Towards a European Master Programme
on Global Software Engineering

Patricia Lago[1], Henry Muccini[2], Ljerka Beus-Dukic[3], Ivica Crnkovic[4],
Sasikumar Punnekkat[4], Hans Van Vliet[1]
[1]*Vrije Universiteit , The Netherlands,* [2]*University of L'Aquila, Italy,* [3]
University of Westminster,UK ,[4]*Mälardalen University, Sweden*
*patricia@cs.vu.nl, muccini@di.univaq.it, beusdul@wmin.ac.uk,
ivica.crnkovic@mdh.se, sasikumar.punnekat@mdh.se, hans@cs.vu.nl.*

Abstract

This paper presents a European Master programme on Global Software Engineering (SE), being put forward by four leading institutions from Sweden, UK, Netherlands and Italy. The Global SE European Master (GSEEM) programme aims to provide students with an excellence in SE based on sound theoretical foundations and practical experience, as well as prepare them to participate in global development of complex and large software systems. GSEEM has been designed with three noteworthy aspects: 1) Three specialization profiles in which the consortium excels: Software Architecting, Real-time Embedded Systems Engineering, and Web Systems and Services Engineering. 2) Two market-driven routes: "professional" to work as professionals, and "scientific" to continue the education towards research degrees. 3) An innovative concept of "shared modules", delivered together by multiple institutions. Four types of shared modules are foreseen: "parallel" twin modules which run remotely between universities, "shifted" modules which teach SE concepts incrementally with shifts in study locations and timeline ,"complementary" modules in which complementary SE concepts are taught in parallel through shared projects, and "common" modules which share the presentations and the project. The profiles realize "integrated knowledge" by complementing partial knowledge available at partner institutions. The paper explains how GSEEM achieves the objectives of educating global software engineers.

1. Introduction

The development of large software systems and software-intensive systems is essential in many different business domains, including telecommunications, industrial automation, avionics and automotive industries, consumer electronics, in-house intelligent devices, different types of information systems, and web-based and service-oriented systems, such as e-government, e-health, traffic control, etc. Software Engineering is also essential for organising sophisticated and efficient production, monitoring the environment, decreasing pollution, and helping in improving services.

The objectives of the Global Software Engineering European Master (GSEEM) programme [5] are to provide students with an excellence in Software Engineering based on both sound theoretical foundations and practical experience, and prepare them to participate in global development of complex and large software systems, or systems that include software. The focus of the programme is on Software Architecting, Software Engineering of Real-time Embedded Systems, and Web Systems and Services Engineering.

The programme is offered by a consortium comprising Mälardalen University (MDH), Sweden, University of L'Aquila (UDA), Italy, University of Westminster (WU), UK, and Vrije Universiteit (VU), The Netherlands. By building the consortium,

the universities significantly added value to their existing individual curricula in a common curriculum, which covers different but highly related areas important for an integrated approach to global software development.

Software Engineering graduates are welcomed as development engineers, service engineers, system architects, project managers and team leaders, information analysts and policy advisors equally in Europe, the USA and in other countries. While such positions are very attractive, they are also challenging. They require not only excellence in technical expertise but also social competence [11]. Typically, software engineers are leading or are involved in team projects, which are often distributed, mobile, and in which members have diverse skills, different backgrounds, and may speak different languages. Software Engineering must address these challenges not only through the development of new techniques, practices and tools but also by educating a new type of a global software engineer with a wider critical view on learning, knowledge and experience.

The outline of the paper is as follows: in section 2 we briefly present related work. Section 3 presents GSEEM and its novelties. Section 4 presents the global approach, followed by the experience of the partner institutions in conducting shared modules in section 5. Section 6 presents some practical implications of GSEEM and the conclusions are given in Section 7.

2. Related Work

Both industry and academia have recognized the need for adjustments in software engineering education, to effectively train the future generations of global software engineers. We are not aware of any comprehensive approach in this direction except for a few isolated efforts and experiences worldwide. The ACM job migration task force stresses in [13] the difficulties in translating an educational response to offshoring into practical curricula, and points out that the US educational system is still in the process of identifying the curriculum changes essential to address application domain knowledge and global workplace in order to keep its innovative edge.

