.

The EMCSE study fees are calculated by considering the regular study fee at each partner institution, the insurance costs, and the additional costs arising from the management of the EMCSE master program (Table 3). Notice that both types of EMCSE fees (for European and non-European students) are considerably lower than the maximum EM funded contribution to participation costs. An important input for calculating the EMCSE fee are the costs identified by the consortium during the first editions of EMCSE. In particular, in August 2008, the current coordinator (Politecnica de Madrid) has asked and obtained from EACEA a raise from the original students' participation costs due to the need to cover administrative expenses (see attached report for details). The tuition fee was raised from 3000 to 3800 euro per year for third country students. We do not change this amount in this proposal.
The EMCSE fee cover costs to manage: studying fee at each university (approx. 1300 Euro per year), social taxes (200 euro per year), insurance (240 Euro/semester), equipment costs, managing the communication inside the consortium and with the students, issue consortium certificates, charge tuition fees, maintain the programme's on-line database, and the web portal. Each partner also takes care of: tracking students from admission to arrival to the university, supporting them during visa application, enrolling students in courses, and interchanging transcripts for credit recognition.

Table 1: Distribution of the major tasks among partners that motivate the costs

Task	Performed by	Reviewed/approved by
Student admission/selection	FUB	BTH; UKL
Student support (visa, housing, language courses, etc.)	BTH; FUB; UKL	
Scholarship management and payment	UKL; FUB	BTH
Quality assessment	BTH; FUB; UKL	
Reporting (progress, 2nd pre-financing, final)	FUB	BTH; UKL
Degree issuance	BTH; FUB; UKL	
Diploma Supplement issuance	FUB	BTH; UKL

Participation costs have been calculated according to the expenses incurred by participating institutions to provide the educational services to students, and according to national regulation in the three countries. Table 3 illustrates the case of 45 students enrolled at EMCSE.

The larger part of costs will be encountered by FUB that will have to employ an additional person to manage the course (Table 2). For covering the costs of the EMCSE administrative assistant, the coordinator keeps 10% of all the EMCSE students' fee. This amount is not enough to cover the cost of this person, but it will be an additional motivation that challenges the consortium to recruit new students for the sustainability of the programme.

In case of no financial funds from EACEA and in the spirit of co-financing, **FUB will cover this extra dedicated position with its internal funds (see attachment "Provvedimento Presidente" in Italian).** In case of EACEA financial support, FUB will have 19,680+5,000 euro = 24,680 euro to cover the expenses for the administrative assistant.

Table 2: Figures on costs and tuition fees for45 students (in euro)

Students	Third countries	Other students
Tuition fee **per semester**	1,600	1,125

Tuition fees per partner **per year**	**CO (FUB)**	**BTH**	**UKL**
Third countries students	8	8	8
other students	7	7	7
Total Third countries tuition fees	30,400	30,400	30,400
Total Other students tuition fees	15,750	15,750	15,750
Total	46,150	46,150	46,150

% coordination 10%		4,615	4,615
Total transferred to partner	55,380 (= 46,150+4,150+4,150)	41,535	41,535

Expenses per partner	**CO (FUB)**	BTH	UKl
Local tuition fees and social taxes (750 euro per student per semester [25])	22,500	165,000[25]	22,500
Students' insurance (240 euro per student per semester)	7,200	7,200	7,200
Travels, web portal, software for students selection and managements, students events (mainly FUB)	6000	1000	1000
Total expenses	35,700	173,200	30,700

Balance to cover administrative tasks (per year)	19,680	negative value	10,835

In case of EACEA funding (15,000/year flat rate)[26]	**FUB**	**BTH**	**UKL**
	5,000	5,000	5,000

In Table 3, we have considered an average tuition fee of 1,500 euro per partner although it is worth noticing that the regular study fees for a master student in our consortium differ, For example, BTH has more than 11,000 euro per year for non-European students whereas FUB has about 1,500 euro. Nevertheless, the EMCSE study fee will be uniform in all three partners for the students and will vary only between European and non-European type of the student. In the case of fees exceeding the EMCSE tuition fee, the partner will waive the reminder. This will ensure students to be treated fairly within the consortium. The tuition fee of EMCSE is 1,125 €euro for European students and 1,900 euro for third-country students per semester. Third countries students' participation fees are higher because the consortium must provide additional service to those students as for example the support for visa applications, guidance with other legal aspects of their staying in Europe, etc. Based on our experience from previous years with the EMCSE course, the work needed for EU students is considerably lower and therefore a lower participation fee reflects this difference.

A.3 Course management, visibility and sustainability measures (20 % of the max. score)

A.3.1 Describe the organisation of the cooperation mechanisms within the consortium.

Academic organization: The EMCSE academic organization and management is based on three main pillars: de-centralization, commons rules, and confidence.

De-centralization: Partners deliver their own courses and uses their own examination and quality assurance procedures. This makes the EMCSE easier to implement. EMCSE students are treated as any other student in the partner universities.

Common rules: Students are required to follow some consortium's rules (i.e.: pass 52 ECTS the 1st year) to keep attending the master course and to defend the thesis. Rules are clear and published on the EMCSE web

[25] Except BTH that has a new local tuition fee of 11,000 euro per year

[26] In case of EC financial contribution the 15,000 Euros lump sum will be divided equally among the partners.

site. Every year the new incoming students meet the local coordinator that instructs new students on the consortium's and local organization and rules. It is obvious that the common program limits the freedom of each partner to drop or modify the EMCSE master courses. On the other side, EMCSE rules have an integration effect. For instance, the 52 ECTS rule aligns the strict permanence rules of German universities with the lesser rules of the other partners.

Confidence: The partners know each other for a long time, because they have cooperated in other activities (common research project, organization of events, etc.). This makes unnecessary to perform controls (i.e.: type and number of credits taken by a student) otherwise required. The existing confidence makes possible the transparent credit recognition and diploma delivery. In addition, a shared wiki and archives (http://emse.grise.upm.es/private/) allows any partner to access any time the documents and the status of each single student enrolled in the program.

Management: A committee composed by the partners' representatives, plus the EMCSE coordinator, performs the management of the EMCSE. Major decisions are made during the two annual meetings (June and December). Day-to-day decisions are made after email discussions among the committee members. To avoid dependence on individuals (either from the management committee or the officials of the partner universities) the Consortium has created a set of regulations, which deal with all relevant management aspects. Such regulations are available to students and academic and administrative members of the consortium at:

http://emse.grise.upm.es/private/index.php/EMSE_private_area:Consortium%27s_regulations.

If needed, consortium regulations have been backed up by the legal representatives of the partner universities after the signature of the new Consortium Agreement.

Infrastructure organization: The EMCSE web portal included all the up to date information public and private for students and partners. A private area allows the exchange of information and the storage of document of sensible data. With the renewal, the EMCSE web site has not changed to facilitate the move to the new EMCSE five years edition. As for the change of the coordinator EMCSE web portal will be moved to the address http://emse.inf.unibz.it/ that will host the EMCSE portal in the future. At the moment, it simply redirects to current one (http://emse.fi.upm.es/). In the following, we will mention the URL addresses of the current one for the sake of the reviewers that can visit them.

Administrative networks. The three institutions in the consortium have staff available for those tasks performed locally. To reach a higher efficiency in these tasks too, the consortium will establish networks of offices involved in the master programme from all three partners. Each network will mainly join the three similar offices. As such, the consortium will have a network of international offices, a network of student secretariats, and a network of faculty secretariats. The FUB office will coordinate the respective network in organizing document exchange, schedule, and administrative tuning in respect of the local regulations and independence of the partner's offices. See also section A.2.1. Locally, the tasks performed by the offices will be

- FUB provides administrative support for teaching activities both at university level, through the Student Secretariat, the International Office, and at the Faculty office, which employs four administrative workers just for the BSc and the MSc degrees in CS. FUB offices will coordinate the administrative tasks at consortium level distributing the relevant information to partners. The Help desk assistant will define schedules and procedures among the partners' offices.
- BTH provides different kinds of support for international students, at university level the International Office is responsible for arrivals, introductions, visas, housing, etc. (two people are specifically working on the masters programmes). At department level, one administrative worker is allocated to Student Help Desk, another is allocated to International Student Support (including bridging to International Office) and one of the teachers is allocated as a local Programme Manager for the EMSCE. In addition, administrative and technical staff from the School of Engineering is providing EMCSE programme teachers with support for course administrative tasks, scheduling, user logins, course management systems, etc.
- UKL has four administrative staff at the student secretariat to manage student and teaching support for the Masters curriculum in the Computer Sciences department. Additionally, the general administrative and technical staff of the university and the department provides support for instructors.

All the partners are running an EMMC and have already expertise in the administrative tasks involved in this kind of programmes.

A.3.2 Provide information on the partner institutions' financial contribution to the EMMC and describe the way the EMMC will be managed from a financial point of view.

Partners are already running the master courses on which the EMCSE programme is grounded, therefore courses guarantee a local sustainability. Partners provide all the necessary services for master students as the international offices, the studying support, the equipment, and the library. These services will be partially covered by the EMCSE fee and partially by each partner. The key costs covered by the EMCSE fees will be dedicated to pay each partner's studying fee that considerably varies across partners. In particular, BTH that traditionally did not have high tuition fees, since 2010/2011 will have an annual local tuition fee of more than 11,000 euro for NON-EU students, way above the EMCSE consortium fee and the maximal contribution of EACEA to the Erasmus Mundus program. **BTH will waive the reminder of tuition fee for the EMCSE students**. Table 3 illustrates the details on costs and tuition fees.
The coordinator will transfer the EMCSE fees (minus the 10%) to the partners and used by each partner to cover costs incurred with the local management of EMCSE. As students' fees will not be enough to cover these costs, partners will use local funds and third parties support for grants and exceeding costs (see next tables).
In the principle of co-financing and self-sustainability EMCSE will exploit sources of financing that have been successful in the first edition of EMCSE. Past EMCSE students have beneficed of grants of local organizations (Table 4), EMCSE will exploit the same sources in the future (Table 5).

Table 3: EMCSE Students granted from other sources than EACEA (source, number of students, total grant amount)

Funding agency	Description
Province of Bolzano, Italy (students that visited FUB)	European students. 2007-2011, 21 students granted total grant = 117,600
Lifelong Learning Program Erasmus (LLP) + FUB contribution (students that visited FUB the first year)[27]	2007-2010, 10 students for 12 months each, total grant = 41,760 Euros
FUB internal funds	2010/2012, 4 students, about 20,000
All partners	Language courses free of charges for European Master students

Financial sustainability of the program will be performed in two ways: with partner actions and consortium actions.
Partner actions are those activities for fund raising, marketing, and graduate employment that a single partner does locally or at a national level. Every partner has its own sources. Table 6 illustrates those programmes that have and will support EMCSE in future.

