A Professional Software Engineering Concept 
and its Interpretation for the United Arab Emirates

Abstract: A study program philosophy for Professional Software Engineers will be outlined oriented towards current deficits and software technical requirements in industry, and considering important aspects of the occupation of future software engineers. In the educational concept the view is taken that the fundamentals of the analysis, modeling, realization and adaptation of software systems are based on the existence of models, technical descriptions and means of expressions. The development of software systems is considered as an overall engineering, model-based process oriented on organizational facts and market needs. Thus by developing software systems a product character is being obtained which has to be reflected in teaching subjects, study programs, technology and knowledge transfer. Since software products cover numerous facets and aspects of complex systems, the training of professional software engineers has to cover overall construction methodologies. To satisfy market needs, professional subjects, software and hardware system labs are defined and described. The Professional Software Engineering (PSEC) merges study complexes and labs into an integrated professional software engineering study program based on international standards. A proposal for specializing professional software engineers in the United Arab Emirates (UAE) is outlined in the appendix.

This draft version recognizes the increasing level of Universities offering Software Engineering Curricula as well as results of characterization efforts, e.g. [SWEBOK2001].

Related major SCI2002 themes: Position Draft on Information Systems Development (sub section on Software Engineering)

Software Engineering as a Professional Engineering Discipline

“Software engineering is the application of principles, methods and techniques to the analysis, design, prefabrication, development and adaptation of programs and systems of programs.”

As an interdisciplinary work and research field Professional Software Engineering (PSE) requires extensive knowledge of various disciplines such as formal semantics, logics, security, software architecture, computer hardware, distributed programming, middleware technology, coordination and planning, decision theory and business organization. In addition to its interdisciplinary character and knowledge from various disciplines, the development of complex modern application systems makes great software technical demands on mobility, adaptability, complex and interoperable interactions, open and human-centered systems. Here it becomes obvious that the practical application of modern technologies is ahead of its engineering substantiation. Examples of the existing deficits are

- non-uniform, non-standardized means of expressions and descriptions
- insufficient technical communication skills
- non-uniform, non-standardized definitions, applications of software building blocks and methodologies
- non-existing, non-standardized order, abstraction layers of building blocks, systems and interactions
- insufficient exploration of approaches and limitations of alternatives for developing systems
- insufficient frame works and platforms for powerful, interoperable and secure application systems
- insufficient development of tools for the intuitive use and handling of modern application systems
- missing concepts of the pre-fabrication and re-use of building blocks, systems and interactions
- no predictability of interactions and dependencies between building blocks, systems and organizations
- ...
The insufficient software technical foundation, a lack of logical levels and substantiated approaches for an engineering analysis, modeling, development, validation and verification of application systems and their building blocks necessitate an engineering treatment and, furthermore, an engineering-based training of future PSE graduates. In analogy to traditional engineering approaches, such a study program has to be based on the teaching of standards, development frameworks and the re-use of pre-fabricated building blocks and partial systems. Thus, the development process of complex systems has to be taught on the basis of reference architectures and solutions as well as on the basis of experience and abstracted knowledge of developers.

**Software systems are built into architectural frames**

Software systems are characterized by complexity, openness, diversity, etc. and require substantiated engineering approaches for modeling, implementation, validation and adaptation. By applying engineering methods, software systems obtain a product character since requirements, models, architectures, specifications and implementation prescriptions are being delivered as parts of the product with every phase of its life cycle. Software products are based and built according to standardized means of expressions and methodologies [HORN2000]. This general orientation has strong implications and has to be reflected in the study program:

*Learning engineering approaches and methodologies such as*

- system theoretical and application technical treatment
- abstraction layers, composition and delegation
- allocation of model types to problem space and solutions
- architectural frames and architecture-based development
- paradigms and principles
- team work
- ...

**according to the characteristics of software systems**

- huge complexity
- (mass of) people involved
- multi-lingual, -paradigm,
- openness (heterogeneity, distribution, evolution)
- ...

The training of Professional Software Engineers has to be oriented towards engineering methodologies and market needs according to the substantiated use of latest software technologies. These requirements are to be integrated into the occupation. Typical practices can be specified for:

- model-based analysis, design, implementation and validation/verification
- team-oriented practicing
- transformation of user-specific technologies
- identification of key solutions to problem space
- product release and marketing plus technology transfer
- ...