Berglund [1] postulates fundamental questions on the what, why, how and where of the learning experience of the computer science students. This case study was based on student interviews carried out in a distributed project course called 'the Runestone initiative' given jointly by Uppsala University, Sweden, and GVSU, USA. Here, the presentation focuses on the way the students perceive their learning environment, to unearth the complex relationships between material taught in the course, and the many ways of experiencing an advanced learning environment like the one including global issues. Navarro and van der Hoek [10] experimented with a project involving students coming from different specializations, hence adding interdisciplinary backgrounds. The work of Verkamo et al. [12] focused on distributed cross-cultural SE. They carried out a case study in two universities in Finland and Russia where they observe that initial cultural differences are solved by experience during the project.

Although global software engineering has received recognition as a discipline both in practice and academia (e.g. [6][7]), education of global software engineers has not reached the SE curricula yet. The valuable initiatives to update SE curricula on a global level (SWEBOK [14], ACM/IEEE [2]) do not yet address the emerging aspects of global SE. Computing Curricula 2005 [2] mentions outsourcing, offshoring, and job migration merely in the context of the pace of change in the workplace and fails to acknowledge the fact that students need to be prepared for global software

development. Similarly, the Software Engineering Body of Knowledge (SWEBOK [14]) conspicuously does not include any reference to global software development.

The GSEEM programme originated from teacher-student mobility agreements between the partner universities under the Erasmus Socrates programme. Additional impetus came from the Erasmus Mundus programme [8] which supports selected European master courses and aims to prepare European citizens for life in a global, intercultural and knowledge-based society.

3. GSEEM

The IT technology of today shows not only a domination of web-based technologies on the one side and embedded computer systems on the other, but also their integration via architectural infrastructures. For example, the integration of web-based applications with different monitoring systems, mobile telephones, sensors and similar in a distributed and mobile architectural framework. Successful development of such integrated systems requires deeper understanding of these three traditionally different domains and hence the GSEEM programme has been built upon the notion of 'profile orientation'. These profiles were also chosen as a natural consequence of the existing research and academic excellence at the partner institutions. Looking at the industrial requirements as well as the preferences expressed by students of the ongoing programmes, we have identified two different 'routes' which can be taken by a student. The profiles and routes of GSEEM are described in the following subsections.

3.1. Profiles

The GSEEM programme provides Master education in Software Engineering with a focus on the domains that are strategically important for IT and IT-intensive products and services. GSEEM provides insights into these domains both separately and in an integrated way by offering modules that crosscut these areas. GSEEM has three separate Profiles:

The **Software Architecting (SA) Profile** comprises general principles for the analysis, design and management of large and complex software systems. These principles can be applied to various types of software systems and include different development approaches and techniques. GSEEM focuses on modern trends that will dominate the future of software development including dynamic architectures, component-based software engineering, service-oriented architectures, interoperability and global system development.

The **Real-time Embedded Systems Engineering (ES) Profile** is related to the domains of embedded and real-time systems. Such systems are increasingly present in all aspects of human life, as practically all products of today include embedded software (from mobile phones, sensors and communication devices, to large products such as vehicles or airplanes). Such systems require knowledge of Software Engineering to provide solutions related to reliability, safety, resource usage, timeliness, predictability and similar properties. The Profile is focused on modelling and analysis of these properties and real-time embedded systems life-cycle.

The **Web Systems and Services Engineering (WS) Profile** focuses on Web and Internet distributed applications and services. Like embedded systems, this class of applications is rapidly growing and playing an increasingly important role in industry and in everyday life. Web applications require specific solutions related to concerns such as security, integrity, timeliness, performance and usability. The focus of this

Profile is on architectural engineering, interoperability, implementation, testing and deployment using component-based and model-based technologies.

3.2. Routes

The GSEEM programme envisages two Routes:
1. The **Professional Route** (60 ECTS, duration 1 year) enables students to work as professionals in the development of large software or software-intensive systems.
2. The **Scientific Route** (120 ECTS, duration 2 years) provides students with an additional qualification and the means to continue towards a PhD degree.