Table 4: Programs that will finance grants for EMCSE students

National or local grants that can be used by EMCSE students (based on previous experience)
Bolzano Province, 5800 euro per student per year[28]
LLP Erasmus + FUB contribution (FUB)
LLP Erasmus (all partners)
Language courses free of charges for European Master students

Consortium actions are those activities performed by the consortium through the Task Force. The Task Force aims at the same objectives of the single partner, but have a larger target application field.
For example, at the moment all the partners have signed LLP bilateral agreement in the Erasmus framework to ensure little grants to students that do not have any grant and move to the second university according their learning path (consortium action). In addition, FUB provides extra monthly money to students granted by Erasmus programme (single partner action)

Management. The Joint Board will dispose funds of the consortium according to the Memorandum of Understanding signed by all partners (including those raised by the Task Force). The coordinator will receive funding and it will distribute them to partners according to the number of students studying each year at a specific partner. FUB will administer centrally students' fees and EACEA grants for students in case of funding. FUB will pay monthly students on their European account as described by the student's contract

[27] For any Erasmus Student that goes abroad FUB adds extra of 130 Euros/month on top of the EC contribution. EMCSE students have the priority on all the other students for the two partner universities.
[28] http://www.unibz.it/en/prospective/housingfa/funding/grantsprovbz.html

(see below and attachment), pay the students insurance, and transferring remaining money to partners according with the students' learning path. Partners that receive a student will use the money to cover local expenses for that student (as described in Table 3 of section 2.5).

Grant A student:

In year 1: After the agreement has been signed, the student shall receive a lump sum of 4,000 EUR. For this purpose, the student shall have informed EMCSE of the details of a bank account opened in his or her name in the European country where he or she is taking the course. The student is responsible for supplying the correct details as and when required.

The other payments shall be drawn as ten monthly payments, from October to July. The payments shall be made in compliance with the usual practice of the FUB controlling office. The student shall receive 12,000 / 10 = 1,200.00 EUR in each monthly payment. If, for any reason, a payment cannot be made, the respective arrears shall be due in the following payment.

In year 2, the payments shall be drawn in 10 monthly payments under the same conditions as specified before. The amount of each monthly payment shall be 16,000 / 10 = 1,600.00 EUR.

In total, the student will receive 32,000 euro.

Money transfer costs will be charged to the student if not covered by the tuition fee.

On top of this amount, the consortium will pay the tuition fee (3800 euro/year) and health insurance (480 euro/year), insurance agency chosen to fulfil EACEA's requirements0

The EACEA total contribution per student in two years would then be: 32,000+7600+960 = 44,560 euro

Grant B student:

The student shall receive the grant in accordance with the following schedule:

The yearly grant of 6.000 EUR shall be drawn as 10 monthly payments, from October to July. The payments shall be made in compliance with the usual practice of FUB controlling office. The student shall receive 6,000 / 10 = 600.00 EUR in each monthly payment. If, for any reason, a payment cannot be made, the respective arrears shall be due in the following payment.

On top of this amount, the consortium will pay the tuition fee (2250 euro/year) and health insurance (480 euro/year), insurance agency chosen to fulfil EACEA's requirements. Money transfer costs will be charged to the student if not covered by the tuition fee.

The EACEA total contribution per student in two years would then be: 12,000+4500+960 = 17,460 euro

Scholar:

Scholarship holders will be paid locally by the visiting partner institution. FUB will transfer the money prior to the visit of a scholar to an account of the visiting partner institution. Money transfer costs will be charged to the scholarship holder.

Students' local costs and social taxes will be managed and paid by each single partner. Transfer and payments will be done as soon as the money are available and in respect of the local financial and organizational rules

Partners agreed to sustain the first edition of the course with grants and administrative funding:

Start up grants. In view of the co-financing principle (with or without the EACEA funding) and to ensure the success of EMCSE, the Fondazione Libera Università' di Bolzano will fund 4 grants of 3800 euro each per year for the first edition of the course, 2012-2014 , For the same period BTH, and UKL, will fund from 3 to 5 grants of 3800 Euro and FUB 2 grants of 2250 Euro. (Table 6).

Table 5: Grants that will be covered by internal funds of the partners 2012/2014

FUB	Fondazione	BTH	UKL
grants covering 4 tuition fees for two years (2*2250 euro per year)	grants covering 4 tuition fees for two years(4*3800 euro per year)	grants covering 3-5 tuition fees for two years (3800*3 – 3800*5 euro per year)	grants covering 3-5 tuition fees for two years (3800*3 – 3800*5 euro per year)
Total: 9,000 Euro	Total: 30,400 Euro	Total: 22,800 Euro – 38,000	Total: 22,800 Euro – 38,000

FUB administrative support. In view of the co-financing principle, FUB, to sustain EMCSE in case of no financial support from the EACEA will fund a position of project manager for the administrative assistance (see attached document "Provvedimento Presidente" in Italian and German). In case of EACEA funding the position will be covered as described in Table 3 (again with a co-financing principle).

Finally, teacher mobility is a key component in the sustainability of the EMCSE programme. To that end, EMCSE might make use of a further source: the CEEPUS network. CEEPUS (Central European Exchange

Program for University Studies – http://ceepus.info) is a well established organization supporting teachers and students mobility. Recently CEEPUS also supports implementation of joint studies.

.

A.3.3 Describe the consortium development and sustainability plan designed to ensure the proper implementation and continuity of the EMMC beyond the period of Community funding

In its renewal EMCSE has established a Task Force (see section A2.1) that will be responsible to identify strategies and tactics for the development and sustainability of the programme. Members of the Task Force are the representatives of the academic partners and associated partners. In the annual meeting, the Task Force will set up short term strategies and revise long terms one. It is worth noticing that the course is self-sustainable in terms of didactics and infrastructure as it is completely based on existing courses, local students / scholar support, and facilities. The major overhead of the programme is coordination at consortium level and administrative /didactic programme synchronization and maintenance at the local level. The first sources to cover these costs are tuition fees. The higher the number of students enrolled the higher is the sustainability of the EMCSE programme. Therefore attracting a large number of high quality students is of paramount importance. **Attracting students is a matter of quality of the programme, infrastructure, research offer, availability of grants or scholarships, affordable tuition fees, and employment opportunities**. The high quality of the programme is ensured with the activities described in section A.5 and the activities of the Task Force and the Joint Board in their respective fields of action (see section A.2.1). Infrastructure and facilities are guaranteed by the high standards regularly supplied to students of the three academic partners.

The tuition fees of the consortium are relatively low and partners foresee to keep them low as much as it is possible. The major strategy is then the establishment of regular channels that provide grants and scholarship to students and scholars and the employability of the EMCSE students. Long terms strategies will focus on further development of the curriculum and extension of the consortium.

Tactics need to implement these aspects.

Mid terms tactics include:

- Inviting scholar knowledgeable in SE and announce their visit in the web site page.
- Offering internships in the private and public organizations. This will help students' employability
- Keep the amount of the tuition fee limited and increase the number of the students
- Promoting EMCSE through ISERN research network. This might attract high quality scholars

Long terms tactics include:

- Consortium extension to one third-countries academic partner. Given the amount of requests coming from applicants in the Asian area, an Asian partner would promote the programme in this area.
- Consortium extension to new associated partners. This would give better chances of students' employability.
- Funds finding for further development of the programme (e.g. EU programmes like TEMPUS)
- Funds finding for teachers' mobility (e.g. the EU CEEPUS programme)
- Funds finding for students recruitment (e.g.

A.3.4 Describe the course promotion measures taken by the consortium to increase the course's (and the EM programme's) visibility and attractiveness.

Each partner has a press office / marketing office that will advertise the course through the regular channels like national and international fairs.

EMCSE experience demonstrates that Internet is an effective way to promote the course worldwide. As such besides the EMCSE official web portal, EMCSE has

- Three dedicated local pages:

FUB: http://www.case.unibz.it/index.php/EMSE/EMSE.html,
BTH: http://www.bth.se/tek/masters_eng.nsf/pages/c708eea0825c3fb2c1257230005da17c!OpenDocument,
UKL: http://www.bth.se/emse

- A wikipedia page
- A facebook page (see section A.4.1)

As we mentioned the EMCSE course is also advertised during the open day and the night of research (see section A1.6).

Finally, one of the most effective measures of promotion will be the EMCSE alumni. They represent a valuable resource for the visibility of the EMCSE course. EMCSE alumni have already undertaken different careers and we are still in contact with many of them. In appendix, we list the graduate EMCSE students with whom EMCSE partners are in contact and their actual employment.

A.4 Students' services and facilities (15% of the max. score)

A.4.1 Describe the nature of the information (/support) provided to students prior to their enrolment and the way this information will be delivered.

A full description of EMCSE is given on the Internet, listing, among other things, general information, detailed information about student mobility and study plans, specific information about lectures including teaching material, information about grants, admission criteria, deadlines, and instructions for applicants at http://emse.fi.upm.es/ and at each EMCSE local pages of the partner (that include partner profiles and research). See section A.3.4 for local page addresses.

On-line application is available at the web site http://emse.grise.upm.es/GEMA/

Our experience demonstrates that Internet is the most effective way to reach third-country prospective students. As such, we have created a Wikipedia page,

http://en.wikipedia.org/wiki/European_Master_on_Software_Engineering

and a facebook one, http://www.facebook.com/emse.eu.

For EU students, we have to differentiate between applicants looking for a (B-type) grant and applicants looking for a study program (that is, students who consider that obtaining a grant is a quite unlikely event). In the first case, Internet is again the most effective way to publicize the EMCSE. In the second case, the promotion policy must be different. Even if the web is the place where students contact EMCSE contact points for the first time, the wide selection of master degrees and the limited availability of funding at the national level make those students more selective and harder to attract. We apply different strategies with those students:

- Advertising at Partners' neighbouring universities. We distribute posters, flyers and when possible we organize informative sessions for prospective students
- Exploit contacts at other EU universities, using essentially the same strategy (posters and flyers)
- Once we contact a prospective student, we keep in touch with him/her by email/phone or in person when possible
- EMCSE students have been invited to talk at university events. For example, at FUB EMCSE students have presented their experience at the open day, opening of the academic year, and degree ceremony. This would raise the attention on EMCSE of local undergraduate students
- Personal contacts

A.4.2 Describe the content (and, if available, provide a model) of the Student Agreement defining the rights and obligations of the two signing parties.

There are two Student Agreements in force at EMCSE. They are both included in the Appendix.

A.4.3 Present the services that will be provided by the partner institutions to host students / scholars, including the nature and coverage of the mandatory insurance scheme.