![Figure 1: Architectural pattern (see [HORN99])](image_url)
PSEC study program

“Solving engineering problems by practising theoretical, mathematical and engineering principles within an overall product life cycle”

PSEC outlines a study program philosophy oriented towards software technical requirements and their engineering substantiation as well as towards the occupation image of future PSE degree holders. PSEC comprises the creation and application of theoretical and engineering fundamentals for models, methodologies, instruments for systematic and team-oriented analysis, development (consisting of construction, implementation, test), application, maintenance, validation and verification of complex software systems. PSEC focusses on:

- abstraction layers and model transformation / mapping
- building block methodologies and abstraction levels
- constructive (meaning operational) view of software systems
- architectural frames and patterns for classes of application systems
- aspect-oriented de-composing of software systems
- measurement of couplings, throughput, performance, building block reliability
- ...

Figure 2: Model-based methodology

Figure no 2 shows the model-based approach depicting the discourse in abstraction layers onto different types of models (analysis, construction, implementation, verification/validation) and interpreting them using a development environment. This environment consists of application software (e.g. CASE, word processor), pre-fabricated components (e.g. class hierarchies), standard technologies (e.g. databases, operating systems) and hardware. Constructing software systems is an iterative and evolutionary development fulfilling verification and validation prescriptives and criteria.

Due to the engineering substantiation of software technical skills based on standards, architectural frames and engineering practice a stringent training concept is formed by PSEC.

Administrative Organization

Since engineering practice is seen as the basis of PSEC, the educational program is deeply influenced by model-based, architectural and constructive concerns. To offer high flexibility, PSEC’s administrative and organizational concept is oriented towards international standards like

- openness of study program using credit point system
- broad training enabled and enforced by interdisciplinary concepts
- internationally recognized degrees (B.Sw.E., M.Sw.E.)
- stay abroad, multi-lingual study life
- environment of labs, equipment, software, tools, literature, library
- interactions and programs oriented towards market needs
- ...
Interaction with the market is especially beneficial to determine technology developments, trends and future engineering skills applied to system construction. Examples of interactions are:

Q: What are features of future application systems?
   A: Interoperability, mobility, openness, anthropomorphic, adaptation, workflow functionality, ...

Q: What tasks do future PSE degree holders have to complete?
   A: Team management, analysis, design and re-engineering of complex systems, system development as one integrated process

In addition to the orientation towards market needs, practical training is one of the major issues of PSEC. Thus training, internships and projects are integrated dealing with

- huge complexity and interdependent processes
- passing through all phases of a life cycle
- team work character involving more than 20 members and distributed locations
- industrial representatives and product character
- ...

Below, study complexes [HORN98] of the PSEC program are introduced and defined.

**Study complexes and lectures**

A study complex is understood as a field of PSE knowledge. Several lectures are allocated to each study complex. Through the planning of study complexes a flexible organization of the study program is achieved. In the first semesters the allocation of lectures to study complexes can be fixed. In higher semesters several lectures can be allocated to one study complex ensuring the freedom of selection within semesters and study complexes. At the same time the open allocation of lectures to study complexes offers the opportunity to

- adapt the study programs to available resources (instructors, projects, students)
- flexibility according to new research results and lectures
- allocate and recognize study results from abroad
- integrate external specialists (engineers from practice, other universities, …)
- ...

Every study complex consists of multiple lectures but is open to numbers and specifications. An exemplary allocation of lectures to study complexes will be outlined.

**Ethics**

An introduction to PSE and Computer Science is given based on an overview of subjects, working fields, description of means and instruments, study programs, study forms, communication skills and styles. Ethics is due to enforce the motivation and orientation of students.

**Mathematics**

Subjects are the partial fields of mathematics related to engineering disciplines in general and to PSE in particular. Examples used in training should cover PSE-relevant content and discourse.

**Lectures:**

- common algebra
- algebra and geometry
- discrete mathematics
- logics
- probability theory and statistics
- analysis and numerical mathematics
**Fundamentals of theoretical Computer Science**
From the point of view of theoretical Computer Science PSE specialists should be able to:

- define a language according to the analyzed context and discourse and to implement the language
- read and describe software building blocks using formal specifications
- map formal specifications of software building blocks onto programming elements and structures
- test an implementation according to its specification (validation)
- estimate the complexity of programs and problem space
- ...