Table 1: Example of selected modules for a specific Profile and Route

Software Architecting Profile – Scientific Route		
Semester 1 : Vrije Universiteit		
Module	**ECTS**	**TYPE**
Advanced topics in software design	6	Core
Software architecture	6	Core
Distributed systems	6	Core
Software asset management	6	Core
Intelligent interactive distributed sys.	6	Option
Semester 1, total:	**30**	
Semester 2: Vrije Universiteit		
Project ISO	9	Core
Software configuration management	6	Core
Network programming	9	Option
Visual design	6	Option
Semester 2, total:	**30**	
Semester 3 : Mälardalen University		
Analysis of Algorithms	7.5	Core
Advanced Component-based SE	7.5	Core
Distributed Software Development	7.5	Option
Research Methodology for CS and SE	7.5	Core
Semester 3, total:	**30**	
Semester 4 : Mälardalen University		
Masters thesis – Scientific Route	30	Core
Semester 4, total:	**30**	

In the Professional Route, a student will spend his/her first semester at one chosen university, and the second at their second chosen university. In each semester, students will study modules worth up to 30 ECTS, where ECTS stands for European Credit Transfer System [4]. In the Scientific Route, a student will spend his/her first year (two semesters) at one chosen university, and the second year at the second chosen university. In each year, students will do modules worth up to 60 ECTS. In its first phase of implementation, GSEEM will start with the Scientific Route only, available for the students enrolling in 2007. The Professional Route, which is already operational in some partner universities, will be added to GSEEM in the second phase, within 2 years.

Hence, a GSEEM student will have to select one Route and one Profile. Each Profile consists of a set of core and optional modules, together with a Master thesis project. Core modules are obligatory, while students can choose a specialised set of optional modules according to their interests. A Profile is split into two parts, delivered across two partner institutions. Each partner contributes to at least two Profiles, by offering

modules and Master thesis projects. The GSEEM programme offers flexibility of choice in options, with about 60 different modules. In Table 1, we show a typical example of selected modules for the Scientific Route and the SA Profile at two different locations.

4. Global Approach

4.1. Integration

From a technical perspective the programme offers an overall and integrated knowledge in Software Engineering. Integrated knowledge is achieved by combining specialties of partner institutions in Web Services and Embedded Systems with Software Architecture as the integrating factor. Integration of these areas is a clearly visible trend in industry and research. Individual institutions would not be able to offer such integration, but only partial knowledge in each area, while lacking a complete vision. Integration will allow complementing partial knowledge, while providing deeper investigation of important themes. By integration of expertise, the partner institutions have built a programme that provides an integrated knowledge base aligned with industrial and research trends.

4.2. Shared modules

From a global perspective, the GSEEM students will experience the technical challenges and the social and cultural diversity of global development: the modules with accompanying projects will be carried out in an integrated way across multiple universities, either by running them in parallel and letting the students interact on a distributed network, or through mobility by letting the students bring the technical and cultural experience gained at one university to a second one.

The consortium will realise the innovative concept of *shared modules*, each delivered jointly by two or three institutions, and result in both students and teachers from different universities constituting a common team. Some modules will intensively use distributed teaching and teachers' mobility. Communication, collaboration, and coordination are three main challenges in global SE. Co-located teams can collaborate using traditional data sharing devices (e.g., whiteboard, projector, flipboard) and computer supported cooperative tools (e.g., shared knowledge database, project management tools, groupware).However, in a global and distributed context collaboration becomes more difficult and novel means are needed [7] In the GSEEM context, teachers and students will use innovative tools to participate in the shared modules (this method has been successfully tested in the ongoing collaborative teaching at MdH and is starting at both VU and UDA). In this way, we integrate an international dimension into the teaching and related research.