The services provided by the Masters consortium start during the application of a student or scholar to EMCSE. FUB as coordinator provides general counselling on all issues related to studying and teaching on our EMCSE and about moving from a third - country to the European Union. The EMCSE administrative assistant will also act as a contact point for all students and scholars whenever any problems occur either before, during or after the study period or stay in Europe. In addition, the International Relation Office (IRO) at FUB will coordinate communication and exchange with the partners' IROs. The IROs provide support with accommodation, travel, boarding, visas, funding, local support, payment of fees at the university or institution where the student is to study, local information, etc. Such international offices have been working for long time providing support to international students (many of them related to the Socrates – Erasmus and Leonardo programmes). As an example, in 2009/2010 two married EMCSE students from Sri Lanka got a baby before arriving at their first university (FUB). They were not able to move the baby with them. Despite the enormous resistance the IRO found, it was able to join the family within the first year.

Information about are

at FUB http://www.unibz.it/en/prospective/exchange/Incoming/ErasmusMundus.html, at BTH

http://www.bth.se/exr/intoffen.nsf/, at UKL http://www.uni-kl.de/wcms/aaa.html.

Faculty secretariat of the partner institutions will be joined in a network. Their services concerns didactic, IT facilities (free wireless at the dormitory too) student cards, and health insurance. In the last case, the insurance

In addition, each partner institution ensures that all administrative contact persons necessary for a successful study experience are fluent in English. These include the local organizers of the study programme, the international office, the computer centre, the library, the examination office, etc.

In addition, each partner institution will ensure that there is sufficient housing for EMCSE students and scholars. Furthermore, language and culture courses in the local language are offered at each site, and will be

available for all EMCSE students and scholars. Partner institutions provide services like free babysitting services, day nursery, special needs services and equal opportunities committee.

A.4.4 Describe the consortium language policy.

The EMCSE language is English. Courses, administrative and academic communication is in English.

At FUB all the MSc in computer science courses are regularly taught in English by international teachers. Exams and theses are in English as well. The language centre at FUB offers intensive and regular courses in Italian, English and German free of charge for European Master students (Italian, German are the two local spoken languages). In Bolzano- Bozen, students have the chance to practice the two local languages.

At BTH, all research and teaching staff in the SE field are used to speaking and teaching in English because the local master programme in SE uses English as the language of instruction since 1995. It is worth noting, that the teaching staff is highly multi-national. The international students are also offered to engage in the free of charge Swedish language courses, if interested.

At UKL, the existing Bachelor and Masters-level programmes already guarantee that a sufficient number of courses will be offered in English to allow international students to gain the Bachelor or Masters degree. In addition, teaching staff have strong international ties. Visiting Professors include researchers from Japan, Brazil, USA and Korea. Introductory compulsory German courses are offered free of charge for EMCSE students.

A.4.5 Indicate the measures taken to facilitate networking among the Erasmus Mundus students and between these students and other students from the partner institutions.

There are five major activity that aim at integrate students also beyond the study itself.

1. Local student associations organise different social activities (parties, trips, sport activities, etc.) aiming at integrating international students.

2. The EMCSE Alumni Association of the Consortium organizes a yearly meeting of all EMCSE students (WEMSE) at one of the partner institutions

(see http://www.emse-students.org/wemse_2009/ and http://emse- students.org/wemse_2010/)

where students can report on their experience, can learn about the best practical assignments, selected students can present the results of their project or master thesis, selected scholars and staff members present their latest research results, and students their culture and experience. As a secondary activity, students discuss with the Consortium members and provide feedback about their initial expectations regarding the EMCSE and the degree of fulfilment after 1-2 years of study. Questionnaires for QA are also distributed and gathered during this event, complementing the partners' QA systems. This event helped with the launch and continued support of an alumni organization.

3. The graduation ceremony is held at FUB on December. EMCSE students of all the partners participate to the ceremony. The initiative has been appreciated as students participated to the ceremony at their own expenses.

4. EMCSE Alumni have a facebook site http://www.facebook.com/emse.eu where they keep being in contact also after graduation.

5. EMCSE students have served as students volunteers at local conferences. This activity helps them interacting with other young people that attend the conferences. For example, in 2010 EMCSE students served at the ESEM 2010 conference in Bolzano, Italy and at ESEM 2008 conference in Kaiserslautern, Germany.

A.5 Quality assurance and evaluation (10 % of the max. score)

A.5.1 Describe the internal evaluation strategy and mechanisms in place.

ECTS mechanisms is a consolidated mechanism in the consortium and in all institutions in the consortium having grading system conversion to ECTS or adopting the ECTS system itself (like BTH). This guarantees the homogeneity and transparency of the grades in particular to students that are not familiar with the European system. The quality assessment mechanisms within the EMCSE consortium is based on two instruments: the local questionnaires provided by the single partner university to all its students and the EMCSE questionnaire that. Both are based on blind questionnaires. The local questionnaire are administrated at the end of each course. They are anonymous and supplied to students without the presence of the instructor. Statistics on the single course are provided to local Master councils that examines and take actions on the results of the answers. Issues related to EMCSE courses are reported to the local representative of the EMCSE Joint Board. Such questionnaires address issues related to both the abilities of the instructor (communication skills, student participation allowance, teaching strategy, …), as well as, characteristics of the subject taught (relevance, duration, student learning effort, …). The EMCSE anonymous questionnaire is provided on-line by the partner BTH at http://bth.webbform.se/. The questionnaire provided by BTH has been discussed and agreed upon the partners. The results of the questionnaire are directly provided to the

Join Board. Namely, the Joint Board's tasks for quality assurance include: analysis of students questionnaires; definition of recommendations for quality assurance and improvement based on students input; definition of recommendations for quality improvement based on leading experts evaluation reports. Additionally, at the end of the EMCSE, as well as, one year after graduation, students are asked to fill in other questionnaires to gather information about their overall impression of the EMCSE and, in the latter case, to gather data about how useful the individual subjects are in their professional life and the kind of jobs that they get after graduation. Such final questionnaires will also be processed by the EMCSE Joint Board. The last but not less relevant source of information for quality assurance is the Alumni Association and the local representatives of the EMCSE students that act as reference for any issue related to the single edition of the course or to the overall EMCSE evolution. Reports on questionnaires will be reported to the EMCSE Task Force. If needed Alumni representative can join the Joint Board meetings to report students' opinion on EMCSE quality. At the annual workshop WEMSE, students and teachers representatives will have interactive discussions on the students' satisfaction of the EMCSE program.

A.5.2 Describe the external quality assurance envisaged.

EMCSE partners (will) undergo to periodic international evaluations. In particular,

- Periodic Official accreditation, of each of the national course programmes of which the EMCSE is composed. This accreditation guarantees a quality control by the national authorities of the institution in each country. Quality inspections are performed regularly to study programmes and subjects by the national quality agencies. Periodicity depends on national laws. For example, FUB undergoes to GRIN certification www.grin-informatica.it/ every year.
- Periodic independent official international accreditation organised by the institutions for didactic and infrastructure quality. Partners will organise independently international evaluations according with their usual quality assurance system. For example, in 2010/2011 FUB underwent to the evaluation of the Austrian Quality Agency AQA, http://www.aqa.ac.at/). The peculiarity of this evaluation is that besides the quality analysis of infrastructure, research, and teaching offer, AQA organised workshops to solve open issues surfaced in the independent reports of the different bodies and offices of the university and to plan future actions based on multilateral expectations.
- Expert evaluation of the EMCSE by leading experts in the world that periodically will asses in-situ our integrated course and report particular recommendations for improvement. Partners have experience in external expert evaluation. For example, the Faculty of Computer Science of FUB underwent to expert academic evaluation in 2006. In this case, representatives of academic bodies, students and administrative staff have been interviewed individually. Three international experts have been selected on the relevance of their research in computer science and their competence in didactic. Interviews were preceded by expert analysis of the documentation and the periodic quality reports available at the faculty. The EMCSE Joint Board will organise / integrate a biannual expert evaluation either exploiting the local ones or using an independent pool of international experts.
- The EMCSE Task Force will be in charge to monitor the quality of the learning outcome in terms of the market and research objectives and demand. As such, the Task force might call for an external evaluation made by practitioners in the field of Software Engineering or participate as member of the biannual expert evaluation.
- National ranking of the Universities in the Computer Science area (where available). For example, FUB undergoes to CIVR ranking https://civr.cineca.it/ every three years.

Index of Appendices and annexes

Partners' EMCSE web pages
Students Graduated at EMCSE and their employment
Study Plans and mobility Patterns
Courses Syllabi
Grading Scheme
Staff and students population in EMCSE
Associated Partners
Short bios of the local coordinators

Relevant annexes:

LOI_FUB
LOI_BTH
LOI_UKL
LOI_Fondazione
LOI_Engineering
LOI_Fraunhofer
Example of Diploma supplement
Student's contract Grant A
Student/s Contract Grant B
Official Proof of Recognition (FUB, BTH, UKL)
Multilateral Memorandum of Understanding
CV Giancarlo Succi
CV Claes Wholin
CV Dieter Rombach
CV Pekka Abrahamsson
Joint Degree Template
Provvedimento Presidente
Provvedimento Rettore
Raise of Tuition fee (report 2008)

Appendix
Partners' EMCSE web pages
http://www.case.unibz.it/index.php/EMSE/EMSE.html
http://www.informatik.uni-kl.de/studium/studiengaenge/erasmus-mundus/
http://www.bth.se/tek/masters_eng.nsf/pages/cf889efe637ef964c12573a100465ab0!OpenDocument

Some students graduated at EMCSE and their employment

Student name	Nationality	job
Jessica Rodriguez	Colombian	private company in Bozen, Italy
Josef Hardi	Indonesian	temporary research contract at FUB, Italy
Stefan Meierhofer	Italian	Doctorate at the University of Nottingham. UK
Michael Unterkalmsteiner	Italian	Doctorate at BTH, Sweden
Hannes Tribus	Italian	Private company in Bozen, Italy
Irene Moriggl	Italian	High school teacher in Bozen, Italy
Sufian Md. Abu	Pakistani	Softer tester at Euroavoinics Navigation Systems, Pforzheim, Geermany
Nattakarn Phaphoon	Indonesian	Doctorate at FUB
Loan Nguyen	Vietnamese	will do Doctorate at FUB
Taslim Arif	Bangladeshi	Employed at the Fraunhofer Institute, Kaiserslautern, Germany
A.K.M. Moinul Islam	Bangladeshi	Research Scientist in University of Kaiserslautern, Germany
Rahmat Bagas Santoso,	Indonesian	Information Analyst in Atos Worldine, Frankfurt, Germany
Himanshu Saxena,	Indian	- Software Engineer in Insiders Technologies, Kaiserslautern, Germany
Mohammed Omer,	Eritrean	- Software Developer in LHS AG, Frankfurt, Germany
Hafizul Anwar Raduan,	Malaysian	- Software Developer in Bosch Access Control Systems, Aachen, Germany
Cheng Chow Kian	Malaysian	Software Engineer in GE Healthcare, Freiburg, Germany
Lubomir Karlik, -	Slovak Republic	IT consultant in ZEB/ROLFES Schiererbeck Associates
Rahadian Bayu Permadi,	Indonesian	- Software Engineer in Amadeus S. A. S., France
Jan Schulte,	German	- Analyst in Banco Santander, Spain
Wan Ai Goh,	Malaysian	Software Engineer in Insiders Technologies, Kaiserslautern, Germany
Qifei Lu	Chinese	Software Developer in Markit Financial Information Services, Frankfurt, Germany
Anh Nguyen Duc,	Vietnamese	- PhD Student in Norwegian University of Science and Technology (NTNU), Trondheim, Norway.
Nha Vi Tran Ngoc	Vietnamese	- Software Developer in CASIS International, Singapore
Vigo Martin	Spanish	Software Engineer in Apple Inc
Arturo Gomez del Castillo		Rich Media Technology Specialist at Google London, United Kingdom