*Lectures:*
- formal languages, machine models
- theoretical foundation of software development
- computability and programming language paradigms
- complexity

**System technical fundamentals**
The description of products and processes using system technical means of expression is commonly used and applied in engineering disciplines. Thus, teaching fundamentals of system techniques is an essential part of the PSEC educational program. A system consists of components related to each other and attached to a unit via connectors. Systems have:

- an external behavior and interact with their environment
- an internal structure which can be analyzed.

Different aspects of the structure and behavior of software systems can be described by various system models. Executing models and simulations of existing or thought ahead (virtual) systems can be examined.

*Lectures:*
- fundamentals of system theory
- coding theory and cryptography
- graphs and their application for system modeling
- system modeling, simulation and optimization

**Technical fundamentals**
Technical fundamentals and practical examples of building blocks of computers, computer architectures and computer networks are taught using system technical models and means of expressions.

*Lectures:*
- computer architectures
- computer communication and networks
- technical building blocks and building block assemblies
- models of hardware and hardware description languages

**Software basis technologies**
Software basis technologies are the partial systems used by PSE specialists to examine and to develop software products. These are operating systems, compiler, interpreter, standard programs for data management, for the representation of data and programs, for computer communication, for searching in networks, for knowledge management, etc.
PSE specialists have to choose and to apply standard basis components and to integrate them into development environments. The development of basis components is domain the of Computer Science graduates and specialists.
Lectures:

Fundamental techniques
- operating systems
- computer networks and protocols
- languages, compiler, interpreter
- databases
- expert systems
- information systems

Complex techniques
- multi-media techniques and tools
- teleconferencing techniques
- workflow modeling and management
- internet and intranet

Analysis of software systems and software artifacts
The aim of this study complex is to impart knowledge of architectures, modes of actions, development processes, operating guidelines of software systems on the basis of existing software products. This education line consists of lectures providing abstracted, ordered and structured knowledge of complex software products and consists of practical sessions held in labs where systems are analyzed by students in teams.

Lectures:
- concepts of system modeling
- software products grading
- re-engineering of software products
- modeling and simulation
- system optimization

Software architecture
Compared to other technical products, software products consist of building blocks and building block assemblies which can be produced and re-used separately, too. Thus software development is the composition of building blocks. Generally speaking, software systems are heterogeneous and distributed. According to [SHAW96] software architecture describes the building blocks software systems are composed of and the interactions of components using connectors (see [SHAW93]). There are various types of connectors, components and systems. In analogy to other engineering disciplines (e.g. mechanical engineering) software product types can be classified and derived from the point of view of software architecture. They are considered and taught as classes of software systems.

Lectures:
- engineering technical fundamentals of software architecture
- basis architecture types
- software building blocks und building block theory
- architecture description languages
- construction theory
- frame works and patterns
- technical support and tools

Fundamentals of programming
This study complex contains fundamental concepts of programming techniques, especially for the programming in the small.
Subjects are

- algorithms and their design
- data structure and data types
- basics of programming paradigms and languages
- typical, representative data structures and standard algorithms
- basics of algorithm grading
- know-how for testing program fragments
- ...

Lectures:
- algorithms, data, programs and standards
- parallel processes and parallel processing
- program tests

Models, methods, paradigms
Theoretical fundamentals, models, methods and tools for the development of complex software products are part of this study complex. It is built upon knowledge acquired in the following lectures:
- basis technologies
- software project
- concepts of system modeling
- software quality assurance
- software architecture
- human computer interaction and software ergonomics

As common in all working processes division of labor is necessary for software development. A horizontal as well as vertical division of labor has to be reflected within software development:

- horizontal: according to parts of systems to solve data processing tasks (e.g. data management, user interface, error and exception handling)
- vertical: according to application cases and fields (e.g. planning production systems, materials administration, accounting)

This division of labor is reflected by the study complexes ‘software architecture’ and ‘basis technologies’.

Lectures:
- models of software development
- methods of software development and tools
- software construction and design
- processes of software development (life cycle)

Software documentation and standards
Principles of software documentation are taught. Students will be introduced into software standards.

Software management
Specific models and methods are taught for the management of software projects.
Software quality and quality assurance

Software as a product is characterized by specific quality features. These features have to be specified for the product development in the target specification and to be proven at the product or its pre-released prototype. As common for any other technical product standards are determining and regulating software quality and quality management. These standards have to be implemented and realized in a product-specific manner for the development of the software.