Four types of shared modules have been designed (see Fig. 1):
- *parallel* twin modules run remotely between universities,
- *shifted* modules, teaching SE concepts incrementally with shifts in study locations and timeline,
- *complementary* modules in which complementary SE concepts are taught in parallel through shared projects, and
- *common* modules, which share the presentations and the projects. The presentations are performed on site and sent on-line to other sites.
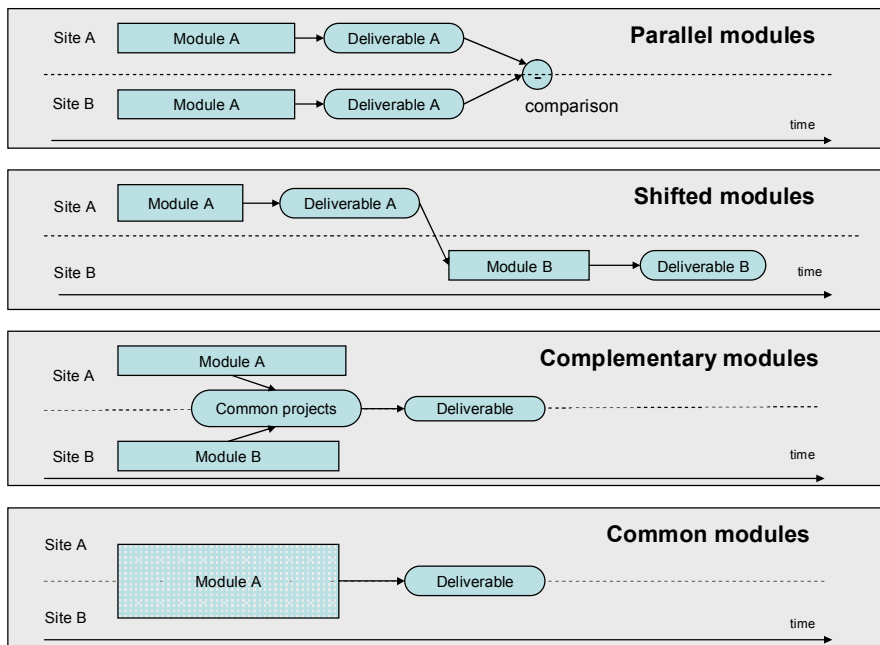
Fig 1: Types of shared modules

The shared modules approach enables learning technical and cultural aspects of SE by working in internationally distributed development teams. The GSEEM students learn both 'technical' Software Engineering and Software Engineering in a global setting. In other words, the programme combines the technical profiles demanded by the European IT market with the global knowledge demanded worldwide. We aim at educating software engineers who are global citizens aware of cultural, geographical, economical and environmental differences, and can sensitively exploit this diversity in their profession. The teachers can also experience diversity in education by participating in teaching different modules at different places.

5. Experience with Shared Modules

We are in the early phase of implementation of the programme – however, each partner institution has ample experience in global education, especially in the implementation of the complementary and common modules.

5.1. Common module

For the last four years the common module Distributed Software Development [9] has been successfully performed between MdH and the University of Zagreb, Croatia. Between 15 to 20 students participated from each site. The module has been designed as a combination of lectures, guest presentations and distributed projects, executed jointly by help of Internet-based tools. The students are split in distributed project groups, in which three to four students from each site participate. Interestingly enough, the module has shown to be especially attractive for international guest students; usually students from five to ten different nationalities participate in the module, which provides an additional multicultural dimension to performing software projects. The experience (based on the project results, the inquiries and student interviews) has indicated certain

difficulties in coordination of distributed projects, especially in the early phases, and some technical limitations in using Internet- and web-based tools. However, any problems were outweighed by higher student motivation for cooperation thus resulting in deliverables of higher quality than expected. From the anonymous inquires made we have learnt that the students are very enthusiastic about the module, not only about the subject itself but also that they collaborate with people they never met in person. On two occasions several students visited other site on their own initiative, which clearly illustrated their enthusiasm. More information about the module (and students' evaluations) can be found in [3].

5.2. Complementary module

A complementary module, composed of modules 'Modeling of Web Applications' (UWA) and 'Advanced Topics in Software Design' (ATSD), is currently taught at UDA and VU. Its main objective is to teach complementary SE concepts in parallel through common projects, and let the two student populations experience the issues around global development. The UWA module, for instance, teaches how to use the UML for Modeling Web Applications. It covers 12 lectures teaching plain UML, and 12 lectures on specific Web modelling methods. Much attention is given to the quality and thoroughness of the delivered design, the ample usage of the learned modelling concepts, and traceability among different models. The ATSD module, instead, includes three lectures providing the basic knowledge about effective design documentation, design rationale and rationale for the selection of UML diagrams, and five additional lectures on specialised design topics, among which is Web design. Much attention is given to the quality of the design document, documentation of the rationale, correct and motivated use of the selected topic. The common project requires both UWA and ATSD students to work in teams on the same project, built to let them learn complementary SE topics from both UWA and ATSD modules. Experiments carried out in the last two years have shown that such complementary modules improve traditional teaching in two ways:

- Learning by doing: by working in joint UDA-VU teams, the students can put theory into practice and learn the contents taught in the other module. Secondly, they gain experience in how global design is carried out in a distributed setting (use of tools for distributed collaboration, effective documentation and communication). Lastly, they gain insight in the core issues specific to global software development, like working with different cultures.
- Learning by osmosis: by collaborating in international projects, the complementary SE contents are transferred more effectively between the two student groups, and the learning period is shortened.