Study plans and mobility patterns

Mobility tracks

First year	Second year
FUB	BTH
BTH	FUB
FUB	UKL
UKL	FUB
BTH	UKL
UKL	FUB

Legend for study plans

Topic	Table Label
Software Quality	Q
Verification and Validation	VV
Software Process Management	SP
Software Design	D
Techniques of Optimization	O
Research Methods	RM
Requirements Engineering	RE
Project	P

	UKL		BTH		UKL		FUB		FUB		BTH	
1 YEAR	VV: Safety and Reliability of Embedded Systems	4	SP: Advance Software Project Management	7.5	VV: Safety and Reliability of Embedded Systems	4	SP: Software Process Management	8	SP: Software Process Management	8	SP: Advance Software Project Management	7.5
	Q: Performance Modelling of Distributed Systems[29]	4	SP: Applied Software Project Management	7.5	Q: Performance Modelling of Distributed Systems	4	O: Theories and Techniques of Optimization	8	O: Theories and Techniques of Optimization	8	SP: Applied Software Project Management	7.5
	RE: Requirements Engineering	4	D: Software Architecture and quality	7.5	RE: Requirements Engineering	4	RM: Technical and Scientific Communication[30]	4	RM: Technical and Scientific Communication	4	D: Software Architecture and quality	7.5
	D: Middleware for Heterogeneous Distributed Information System	**8**	RE: Practical Requirements Engineering	7.5	D: Middleware for Heterogeneous Distributed Information System	**8**	RE: Requirements and Design of Software Systems	8	RE: Requirements and Design of Software Systems	8	RE: Practical Requirements Engineering	7.5
	Q: Quality Management software and System OR Q: Software Quality Assurance	4			Q: Quality Management software and System OR Q: Software Quality Assurance	4						
	D: Product Line Engineering	4			D: Product Line Engineering	4						
	TOTAL 1 SEMESTER	28		30	**TOTAL 1 SEMESTER**	28		28	**TOTAL 1 SEMESTER**	28		30
	VV: Formal Specification and Verification Techniques[31]	8	VV: Verification and Validation	7.5	VV: Formal Specification and Verification Techniques	8	VV: Software Reliability and Testing	8	VV: Software Reliability and Testing	8	VV: Verification and Validation	7.5
	SP: Software Project and Process Management	4	RM: Research Methodology	7.5	SP: Software Project and Process Management	4	RM: Empirical Software Measurement	8	RM: Empirical Software Measurement	8	RM: Research Methodology	7.5
	D: Software Architecture of Distributed Systems and D: Service Oriented Architecture[32]	**8**	D: Multiprocessor systems	7.5	D: Software Architecture of Distributed Systems and D: Service Oriented Architecture	**4**	NI: Computer Networks	8	NI: Computer Networks	8	D: Multiprocessor systems	7.5
	SP: Process Modelling	4	Q: Software Metrics	7.5	SP: Process Modelling	8	D: Open Source Software Engineering	8	D: Open Source Software Engineering [33]	8	Q: Software Metrics	7.5
	RM: Empirical Model Building and Methods and Software Engineering Seminar	8			RM: Empirical Model Building and Methods and Software Engineering Seminar	8						
	TOTAL 2 SEMESTER	32		30	**TOTAL 2 SEMESTER**	32		32	**TOTAL 2 SEMESTER**	32		30
2 YEAR FUB	RM: Research Methods	2	RM: Research Methods	2	SP: Advanced Topic in computing	7.5	SP: Advanced Topic in computing	7.5	VV: Safety & Reliability of Embedded systems	4	VV: Safety & Reliability of Embedded systems	4
	D: Infrastructure for Open Source Systems and services	8	D: Infrastructure for Open Source Systems and services	8	D: Product Line Architecture	7.5	D: Product Line Architecture	7.5	Q: Performance modelling of distributed systems OR D: Security of Distributed Systems	4	Q: Performance modelling of distributed systems OR D: Security of Distributed Systems	4
	P: Project	8	P:Project	8	P:Project	7.5	P:Project	7.5	P: Project	8	P: Project	8
	NI: Advanced Internet Technologies	8	NI: Advanced Internet Technologies	8	RE: Global Software Engineering	7.5	RE: Global Software Engineering	7.5	D: Product line Engineering	4	D: Product line Engineering	4
	RM: Technical and Scientific Communication	4	RM: Technical and Scientific Communication	4					D: Middleware for Heterogeneous and Distributed Information Systems	8	D: Middleware for Heterogeneous and Distributed Information Systems	8
									RM: Software Engineering Seminar	4	RM: Software Engineering Seminar	4
	TOTAL 3 SEMESTER	**30**	**TOTAL 3 SEMESTER**	**30**	**TOTAL 3 SEMESTER**	**30**	**TOTAL 3 SEMESTER**	**30**	**TOTAL 3 SEMESTER**	**32**	**TOTAL 3 SEMESTER**	**32**
	Master Thesis	30	Master Thesis	30	Master Thesis	30	Master Thesis	30	Master Thesis	30	Master Thesis	30
	TOTAL 4 SEMESTER	**30**	**TOTAL 4 SEMESTER**	**30**	**TOTAL 4 SEMESTER**	**30**	**TOTAL 4 SEMESTER**	**30**	**TOTAL 4 SEMESTER**	**30**	**TOTAL 4 SEMESTER**	**30**

[29] Depending on teaching offer and the student free choice this can be substituted with D: Security in distributed Systems OR D: Software Architecture for Distributed Systems

[30] Depending on teaching offer and the student free choice this can be substituted with D: Seminars in Human Machine Interactions

[31] on teaching offer and the student free choice this can be substituted with VV: Verification of Reacting Systems OR VV: Concurrency Theory OR RM: Regression and Time Series Analysis

[32] Depending on teaching offer and the student free choice this can be substituted with D: Automotive SE OR D: Service Oriented Architecture

[33] Depending on teaching offer and the student free choice with D: Open Tools and services OR SP: Software Quality Factory

Courses' Syllabus

FUB

Software Process Management

Teacher: prof. Alberto Sillitti

Content: To provide a comprehensive background for understanding and managing the development process in a software company including both SMEs and IT departments of large companies**.** The Plan-based approach and the Agile approach: XP, SCRUM. Personal Software Process (PSP). Team Software Process (TSP). Process Improvement

- Access the status of a software process
- Define and implement a measurement plan
- Define and implement a Process Improvement Plan

Discussion with managers of the software industry

Learning outcome: At the end of the course students will be able to:

- understand problems related to the software process
- adapt the standard process to the needs of a specific environment
- manage the execution and the optimization of a software team.

Theories and techniques of optimization

Teacher: prof. Davide La Torre

Content: This course provides optimization methods and techniques for students in computer science. A wide variety of real models (in economics, engineering, finance) will be introduced and students will continuously see how the mathematics they are learning can be used. Students are expected to read the assigned material and to attempt to solve the assigned exercises and problems for each class. Lectures are aimed at clarifying concepts presented in the text, answering students' questions, and taking up some of the assigned problems. A large amount of material is covered in this course. Each class builds on preceding ones and this makes it important to understand each concept as the course proceeds. Past experience indicates that the success rate is much higher amongst students who come prepared for each class. Therefore, it is vital for students to read the assigned material on a day to day basis, to do the proposed exercises and seek clarification when needed. Students are encouraged to solve more problems than those assigned for each class and to refer to other textbooks and relevant websites. Office hours will be useful for those students requiring extra assistance.

Learning outcome: Deep knowledge of optimization methods and techniques

Technical and Scientific Communication

Teacher: prof. Cinzia Colapinto

Content: For IT people knowledge transfer is crucial: as a result attending this specifically designed course will extend students' language competence and skills, with a focus on written & spoken production.
In particular, the course reduces anxiety toward public speaking.

- Technology/Knowledge transfer;
- Use of reading texts and videos from the world of computing;
- Gain skills for giving opinions & developing logical argument;
- Extension of soft skills – team work, problem solving, critical thinking, scientific writing;
- Techniques for the organization and delivery of results for oral, written and poster presentations;
- Vocabulary acquisition.

Learning outcome: This course will develop your use of several genres of scientific and technical writing, as well as teach you planning, drafting, and revision strategies that will improve the quality of your communication. The major aim of the lecture series is for students to be able to write and speak clearly and concisely for academic or professional purposes at an advanced level i.e. thesis, project proposals, reports, recommendations, journal articles, conference presentations.