Lectures:

- software quality and standards
- testing of software systems
- security and reliability of software systems
- error and exception handling
- human-computer interaction
- measurement and evaluation of software systems
- software and system ergonomics
- software quality management

Allocation of study complexes to education lines

Education lines are oriented towards software products and not towards the partial education fields of Computer Science. Thus the PSE study program differs substantially from this of Computer Science (compare [PARNAS98], [McCONNELL99]). As education lines will be introduced:

- theoretical fundamentals of PSE
- system technical fundamentals of PSE and software analysis
- technical fundamentals of PSE
- software basis systems
- software and software development
- software product aspects.

Theoretical fundamentals of PSE

Theoretical fundamentals of PSE comprise mathematical fundamentals of PSE and relevant subjects of theoretical Computer Science.

System technical fundamentals of PSE and software analysis

PSE as well as other engineering disciplines are based on common system techniques and theory. Since these are of great importance an own education line is being formed. The core of this education line is software analysis, too.

Technical fundamentals of PSE

This education line deals with subjects of technical Computer Science relevant to PSE.
**Software basis systems**
To realize software products, PSE specialists make use of numerous program systems interpreted by the basis machine of a computer. In contrast to the study program for computer scientists, PSE specialists are not taught to be the developers of such systems but their users.

**Software and software development**
An important aim of an engineering discipline is the development of products. This education line teaches knowledge about parts of software products, technical aspects of product development and product development processes.

**Software product aspects**
This education line deals with knowledge about important aspects of software and software products in social working processes and about the environment of software products.

Study complexes are allocated to the introduced education lines as depicted in figure no 3.

![Figure 3: Allocation of study complexes to PSE education lines](image)

To satisfy market needs and improve the technical skills of future PSE degree holders (compare [DIAZ98]) practical exercises and laboratories are integrated into PSEC.

**Enforcing technical skills**
To improve technical skills and to enforce the constructive part of PSEC, practical training will be held in system laboratories. Labs are specialized according to education lines and thus are classified into software system and hardware system labs.

**Software system lab (software development studio)**
The software system lab serves the application of acquired knowledge for the analysis, development, adaptation and further development of large software products.

By realizing complex practical work requiring the application of knowledge from various study complexes and lectures, an important role played by the software system lab. It is recommended to have several software system labs established to initialize, fundraise and coach appropriate projects.
PSE recommends the establishment of three software system labs:

- Software system lab I: analysis and re-engineering of software products
- Software system lab II: development of software products
- Software system lab III: marketing of software products

PSE students take part in a one-term software project held in the software system lab II. According to industry’s needs all phases of a product life cycle are passed through by developer teams under real market conditions.

**Software system lab I: analysis and re-engineering of software products**

The aim is the examination of existing and large software products using various techniques (e.g. techniques of system modeling, software architecture, software ergonomics), the representation of analysis results as well as the re-engineering of system components or partial systems. Partners from industry interested in analyzing and re-engineering their software products and capable to support such projects have to be found. Projects must result in economic benefits.

**Software system lab II: development of software products**

The aim is the development of software products using acquired knowledge from study complexes. As a starting point a task to develop a new software product or to adapt an existing product will be assigned by an industrial principal. All phases of development starting with the compilation of target specification up to releasing the product to the principal have to be passed through by development teams. Issuing documentation is included, too. The software development project has to realize vertical and horizontal division of labor as well as division of roles (e.g. programmer, people involved in quality assurance) in a practice relevant manner.

**Software system lab III: marketing of software products**

A marketing concept for a given software product has to be specified and realized. Parts of the concept are

- product advertising
- product presentation on the net
- preparation of and participation in fairs and exhibitions
- development of exemplary application cases
- compilation of user manual
- selling strategies, contract and license management
- coaching and market introduction/implementation
- hotline organization
- ...

Software system lab III deals with software projects from lab I and II or with given products by an industrial principal. The sequence of students’ passing the labs is according to the number of the lab.
**Hardware system lab (hardware performance studio)**

The hardware system lab serves the application of acquired knowledge of technical aspects of software products. In contrast to software labs, students apply components and partial systems to realize complex and embedded software products. PSE recommends two hardware system labs:

- Hardware system lab I: testing, analysis and composition
- Hardware system lab II: distributed and mobile products

**Hardware system lab I: testing, analysis and composition**

The aim is to impart technical fundamentals by practical training focusing on the application of hardware building blocks and building block assemblies for the realization of software products. Practical work comprises the analysis and testing of hardware components, the composition of components to complex systems and the use of peripherals (e.g. robots).