6. Practical implications

The GSEEM students will study at two universities. After one semester (the Professional Route), or after one year (the Scientific Route) they will move from the first university to the second according to the selected profiles and modules.

For each student, the programme leads to two officially recognised Master Degrees, one issued by the university at which the student is enrolled and the other from one of the partner universities.

The GSEEM students will acquire knowledge, competences and experiences in engineering software and software-intensive systems in a global setting, which implies

the ability to cope with complexity of understanding, designing and implementing such systems in the global market place. This comprises techniques from SE disciplines, including requirements engineering, software processes, software architecture and design, system analysis, testing, verification and validation. In addition to this, the GSEEM students will specialise in one or more of the following fields: web and service engineering, component- and model-based engineering, real-time embedded systems, industrial controllers, and mobile applications.

The GSEEM students will be educated as global citizens. With specialised modules in research methodologies and professional ethics being part of the GSEEM curricula we expect to create an impact on the quality of our graduates and societal relevance of their education. The GSEEM students will be aware of, and trained to work with diversity (e.g. cultural, social, and economical), know how to communicate in a global network and a global team, interpret diversity and exploit it in their professional and personal lives.

7. Conclusions

Software Engineering is a broad discipline and in a Master programme it is not possible to cover all its facets. Two are the risks: either covering too many facets too superficially, or focusing on particular facts and neglect some essentials. The GSEEM programme overcomes these risks by offering more modules and keeping them tightly integrated. From a technical perspective, the GSEEM provides a broader state-of-the-art knowledge than it would be possible to offer in each individual university. From a global perspective, it offers its students with an opportunity to experience the technical, social, and cultural challenges of global development.

8. References

[1] A. Berglund. "Learning computer systems in a distributed project course. The what, why, how and where". PhD thesis, Uppsala University, Department of Information Technology, 2005.

[2] ACM/IEEE, Computing Curricula 2005, http://www.acm.org/education/curricula.html, 2005.

[3] Distributed Software Development Course, http://www.idt.mdh.se/kurser/cd5610/2006.

[4] ECTS - European Credit Transfer and Accumulation System, http://ec.europa.eu/education/programmes/socrates/ects/index_en.html.

[5] The GSEEM – An Overview, http://www-gseem.eu

[6] IEEE Software, Global Software Development, Vol. 23, No. 5, Sept/Oct 2006.

[7] IEEE International Conference on Global Software Engineering (ICGSE '06), http://www.icgse.org, 2006.

[8] Erasmus Mundus, http://ec.europa.eu/education/programmes/mundus/faq/faq1_en.html

[9] I. Crnkovic, I. Cavrak, J. Fredriksson, R. Land, M. Zagar, M. Åkerholm, On the Teaching of Distributed Software Development, 25th International Conference Information Technology Interfaces, IEEE, Dubrovnik, Croatia, June, 2003.

[10] E. O. Navarro and A. van der Hoek. Scaling up: How thirty-two students collaborated and succeeded in developing a prototype software design environment. Proc. of the Conference on Software Engineering Education and Training, pages 155–162, 2005.

[11] P. Inverardi and M. Jazayeri: "Software Engineering Education in the Modern Age", Software Education and Training Sessions at the International Conference on Software Engineering, ICSE 2005, St. Louis, MO, USA, May 15-21, 2005, Revised Lectures, Springer 2006.

[12] A. I. Verkamo, J. Taina, T. Tuohiniemi, Y. Bogoyavlenskiy, and D. Korzun. Distributed crosscultural student software project: A case study. Proc. of the Conference on Software Engineering Education and Training, pages 207–214, 2005.

[13] W. Aspray, F. Mayadas and M. Y.Vardi (ed), Globalization and offshoring of software, Report of the ACM job migration task force, 2006.

[14] P. Bourque and R. Dupuis, eds., Guide to the Software Engineering Body of Knowledge, IEEE CS Press, 2004.