Requirements and Design of Software Systems

Teacher: prof. Gabriella Dodero

Content: Students will acquire theoretical insights and practical experience from processes, tools and techniques that are used in requirements engineering in traditional, agile, open source, and also large-scale software development. Subsequently, they will understand how to design a software architecture that supports the collected requirements. In particular:•

- Introduction to Requirements Engineering
- Data and functional requirements
- Quality requirements
- Requirements Elicitation
- Checking and validation of requirements
- Architectural Structures and Views
- Architecture and Quality Attributes
- Architecture in the Life Cycle
- Evaluating Architectures through ATAM
- Case studies:
    - Service oriented architectures
    - Architectures for Product Lines
    - The architectural process in agile environment

Learning outcome: On completion of the course the student will be able to:

- understand the problems related to requirements engineering such as the ambiguity, quality or change of requirements understand security issues during requirement engineering
- use tools and techniques in order to elicit, specify and manage requirements according to the company-specific environment
- understand how to design software architectures based on the requirements coming from the stakeholders
- understand how to assess software architectures based on the trade-offs between architectural choices

Software Reliability and Testing
Teacher: prof. Sandro Morasca
Content: Students will acquire skills for monitoring, managing, and controlling software quality during the development and operational use. Specifically, the students will learn systematic, structured software testing and analysis techniques at unit, integration and system level including tool-supported test automation. In addition, the course enables students to plan and manage software testing and analysis in the context of modern software life cycles models, and provides insights in applying testing and analysis techniques and tools in project contexts. Students will learn methods to identify and analyze software defects, faults and failures with a rigorous and modern approach. They will be introduced to modeling software reliability with stochastic processes, in particular Markov Chains. They will discuss Software Reliability Growth models and measures of accuracy and prediction of the models according to the classification of Musa Okumoto. Students will be able to apply this theory during the laboratory activities.
Learning outcome: Be able to

- Planning and time management
- Critical thinking
- Information search
- Problem solving
- Analytical ability
- Team work

Empirical Software Measurement
Teacher: prof. Pekka Abrahamsson
Content: The objectives of the course are to give students a full understanding of measurement in software engineering and to propose appropriate empirical approaches to design, plan, execute, and analyze software measurement programs. The course focuses on:

- Introduction to Software Measurement;•
- Planning a Software Measurement Program;
- Software Measurement Standards (e.g. ISO/IEC 15939:2002);
- Collection and analysis of Software Measurement Data;
- Measuring external product attributes and resources;
- Software Engineering & Empirical Studies;
- Generalization and Theory building in Software Engineering;
- Empirical Strategies & Research Design;
- Case Studies Introduction, collection of evidence, analysis, and reporting formats;
- Surveys, Focus Groups, Ethnographies;

- Experiments: planning, operation, analysis, interpretation,
- presentation and proper package of knowledge;
- Replication of experiments & meta analysis;
- Advanced data analysis for experimentation: parametric and non-parametric techniques;
- Evaluation of Software Measurement programs with empirical studies.

Learning outcome: Students will learn how to create a proper software measurement program. They will be able to plan, design, and execute proper experiments to evaluate software and process attributes. They will understand proper techniques for data collection and analysis, discern good designs and criticize software measurement programs.

Computer Networks
Teacher: prof. Gabriella Dodero
Content: This course recalls the basic features of the supporting Internet technologies and protocols, and provides the student with practical experience on application layer protocols. In particular the course focuses on:

- Basic of computer networks
- The application layer
- The transport layer
- The network layer
- The link layer
- Wireless and mobile networks
- Networking and Multimedia
- Elements of network security

Learning outcome: Knowledge / Understanding:

- Understanding of network protocols and standards
- Understanding of issues in distributing an application across a network;

Skills:

- Network programming, use of sockets, distributed programming
- Inspecting and debugging a protocol over a network

Open Source Software Engineering
Teacher: Dr. Etiel Petrinja
Content: Develop a critical understanding and awareness of open source software motivations, practices, techniques, and solutions available for software development. In particular, the course focuses on

- Motivation for OSS engineering
- Understanding Open Source and Free Software Licensing
- OSS development practices
- Distributed development in OSS
- Communication in OSS
- OSS Development environment
- OSS projects governance
- OSS Business models
- Quality of OSS products
- Quality of the OSS development process
- Challenges of OSS

Learning outcome: After participation to this course the student will acquire a broad overview on key aspects of Open Source Software and Free Software Engineering practices.

Research Methods
Teacher: prof. Jurg Nievergelt
Content: Introduce students to general aspects of how to plan and execute research projects.
Students present a proposal for a possible MS thesis project. In particular, the course focuses on

- Styles of thinking:
- Different approaches to problem solving.
- Case studies of the development of scientific disciplines over long periods of time:
- Understanding the solar system
- The development of algorithmic

- The development and formalization of logic

Guest lectures and discussions on research projects conducted in the Faculty of Computer Science. Students present their proposal for a possible MS thesis project.
Learning outcome: Prepare students to start an MS research project.

Infrastructure for Open Source Systems and services
Teacher: Dr. Francesco Di Cerbo
Content: Service Oriented Computing, and Service Oriented Architecture have become embraced by organizations (big companies, but also SMEs) for their beneficial effect for the IT strategical management. Among other important advantages, SOC and SOA help in providing better alignment between business and IT for instance increasing agility for responding to demands in the marketplace. This makes SOC and SOA a crucial asset for developers and managers.
The objective of the course is to give students an understanding of SOC, and to enable them to create SOAs using a suite of open source products, that are market leaders.
Learning outcome: The students shall learn how to design effective orchestration and compositions of services using standard web technologies and tools.

Advanced Internet Technologies
Teacher: prof. Giancarlo Succi
Content: The objective of this course is to provide a comprehensive knowledge regarding Internet Technologies. This will include a significant study on design and development of web and mobile services. The course is not on a specific programming language; however, for simplicity most examples will be taken from Java or C#. The following major topics will be covered in the course:
- Design of web applications
- Tools and languages to develop web applications
- Design of web applications for mobile devices
- Tools and languages to develop mobile web applications

Learning outcome: Competence in analysing, designing, developing, and testing standard and mobile systems for the Internet.

BTH
Advanced Software Project Management
Teacher: Dr. Cigdem Gencel
Content: Most software is today developed in teams. It is therefore vital that software engineers have detailed knowledge and skills to manage and work effectively in project teams. The objective with this course is to provide the participants with a strong theoretical foundation in the field of general project management, behavioural sciences and organizational studies in connection to software project management (SPM) issues. The course is built around four blocks:
- Project management
- Organizational theory
- Leadership
- Behavioural sciences

Regarding Project Management (PM), the participants are introduced to the history of PM, concepts and applications, and PM in the context of software development, i.e. SPM. Concerning organizational theory, the participants are introduced to both macro theories (institutional, evolutionary, networks, etc.) and micro theories (cognitive, motivation, group, etc.) In the leadership block, topics such as leadership types, classifications and terminologies are introduced and discussed. With respect to behavioural sciences the role of the individual in a project context is examined and theories covering, in particular, decision science (psychology, management, etc.) are introduced and their role in project management is discussed.
Learning outcome: On completion of the course the student will be able to:
- Independently describe, and in a group, discuss the area of project management (history, standards, definitions etc.) and name a number of key issues.
- Independently describe, and in a group, discuss the area of organizational theory and how it applies to SPM.
- Independently describe and, in a group, discuss the area of behavioral sciences and their connection to SPM.

- Independently describe and, in a group, discuss the role of the leader in different types of organizations.
- Independently describe, compare and contrast different leadership types such as e.g. situational leadership.
- Independently develop an understanding of the key problems and benefits associated with managing people.
- In a group compare and contrast the different methods and techniques used to assure the quality of a software product in a SPM.

Applied Software Project Management
Teacher: Dr. Cigdem Gencel
Content: Most software is today developed in teams. It is therefore vital that software engineers have knowledge in this particular area. The objective with this course is to learn how to prepare (plan), execute and finalize a project. This includes, but is not limited to, steering and administrating a project from start to finish, applying methods and techniques for making sure the project ends in a successful manner and, additionally, understanding and learning how to interpret stakeholders' interest in a typical project. The course comprises the following:

- An Introduction to Project Management: Motivation, goals, state-of-the-art, future research directions, and related areas.
- Project Life Cycle and Organization: Planning, design, implementation in project management.
- Product Management Processes: Groups, interactions and mapping
- Management Knowledge Areas: Integration, scope, time, cost, quality, human resources, communication, risk and procurement planning.

Learning outcome: On completion of the course the student will be able to:

1. Independently describe the area of software project management (standards, key concepts, software quality definitions etc.) and name a number of key issues related to this area.
2. In a group, or independently, identify project management tools suggesting likely candidates for different project types.
3. Independently define risk management issues in software project management.
4. In a group, execute and finish a project with accompanying deliverables through involvement in a team project.
5. Independently define and categorize quality issues in software project management.
6. Independently explain the difference between traditional project management and contemporary software project management.
7. In a group, prepare a project plan for a software project that includes estimates of size and effort, a schedule, resource allocation, configuration control, change management, and project risk identification and management.
8. Independently be able to conduct post-mortem analysis with accompanying process improvement suggestions.

Software Architectures and Quality
Teacher: Asoc.Prof. Mikael Svahnberg
Content: Software Architecture (SA) is an essential technological concept in the modern software industry. It describes the elements and properties of software systems and the relationships between them and is a key regulator of system quality and thus of business success and growth. In this course the students are expected to attain detailed knowledge of software architectures and software quality and, in particular, how the latter is affected by the former. Furthermore, the students are expected to attain an understanding of how to design and evaluate software architectures based on modern methods and ideas such as design patterns, object-oriented frameworks and component-based software engineering. The course comprises the following elements:

- Quality aspects in software and software architecture.
- Architectural styles, languages and patterns.
- Methods for architectural design and evaluation.
- Component-based software engineering.

Learning outcome: On completion of the course the participant will:

- Be able to clearly express an in-depth insight in the area of software architecture (standards, key

concepts, software quality definitions etc.) and be able to name and describe a number of key issues related to this area.
- Be able to clearly express an in-depth insight of quality in software, and how this is realised in quantifiable goals.
- Be able to independently, both on a theoretical level and in practice, select between a number of architectural styles, languages and patterns depending on the requirements, and discriminate between them.
- With an attention to details be able to create and document a software architecture consisting of several views and taking several different concerns into consideration.

Practical Requirements Engineering
Teacher: Asoc.Prof. Mikael Svahnberg
Content: The first challenge in software development is making sure that you are developing the right system, i.e. Requirements Engineering. In this course the students are expected to gain an understanding of how to collect relevant requirements from relevant sources before software development commences. In this course we focus on the importance of, and how to ensure that the requirements are used and kept up-to-date throughout the entire development effort. The students will be taught how to find the right stakeholders of a system and to get information (requirements) from them. The students will be taught how to write and maintain a usable requirements specification. The students are offered an insight into how requirements are prioritized and packaged for different releases of software systems. This course covers the problems that arise when requirements engineering is conducted in a fast-paced, cost-sensitive industry reality. The following topics are included in the course:
- Processes for In-Project Requirements Engineering (Bespoke Requirements Engineering)
- Stakeholder Identification and Management
- Methods for Requirements Elicitation Writing Requirements and Requirements Specifications
- Quality Assurance of Requirements
- Connecting Requirements to subsequent development phases (e.g. design and testing phases)

Learning outcome: On completion of the course the student will:
- be able to skilfully elicit software requirements
- be able to clearly and unambiguously document software requirements according to industry standards
- thoroughly understand and be able to describe how to conduct bespoke requirements engineering in terms of common processes and techniques
- thoroughly understand and be able to describe the challenges involved in traditional requirements engineering
- be able to comprehensively assess current requirements engineering practices in a software project or a software development company
- be able to suggest relevant improvements on the requirements engineering processes in a convincing way

Verification and Validation
Teacher: Dr. Kai Petersen
Content: All software of any significant size needs to be verified and validated accordingly to assure a certain level of quality. The objective with this course is to provide the participants with an overview of verification and validation (V&V) of software systems. In addition several techniques (as used in industry or academia) are practiced and further examined. The course is built around three interdisciplinary thematic blocks:
- Software inspection (verification metrics, requirements, design and code inspection)
- Software testing (levels, methods and types of test, informal and formal validation, test planning, validation metrics)
- Software reliability engineering (reliability predictions, operational profiles, test efficiency, deployment, software reliability models, step-by-step process implementation)

All three blocks will cover the history (background), concepts (techniques) and applications, and applied usage of at least one technique from each theme respectively. In addition, a framework for V&V (providing a crude systematic review of V&V) will be given and, in the end, improved on by the participants.
Learning outcome: On completion of the course the student will be able to:

- Independently describe and, in a group discuss, techniques in the areas of software inspection, software testing and software reliability engineering.
- Individually name and explain a number of key research questions remaining to be answered in each of the three thematic blocks.
- In a group be able to apply at least one technique in each of the three thematic blocks.
- Individually apply software reliability engineering techniques.