**Hardware system lab II: distributed and mobile products**

The aim is the handling of distributed, heterogeneous and mobile systems (e.g. satellites), practical fundamentals of embedded systems, computer communication, networks, security (e.g. intranet).

In figure no. 4 the study complexes, software and hardware system labs are allocated to education lines and semesters in an exemplary manner.

<table>
<thead>
<tr>
<th>semester</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>education line</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>theoretical foundations of PSE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mathematics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fundamentals of theoretical computer science</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>application techniques</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>system-technical foundations of PSE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>system-techn. fundamentals</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>concepts of system modeling</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software basis systems</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fundamentals of programming</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>basis technologies</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software system lab I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>technical foundation of PSE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>technical fundamentals</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hardware system lab I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hardware system lab II</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software and software development</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>models, methods, paradigms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software architecture</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software development</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software product aspects</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>introduction into PSE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software management</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software quality assurance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software management</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software system lab III</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>project preparation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software documentation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software architecture</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software system lab II</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software basis systems</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fundamentals of programming</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>basis technologies</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software system lab I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>technical foundation of PSE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>technical fundamentals</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hardware system lab I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hardware system lab II</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software and software development</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>models, methods, paradigms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software architecture</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software development</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software product aspects</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>introduction into PSE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software management</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software quality assurance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software management</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software system lab II</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>project preparation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software documentation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software architecture</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software system lab I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>technical foundation of PSE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>technical fundamentals</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hardware system lab I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hardware system lab II</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software and software development</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>models, methods, paradigms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software architecture</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software development</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software product aspects</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>introduction into PSE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software management</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software quality assurance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software management</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software system lab I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>project preparation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software documentation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software architecture</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software system lab II</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>technical foundation of PSE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>technical fundamentals</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hardware system lab I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hardware system lab II</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software and software development</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>models, methods, paradigms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software architecture</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software development</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>software product aspects</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Specialization and substantiated experience of future PSE graduates

In addition to study complexes, software projects and labs, an increasing complexity and market needs justify the specialization of future PSE degree holders. This specialization demand is reflected by PSEC providing M.Sw.E. students with substantiated experience according to:

- roles in software projects and labs (horizontal, vertical)
- functional and non-functional aspects of software products (e.g. security, coordination specialists)
- selected classes of software products (e.g. embedded, workflow or multi-media systems).

The specialization of future PSE graduates is not based on the restriction or limitation of study complexes but is realized by the interpretation of overall fundamentals, methodologies and knowledge for specified project roles, product aspects and classes of software products.

Conclusion

PSEC outlined a study program philosophy oriented towards software technical requirements in industry and market needs. Thus important aspects of the PSE graduates occupation image were considered and evaluated concerning their engineering substantiation. According to existing deficits and to software technical demands, a special study program was introduced and explained in detail. To offer high flexibility, lectures of the educational program were assigned to study complexes. To satisfy the market needs, software and hardware system labs were proposed and described. The study complexes and labs were merged into an integrated professional software engineering study program based on international standards.

The concept of PSEC is ready to be integrated into international discussions as well as into standardization and accreditation programs: Having agreed upon PSEC study complexes, education lines, software and hardware system labs, this study program can be established focusing on the application and adaptation of substantiated engineering traditions, habits, methodologies and approaches. To train M.Sw.E. students, a specialization according to roles, functional and non-functional aspects and classes of software products is recommended.

Reference


[HORN98] Horn, E. (1998) “Classification and delimitation of the study program ‘software systems engineering’ at University of Potsdam”.


Specializing Professional Software Engineering for the UAE

According to PSEC, specialization is not understood as restricting fundamentals but as the interpretation of fundamentals for selected classes of application cases at the end of B.Sw.E. (5th to 7th semester) and as full part of M.Sw.E. study programs. The advantage of restricting to representative application cases is motivated by the availability of experience and abstracted domain knowledge (semantics) on processes, components and their relations.