Research Methodology
Teacher: Asoc.Prof. Mikael Svahnberg
Content: Empirical studies have become a key issue for evaluating and comparing competing software engineering and computer science methods, techniques and tools. Applying a rigorous scientific method ensures that the results from an evaluation or comparison are reliable. This is important for providing decision support when it comes to improving the way software is developed. Different types of empirical studies may be applied depending on the goal and the situation. In this course the student will be trained in research methodologies applied in software engineering and computer science. The course treats how to evaluate literature, how to formulate goals for your research project, how to formulate a research question and set up hypotheses to match your research question. The course presents qualitative as well as quantitative research methodologies, such as case studies, surveys, experiments with human subjects, mathematical modeling and simulations. The course also discusses research ethics, and how to package and present your research study in a research article.
Learning outcome: On completion of the course the student will:
- Be able to accurately find, read, and evaluate research articles .
- Be able to concisely formulate a research question .
- Be able to thoroughly design a qualitative research study .
- Be able to thoroughly design a quantitative experiment .
- Be able to write a research article that conforms to common academic practice and that thoroughly describes a research study.
- Be able to conduct a research project from start to finish.
- Thoroughly be able to describe the ethical implications involved in conducting a research project .
- Be able to rigorously reference other peoples work according to academic standards.

Multiprocessor Systems
Teacher: Prof. Håkan Grahn
Content: Since many years has parallelism been an approach to obtain high performance in computer systems. However, developing parallel programs is difficult and time consuming, and has therefore only been used in large-scale server systems. Today, that picture has changed with the introduction of multicore processors. Now is almost every computer equipped a small multiprocessor. In order to utilize the performance potential, the programs also need to be parallel. In other words, multiprocessor systems and parallel programming will be fundamental building blocks for contemporary as well as future computer systems. In order to be able to develop high-quality programs for future computers it is essential that the student develop a thorough understanding of different design principles for multiprocessor systems, as well as a thorough understanding of different methods and techniques for developing parallel computer programs. The course covers the following areas :
- introduction to multiprocessor systems and parallel programming
- design principles for multiprocessor and parallel computer systems
- design principles for parallel programs
- programming models for parallel programs
- practical training in development of parallel programs

Learning outcome: On completion of the course the student will:
- be able to generally describe the design and working conditions of different types of parallel computer systems
- independently be able to thoroughly describe different programming models for parallel computer systems
- independently be able to thoroughly describe the possibilities and problems that are present when developing parallel programs
- independently and practically be able to apply different techniques for developing parallel programs

Software Metrics
Teacher: Dr. Cigdem Gencel
Content: In any other engineering discipline, the term "best engineering practice" is repeatedly applied to all aspects of the development of a system. It might be a bridge, an automobile, or a large building. Best engineering practice embodies a long tradition of experimentation, analysis, and measurement. It is the application of scientific principles to the solution of a complex development project. At the core of the best engineering practice lies the measurement. For example, measurements in radar systems enable us to detect aircraft when direct vision is obscured. Medical system measurements enable doctors to diagnose specific illnesses. In software engineering, we need to understand and control the software projects. We need to know: What does each process cost?, How productive is the staff?, How can we improve?, etc. We need to measure to answer these questions. The course comprises the following elements:

- Fundamentals of measurement: The need for measurement, scope of software metrics; Measurement theory: scales, validation, and meaningfulness; Goal-Question-Metric paradigm; Measurement data collection and analysis; A classification of software metrics
- Software measures: Internal Product Attributes; External Product Attributes; Resource Measurement; ISO 9126 software product quality characteristics
- Software measurement process: Measurement process models; ISO/IEC 15939: Software Measurement Process; CMMI's measurement requirements

Learning outcome: On completion of the course the student will be able to

- present and discuss the fundamentals of software measurement
- professionally present, argue, discuss how software measures can be used during the software processes and be able to professionally apply them
- discuss the state-of-the-art in software measurement process models and CMMI's measurement requirements
- professionally present, argue, discuss the issues related to applying software measurement and how to set up measures and models

Advanced Topic in Computing
Teacher: Prof. Bo Helgeson
Content: The course covers an area in Computer Science or Software Engineering and is determined jointly by the student and the teacher or tutor. The course will be given in the form of lectures, seminars with the final seminar. In the seminars students write short papers, which are presented and discussed. Active feedback from teachers is the key element of the course.
Learning outcome: On completion of the course the student will be able to

- have acquired a deeper knowledge of a specific area within Computer Science or Software Engineering
- be familiar with the current state in a chosen area
- demonstrate an ability to articulate the grounds for a more detailed research work and to discuss and justify the choices made
- write a scientific paper

Product Line Architecture
Teacher: Dr. Ludwik Kuzniarz
Content: The structuring of software products into product-lines allows sharing of development effort within the product-family and as such counters the impact of ever growing system complexity. This makes it possible to sustain the rate of product innovation, while keeping guaranteed levels of overall system performance and quality. The fundamental concept of a software product line is a domain specific product architecture based upon a layered set of platforms. This is supported by a software engineering process focused on pervasive reuse. The issues in software product line are not just technical but include process, organizational and business aspects as well. The purpose of the course is to give in-depth insight in the area of software product lines. The course comprises the following elements:

- scoping and domain analysis,
- design of the software product-line architecture,
- development of software product-line components,
- instantiation of family members, i.e. products, and

- evolution of software product-line assets, i.e. the product-line architecture, the components and the products.

Learning outcome: On the completion of the course the students will:

- Have in-depth insight in the area of software product lines.
- Independently be able to conceptually describe and understand the fundamental concepts and ideas behind software product lines on a theorethical level.
- Independently be able to specify, analyze, design, evaluate, implement and use small size software product lines and effectively communicating the results in written and oral form.

Global Software Engineering

Teacher: Dr. Darja Smite

Content: With the extension of global markets software nowadays has been more often developed by global software teams. This transition has a significant impact on software engineering processes and practices. The objective of this course is therefore to provide knowledge, competence and practical experience regarding communication, cooperation and coordination of software engineering activities given the global setting. The student will be made aware of, and trained to work with diversity (e.g. cultural, social, and organizational); they will know how to communicate on a global network, in a global team, and interpret and sensitively exploit diversity in their professional (and personal) life. The course investigates and discusses GSE from the following 4 perspectives:

- Economic
- Technical
- Organizational
- Cultural

Economic perspective of GSE focuses on discussing the reasons of outsourcing decisions. Technical perspective touches the important topics of communication, cooperation and coordination with respect to distributed infrastructure management. Organizational perspective provides knowledge on how to distribute, maintain and manage software engineering processes throughout the project. Cultural perspective introduces a 'soft dimension' and its importance in the context of globally distributed software projects.

Learning outcome:

On completion of the course the student will:

- Independently cope with complexity of understanding the particularities of GSE and transcend global diversity and work in international teams;
- Independently describe and in a group discuss the application of general software engineering principles with respect to GSE setting;
- In a group demonstrate critical understanding of the tools, methodologies and techniques for communication, cooperation and coordination in global projects and be able to apply them to a range of development scenarios.

UKL

Safety and Reliability of Embedded Systems
Teacher: Prof. Peter Liggesmeyer
Content: Safety and reliability are particularly important quality criteria for software applications in the technical sector. In many domains - e.g. rail-mounted vehicles, avionics, automotive engineering, medical technology - a software failure can endanger human lives. Hence, for example, safety has to be proved before the initial start-up of such systems. These proofs must be complete in general or have to prove at least that a tolerable residual risk is not exceeded. The lecture is divided into a basic part and a practical part. In the basic part current techniques for the safety and reliability analysis are presented (Symbolic Model Checking and stochastic reliability analysis). In the practical part representatives of industrial companies, which develop security-critical software-intensive systems, report on the situation in practical use.
Learning outcome: Handling of special formal and stochastic techniques for the safety and reliability analysis of software and systems, knowledge for the application of relevant methods for analysis in practical use.

Empirical Model Building (Formation and Methods)
Teacher: Prof. Dieter Rombach
Content: Definition of the basic terms of empirical / experimental software engineering; the procedure of experimentation; formulation of hypotheses; design of empirical studies; acquisition, validation and analysis of data; interpretation of data; empirical modelling; examples of concrete empiric studies
Learning outcome: This lecture provides the basic knowledge that a successful researcher or quality manager needs today. On the basis of practical examples, it shows: how to plan and realise empiric methods in software engineering, which empirical methods are used for testing research hypotheses, which empirical methods are used for descriptive modelling of software products and processes, which empirical methods are used for improving either software product models or software process models.

Security in Distributed Systems
Teacher: Prof. Jens Schmitt
Content:
History of secure communications
Symmetric cryptography: DES, 3DES, AES
Asymmetric cryptography: RSA, Diffie-Hellman, El Gamal
Cryptographic protocols: Needham-Schroeder, Kerberos, X.509
Security protocols in the link layer: PPP, EAP, PPTP, L2TP
Security protocols in the network layer: IPSec
Security protocols in the transport layer: SSL/TLS, SSH
Security in mobile systems
Security in WLAN
Security in wireless sensor networks
Learning outcome:
Students have knowledge of theoretical as well as practical aspects of security in distributed systems and skills to apply cryptographic fundamentals in fixed as well as in wireless and mobile systems.

Performance Modelling of Distributed Systems
Teacher: Prof. J. Schmitt
Content:
•Different techniques for performance evaluation and modelling: measurement , simulation, and analysis
•Statistical planning and evaluation of experiments
•Simulation techniques as for example random number generation
•Analytical methods: queuing networks, network calculus
•Application examples / case studies
Learning outcome: Learning the art of performance-related modelling of complex distributed systems
•Performance evaluation of existing systems
•Performance dimensioning of planned systems

•Performance control of running systems
•Quality of service guarantees in distributed systems
•Resource management in distributed systems

Verification of Reactive Systems
Teacher: Prof. Klaus Schneider
Content:

- syntax and semantics of logics for the specification of temporal properties: μ-calculus, ω-automata, temporal logics, monadic predicate logics
- translation to automata and fixpoint representations for model checking
- symbolic representation of transition systems
- symbolic model checking
- abstraction and other state reduction techniques
- model checking of infinite state spaces
- hybrid systems (discrete/analog systems)

Learning outcome:

- specification of temporal properties
- application of abstraction/reduction techniques
- practical use of model checking tools

Concurrency Theory
Teacher: Junior Prof. Roland Meyer
Content:

- Multi-threaded programs and Petri nets
- Petri net specific analyses
    - Karp and Miller coverability graphs
    - Invariants
    - Unfoldings + SAT
- Static networks and lossy channel systems (lcs)
- Analysis of well-structured transition systems (wsts), with lcs as an example
    - Wsts and Finkel's finite reachability tree
    - Abdulla's backwards algorithm
    - Geeraert's EEC algorithm
- Reconfigurable networks and process algebras
- Analysability
    - Structural stationarity, depth, and breadth
    - Well-structuredness in bounded depth
    - Turing completeness in bounded breadth
- Bisimilarity, an alternative correctness notion
- Analysis and its limits
    - Fixed points in the finite
    - Communication freedom and prime elements in the infinite
    - Undecidability following Jančar

Learning outcome: Goal is to develop operational models for systems of interacting components. You will get to know different correctness notions and understand the corresponding verification algorithms. At the end of the course, you will be able to design similar analysis algorithms for related system models.

Regression and Time Series Analysis Teacher: Dr. Claudia Redenbach
Content: Linear regression models, parametric curve fitting, likelihood ratio tests, data adaptive model selection, analysis of variance (ANOVA), experimental design. Stationary stochastic processes, autoregressions and ARMA-processes, parameter estimation and model selection for time series, trend and seasonality, forecasting by exponential smoothing and the Box-Jenkins method, linear filters.
Learning outcome: Students know and understand models as well as estimation, testing and forecasting of regressions, variance, and time series analysis. Student will know about examples of mathematical methods

for data-driven selection and validation of models in complex application scenarios.

Quality Management Software and System
Teacher: Prof. Peter Liggesmeyer
Content: •Organization forms of the quality management for software and systems
•Techniques for the quality management, e.g. TQM, QFD, standards in quality management, procedures based on stage of maturation in the quality management.
•Construction of a quality management system
Learning outcome: The students know the technical contents and the possible organization forms of quality management. Beyond that they are able to evaluate different approaches and to apply them in the sense of a quality management system

Formal Specification and Verification Techniques
Teacher: Prof. Klaus Madlener
Content: •The role of formal specifications in system development
•ASM Method (computation model, states and refinement notions)
•Algebraic specification methods (semantics, algebras, models: initial and final models, loose semantics, term generated, specification morphisms)
•Equational calculus and programming
•Reduction systems and term rewriting systems (termination and confluence properties)
•Verification of algebraic specifications (Completion methods and inductive theorem proving)
•Related calculii: lambda-calculus, calculus of combinators (functional programming)
•Implementation techniques of for algebraic specifications, rapid prototyping , reduction strategies, graph rewriting
•Tools for term rewriting
Learning outcome: Acquisition of knowledge of methods in formal specification, verification and implementation using concrete specification and verification techniques. In particular knowledge on
•Specification styles (axiomatic descriptive, model based operational)
•Mechanisms for structuring specifications (the principles of decomposition and modularization, parameterising, refinement)
•Correctness (verification and validation)
•Operationalisation of specifications: rewriting methods
•Tools for the construction and analysis of specifications

Software Project and Process Management
Teacher: Prof. Dieter Rombach
Content: Based on the contents of the lecture "Foundations of Software Engineering", which explains the technical aspects, this lecture explains how to establish and realise a successful project management and quality assurance (QA). On the basis of practical examples, it shows:
•how large projects are planned,
•which elements a project plan contains,
•which methods exist for project management and QA,
•how project management and QA are different to each other and how they complement one another
Topics of the lecture:
•Definition of project management and QA
•Basic terms of software project execution
•Process and quality models as a basis for systematic project management and QA
•Principles of project and process management
•Principles of measurement and evaluation
•Techniques / methods / tools for supporting project management and QA
•Examples from practical projects
Learning outcome: This lecture provides knowledge about principles, methods and tools of software engineering

Middleware for Heterogeneous Distributed Information Systems
Teacher: Prof. Stefan Deßloch
Content:

IS Architecture Concepts
Distributed Transaction Processing
Data Integration
DB-Gateways (ODBC, JDBC, SQLJ)
Web-based DB Access
Distributed Objects and Components (CORBA, EJB, J2EE - transactions, persistence, security, integration of legacy systems)
Message-Oriented Middleware (message queuing, message brokering)
XML (fundamentals, XML as an integration technology)
Web Services (fundamentals, extensions for robust web services)
Business Processes (modeling, workflow management, transactional workflows, orchestration of web services)
E-Business Integration
Learning outcome: The course provides a solid foundation in concepts, methods, and technologies for developing modern information systems using middleware.

Process Modeling
Teacher: Jens Heidrich
Content:
Introduction and classification (objectives, research and application areas)
Terminology (process model, role, 4-domain-principle)
Prescriptive process modeling (life cycle models, standards, examples, assessment criteria, process gates)
Descriptive process modeling (possible usages, procedure, process elicitation)
Process modeling notations (Appl/A, Funsoft Nets, Marvel, Statemate, MVP-L, IDEF0, ETVX)
Process modeling tools (ECMA/NIST reference model, modeling tools, PSSEs, examples)
Software project planning (effort estimation, schedule planning, personnel planning, sequence planning)
Project monitoring and management (data collection, visualization of metrics)
Other usages (SPI, QIP, ISO 15504, ISO 9000, CMMI, process simulation)
Future developments (agile process documentation, process machines, process patterns)
Learning outcome:
Gaining knowledge and capabilities for designing, creating, analyzing, and applying software development processes
Becoming acquainted with industrial software development processes
Independent modeling of software development processes
Advantages and disadvantages of process modeling techniques
Applying process models effectively for different purposes

Safety & Reliability of Embedded systems
Teacher: Prof. Peter Liggesmeyer
Content: Safety and reliability are particularly important quality criteria for software applications in the technical sector. In many domains - e.g. rail-mounted vehicles, avionics, automotive engineering, medical technology - a software failure can endanger human lives. Hence, for example, safety has to be proved before the initial start-up of such systems. These proofs must be complete in general or have to prove at least that a tolerable residual risk is not exceeded. The lecture is divided into a basic part and a practical part. In the basic part current techniques for the safety and reliability analysis are presented (Symbolic Model Checking and stochastic reliability analysis). In the practical part representatives of industrial companies, which develop security-critical software-intensive systems, report on the situation in practical use.
Learning outcome: Handling of special formal and stochastic techniques for the safety and reliability analysis of software and systems, knowledge for the application of relevant methods for analysis in practical use.

Empirical Model Formation and Methods
Teacher: Prof. Dieter Rombach

Content: •Definition of the basic terms of empirical / experimental software engineering
•The procedure of experimentation
•Formulation of hypotheses
•Design of empirical studies
•Acquisition, validation and analysis of data
•Interpretation of data
•Empirical modelling
•Examples of concrete empiric studies
Learning outcome: This lecture provides the basic knowledge that a successful researcher or quality manager needs today. On the basis of practical examples, it shows:
•how to plan and realise empiric methods in software engineering,
•which empirical methods are used for testing research hypotheses,
•which empirical methods are used for descriptive modelling of software products and processes,
•which empirical methods are used for improving either software product models or software process models.

Product Line Engineering
Teacher: Prof. Dieter Rombach
Content: •Basic concepts of product lines (commonality, variability, decisions)
•Role and concepts of architectures (styles, patterns, and scenarios)
•Implementation technologies (MDA, Preprocessors, aspect-orientend development)
•Technology transfer (Adaptation and adoption of technologies, migration strategies)
•Reverse-Engineering (basic and detailed analyses, reconstruction of architectural views and structures)
•Domain analysis (product map, management of varying requirements and system characteristics)
Learning outcome: Transfer of knowledge and education in activities required for a systematic planning and realization of product lines (PL), or respectively software reuse in general.
•Organizational issues (reuse life cycle, migration)
•Definition, development and assessment of product line architectures
•Modelling and implementation of generic components
•Analysis of product variants
•Support of software development by reverse engineering

Security of Distributed Systems
Teacher: Prof. Jens Schmitt
Content: •History of secure communications
•Symmetric cryptography: DES, 3DES, AES
•Asymmetric cryptography: RSA, Diffie-Hellman, El Gamal
•Cryptographic protocols: Needham-Schroeder, Kerberos, X.509
•Security protocols in the link layer: PPP, EAP, PPTP, L2TP
•Security protocols in the network layer: IPSec
•Security protocols in the transport layer: SSL/TLS, SSH
•Security in mobile systems
•Security in WLAN
•Security in wireless sensor networks
Learning outcome: Students have
•knowledge of theoretical as well as practical aspects of security in distributed systems
•skills to apply cryptographic fundamentals in fixed as well as in wireless and mobile systems.

Grading Scheme

ECTS-grade	FX/F	E	D	C	B	A
BTH	Fail	Pass (weak)	Pass	Pass (strong)	Pass with Distinction	Pass with Distinction (strong)
UKL	5,0		4,0; 3,7	3,3; 3,0; 2,7	2,3; 2,0; 1,7	1,3; 1,0
FUB	Fail	18-20	21-25	26-28	29-30	30/30L

Staff and Student Population involved in EMCSEAppendix

We expect to have an intake of about 45 students per year, who will be taught by over 25 lecturers and their assistants. The staff members and students involved are distributed as follows:

- **FUB**: 3 full professors (Giancarlo Succi, Pekka Abrahamsson, Gabriella Dodero), 2 associate professors (Barbara Russo, Alberto Sillitti), 1 assistant professor (Etiel Petrinja), 5 non-tenured assistant professors (Andrea Janes, Francesco di Cerbo, Bruno Rossi, Ilenia Fronza, Tadas Remencius) and 10 PhD students as teaching assistants. A maximum of 15 students per academic year.
- **BTH**: 3 full professors (Claes Wohlin, Håkan Grahn and Lars Lundberg), 2associate professors (Mikael Svahnberg, Tony Gorschek), 8 assistant professors (Jürgen Börstler, Robert Feldt, Samuel Fricker, Cigdem Gencel, Ludwik Kuzniarz, Kai Petersen, Darja Smite, Richard Torkar), 1 lecturer (Nina Fogelstöm-Dzamashvili), and 8-10 PhD students as teaching assistants. A maximum of 15 students per academic year.
- **UKL**: 4 full professors (H. Dieter Rombach, Reinhard Gotzhein, Peter Liggesmeyer, Arnd Poetzsch-Heffter) 2 associate professors (Andreas Rauzh and Klaus Schmid) and 4 assistants. A maximum of 15 students per academic year.