Assignment of application classes

In United Arab Emirates available semantics is existent for following, representative classes of application systems and can be prepared and taught in the study complex ‘application techniques’

- computer aided workflows in mineral and petroleum industry
- workflow and information systems in manufacturing industry (chemistry)
- internet-based systems for traveling services and banking industries
- …

Assignment of tasks to study complexes and semesters

As typical characteristics all these classes of software systems are complex, distributed, heterogeneous and interoperable. They enable real-time structured, partial-structured and dynamic (in terms of ad-hoc) workflows and have multiple requirements e.g. on openness like platform and operating systems independence, evolution of architecture types and frames at run time, integration of new building blocks, scalability. Thus substantiating application classes following tasks can be proposed for PSCE-UAE and assigned to study complexes as well as to terms

- abstraction and examination of special properties as aspect models
- prefabrication of aspect models (building blocks with connectors)
- interpretation of aspect models for classes of application systems
- refinement of aspect models
- prefabrication as run time libraries for platforms
- construction set and operations for integrating aspect models
- …

Assignment of topics to study complexes

Abstracting application classes, requirements and tasks special fields of PSCE-UAE-related interests can be identified and assigned to study complexes

- object-oriented model types of special building blocks
- architecture types for classes of application systems
- hierarchical order of architecture types
- methods of the architecture type-based development
- analysis of heterogeneous systems (interoperability)
- security, coordination aspects in open, distributed systems
- platform meeting substantiation criteria
- basis for implementing aspects of application systems
- …
Assignment of duties to PSEC-UAE actors

PSEC is focusing on interdisciplinary co-operations with related industry to develop application systems in an experienced and substantiated way and to open up as well as to develop new fields of application systems (e.g. e-commerce, mobile software infrastructures) re-using knowledge, prefabricated building blocks and specializing them according to semantics. A cooperation proposal is given assigning tasks to industry and academia (in terms of duties and activities)

<table>
<thead>
<tr>
<th>academia</th>
<th>industry</th>
</tr>
</thead>
<tbody>
<tr>
<td>overall project methodologies *)</td>
<td>overall project methodologies *)</td>
</tr>
<tr>
<td>- …</td>
<td>- review of existing software systems</td>
</tr>
<tr>
<td>- historical, technical overview and</td>
<td>- open up new fields of application for</td>
</tr>
<tr>
<td>preparation of appropriated technologies</td>
<td>software systems</td>
</tr>
<tr>
<td>- abstraction of semantics</td>
<td>- establishment of customer-specific</td>
</tr>
<tr>
<td></td>
<td>specification, verification and validation</td>
</tr>
<tr>
<td></td>
<td>techniques</td>
</tr>
<tr>
<td>depiction of semantics onto appropriated</td>
<td></td>
</tr>
<tr>
<td>model types</td>
<td></td>
</tr>
<tr>
<td>- foundations of aspect models (e.g.</td>
<td>- verification, validation of aspect model</td>
</tr>
<tr>
<td>mobility, security, coordination)</td>
<td>types</td>
</tr>
<tr>
<td>- mapping of singular aspects onto an</td>
<td>- verification, validation of construction</td>
</tr>
<tr>
<td>integration model type</td>
<td>methods</td>
</tr>
<tr>
<td>- construction set for selected application</td>
<td>- verification, validation of building blocks</td>
</tr>
<tr>
<td>systems</td>
<td>and architecture type-related activities</td>
</tr>
<tr>
<td>- methods of the architecture type-based</td>
<td></td>
</tr>
<tr>
<td>development of application systems</td>
<td></td>
</tr>
<tr>
<td>co-operative development of application</td>
<td></td>
</tr>
<tr>
<td>systems (A, …, D) on the basis of scientific-</td>
<td></td>
</tr>
<tr>
<td>engineering findings</td>
<td></td>
</tr>
<tr>
<td>substantiating new technologies and aspects (e.g. blocks, middleware, distribution, mobility, …)</td>
<td></td>
</tr>
</tbody>
</table>

*) Despite of application-specific project coordination methodologies, the establishment of unified means of expressions for architecture type-based and frames modeling, aspect model types, platform modeling and refinement as well as for project co-ordination has to be done according to latest findings in theory, their application-technical approval and relevance.

PSEC-UAE goal orientation

Cooperation is goal-oriented. Academic education and product releases shall benefit from PSEC-UAE. The following results are content and to be expected within PSEC-UAE for both, academia and industry community

- development and substantiation of a software engineering development method for concurrent, mobile real-time A, …, D application systems
- reference architecture frameworks for classes A, …, D
- building blocks and their environment according to A, …, D semantics and specifics
- libraries of building block and property classes for application systems (A, …, D)
- experiences with practical applications for A, …, D systems
- substantiation of mobility, security, interoperability and coordination media for A, …, D
- proposals for and participation on standards for A, …, D
- A, …, D product modeling, developing, documentation, releases
- …