Associated partners

- **Engineering**. The reference person is Stefano de Panfilis. Engineering is the leading company of the Engineering Group. This is the Italy's largest systems integration group and a leader in the provision of complete IT services and consultancy. Engineering Group has about 6500 employees and 35 branch offices, throughout Italy, Ireland and Belgium, and (outside the EU) in Brazil and Latin America. It has a global production capacity in 30 different countries, mainly related to projects in the industrial and telecommunications sectors. In 2009 Engineering consolidated revenues are 724,00 M€ with an EBITDA of 93,6 M€. The company operates through 7 business units: Finance, Central Government, Local Government and Healthcare, Oil Transportation and Services, Utility, Industry and Telecom, supported by an SAP transverse skills centre and by its Central Division for Research & Innovation. Engineering has been listed on the Milan Stock Exchange since December 2000 and ordinary shares are traded in the AllStars segment (33% are traded on the market and 67% are held by company founders). In the early 1990s, Engineering was one of the first Italian companies to adopt the Quality standard ISO 9001. Since 1996 the company has adopted NATO standard AQAP 2110/160 certification. And recently the production units have been certified CMMI® level 3. Engineering enhances its competences and services in the area of managed operations at the Pont Saint Martin Service Centre (PSM), which is a concrete example of European-level technological excellence for outsourcing services, provided to more than 100 Italian and international customers. The PSM center manages 40.000 workplaces, 1.000 remote connections, 10.000 electronic mail boxes and about 7.000 SAP users with 100 outsourcing contracts for over 3.500 servers. In 1987 the company started the R&D Lab Department and since then, through the lab, the company participated in more 130 research projects about 100 of which EU funded.
  http://www.eng.it/web/eng/home
  Stefano De Panfilis is the Director of the Research & Development Laboratories Department of Gruppo Engineering, leading a team of about 100 researchers in Italy and Europe.
  His interests are in software engineering, with a particular care of process customisation, support, monitor, and control. He graduated cum laude in Mathematics from the University of Rome "La Sapienza". In April 1994, he started his involvement in European funded R&D projects, where he coordinated NEXOF-RA (IST 7FP), an Integrated Project aiming at implement a Reference Architecture for NESSI, SeCSE (IST 6FP), an Integrated Project aiming at implement a platform to support Service-Centric Systems Engineering, he managed the CBSEnet (IST-5FP Network of Excellence), CLARiFi (IST 5FP), DOOR (ESPRIT 4FP), SQUAD (INCO-DC), and VALSE (TV&TT Initiative) projects, as well as the ENG-SODEPRO, DECO' and DOMINARE ESSI PIEs (Process Improvement Experiments). Currently, he has also coordinated QualiPSo (IST 6FP), an Integrated Project aiming at leverage Open Source Software development to widely recognised industrial standards, leading a consortium of more than 20 partners spanning Europe, Brazil and China. He actively participated since its beginning to the creation and setting up of the NESSI ETP of which he is the Coordinator of the Strategic Research Agenda Committee. Author of several scientific papers appeared on international journals and in conferences proceedings. He is member of the Board of Director of OW2 and member of several international conferences Program Committees.
- **Fraunhofer Institute of Software Engineering, Kaiserslauter, Germany.** The Fraunhofer-Institute for Experimental Software Engineering IESE is one of the worldwide leading research institutes in the area of software and systems development. A major portion of the products offered by our collaboration partners is defined by software. These products range from automotive and transportation systems via automation and plant engineering, information systems, health care and medical systems to software systems for the public sector. Our solutions allow flexible scaling. This makes us a competent technology partner for organizations of any size – from small companies to major corporations. Under the leadership of Prof. Dieter Rombach and Prof. Peter Liggesmeyer, the past decade has seen us making major contributions to strengthening the emerging IT location Kaiserslautern. In the Fraunhofer Information and Communication Technology Group, we are cooperating with other Fraunhofer institutes on developing trend-setting key technologies for the future. Fraunhofer IESE is one of 60 institutes of the Fraunhofer-Gesellschaft. Together we have a major impact on shaping applied research in Europe and contribute to Germany's competitiveness in international markets.
- **Fondazione Libera Università di Bolzano.**

- The Foundation is a non-profit entity that pursues the goal of the acquisition of goods and services for the Free University of Bolzano-Bozen and the running of activities supporting teaching and research in science and technology with specific reference to the following activities: to support initiatives and activities of students and teachers of the LUB; to promote, support and manage scientific, cultural and educational activities of the LUB, including the funding of teaching convention or the allocation of contributions and grants to researchers studying or graduates for research and training; to support or implement projects consistent with the purposes of the Foundation; to support the collection of private and public funds and demands for public and private contributions to be allocated for the purposes of the Foundation; to participate in or encourage the establishment of consortia and associations that share the same objectives.
- Avv. Johannes Egger, was born in Bolzano. He graduate in law at the Universities of Innsbruck, Padova and Paris. He is lawyer since 1966 and member of various cultural associations.

Short bio of the local coordinators (Joint Board representatives):

Barbara Russo (FUB) Barbara Russo is Associate Professor at the Faculty of Computer Science of The Free University of Bozen-Bolzano and is director of the Master Course in Computer Science. She got her PhD in Mathematics from the University of Trento, Italy. She was visiting researcher at the Max-Planck Institut für Mathematik in Bonn and the University of Liverpool as Marie Curie Fellow. She has been awarded by the Italian Association for Enterprises for the project "Bachelor studies for working students." She has several publications in international journals including Journal of Empirical Software Engineering and Journal of Software Systems. Her books include Adopting Open Source Software, A Practical Guide (with Fitzgerald, Kesan, Shaikh and Succi), MIT Press, forthcoming, 2011. She has participated to several national and international projects (as NAME EU FP5, COSPA EU FP6, CALIBRE EU FP6, MAPS Italian Ministry, TEKNE Italian Ministry). Her research interests are in the field of empirical software engineering and software measurement. Her competences are in statistical models of software data for software reliability and process improvement with focus on open source software and agile methods.

Darja Šmite (BTH) is an assistant professor of software engineering at Blekinge Institute of Technology and an associate professor at University of Latvia. Her research interests include global software engineering with the emphasis on improving distributed team efficiency, requirements engineering, team coordination and software process improvement. Smite received a PhD in computer science from University of Latvia. Prior to academic career she was working in several software houses in Latvia. Contact her at Blekinge Inst. of Technology, PO Box 520, SE-372 25 Ronneby, Sweden; Darja.Smite@bth.se

Dieter Romabch (UKL) Prof. Dr. H. Dieter Rombach studied mathematics and computer science at the University of Karlsruhe and obtained his Ph.D. in computer science from the University of Kaiserslautern (1984). Since 1992 he has held the Software Engineering Chair in the Department of Computer Science at the University of Kaiserslautern. In addition, he is the founding and executive director of the Fraunhofer Institute for Experimental Software Engineering (Fraunhofer IESE) in Kaiserslautern. From 2006 until 2009, he was also a member of the Management Board of the Fraunhofer-Gesellschaft e.V. and chairman of the Information and Communications Group, which consists of 17 institutes. Prior to being appointed director of Fraunhofer IESE, Prof. Rombach founded the Software Technologie Transfer Initiative (STTI) Kaiserslautern and was its director for four years. This initiative led to the foundation of Fraunhofer IESE. From 2001 to 2006, Prof. Rombach was also a Visiting Professor at the Computer Science Department of the University of New South Wales, Sydney, Australia. During the course of the years, Prof. Rombach has declined several university offers (including one to TU Vienna).

Previous career steps included the Institute for Data Processing in Technology at the Karlsruhe Nuclear Research Center (scientist; 1978-79) and the Department of Computer Science at the University of Kaiserslautern (scientist; 1979-1984). This was followed by positions as a guest professor at the University of Maryland and at NASA (1984-1986), as a professor for computer science at the University of Maryland (1986-1991), and as a professor at the Institute for Advanced Computer Studies at the University of Maryland and project manager at the Software Engineering Labor (SEL) at NASA's Goddard Space Flight Center (1986-1991). Prof. Rombach spent the summer semesters of 1988 and 1989 as a visiting professor at the Software Engineering Institute of Carnegie Mellon University in Pittsburgh, USA.

Prof. Rombach's research interests are in the area of "software engineering", particularly in engineering-style methods for the development of software with predictable quality; quantitative methods for the measurement of software products and processes for the purpose of project management and quality assurance; languages, methods, and tools for the creation and management of development processes on the basis of explicit software process models; as well as empirical methods and their application for determining the effects of software development methods.

In the context of his activities as director of Fraunhofer IESE, Prof. Rombach regularly serves as an expert, auditor, reviewer, and consultant for industry. He provides advisory services to a number of government bodies on the state and federal level as well as on the international level on issues concerning research as well as education and training in the area of computer science and on strategic decisions related to software. Prof. Rombach serves as a scientific adviser to various companies and research institutions.

Prof. Rombach is the author of more than 200 scientific publications. In 1990, he received the "Presidential Young Investigator Award" (endowed with US $ 500.000) of the National Science Foundation (NSF) in the U.S. for his excellent work in the area of software engineering. In 2000, his contributions to the scientific and economic development of the state were recognized when he was awarded the Service Medal of the State of Rhineland-Palatinate. In 2003, he received the Distinguished Postdoctoral Award of the College for Computer, Mathematical and Physical Sciences of the University of Maryland. In the year 2009, he was awarded the Federal Cross of Merit on Ribbon of the Federal Republic of Germany and the University of Oulu, Finland bestowed upon him an honorary doctorate degree in recognition of his lifetime achievements as a software engineer. Since 2009 he has been the chairman of the IEEE Awards Committees for the Software Process Achievement Award (SPA, awarded jointly with the Software Engineering Institute (SEI) of Carnegie Mellon University) and for the Harlan Mills Award.

Furthermore, Prof. Rombach is co-editor of several international journals (e.g., McCluwer Journal for Empirical Software Engineering) and is regularly called upon to act as a program committee member of important software engineering conferences. His appointments include having been the General Chair of the International ACM/IEEE Conference on Software Engineering held in Berlin in 1996 and the Program Co-Chair of the International ACM/IEEE Conference on Software Engineering held in Shanghai in 2006. He is a member of the Gesellschaft für Informatik (GI) and is a Fellow of the ACM (since 2010) and a Fellow of the IEEE Computer Society (since 2003).