

الوصف المختصر لمساقات الخطة الدراسية بمسارها الامتحان  
الشامل والرسالة لبرنامج الماجستير في هندسة البرمجيات

## **Software Processes Management – 721711**

### **Aims:**

This course aims to provide students with knowledge about: Description of commonly used software life cycle process models and the content of institutional process standards. Definition, implementation, measurement, management, change and improvement of software process, and use of a defined process to perform the technical and managerial activities needed for software development and maintenance.

### **Keywords:**

Software Engineering conventional processes overview, Software Processes Modeling Concepts Software Process Meta modeling concepts, Software Process Management , Software process Evolution.

### **Textbooks:**

Münch, J., Armbrust, O., Kowalczyk, M., Soto, M. Software Process Definition and Management. Springer, 2012.

## **Requirement Engineering – 721712**

### **Aims:**

This course aims to provide the state of art for research and practice in requirement engineering. It covers in depth the entire requirement engineering life cycle (activities, methods, techniques, documentation, tools, modeling languages..).

### **Keywords:**

Requirement engineering life cycle, system/software requirements, state of art in requirements engineering activities (elicitation, specification, analysis, modeling, validation); Functional, Non Functional and Quality Requirements; Goal oriented requirement engineering approach.

### **Textbooks:**

- 1- Requirement Engineering for Software and Systems, Philippe A. Laplante, CRC Press, 2014.
- 2- Requirements Engineering, Elizabeth **Hull**, Ken **Jackson**, Jeremy Dick, Springer, 2011.
- 3- Requirements Engineering: From System Goals to UML Models to Software Specifications, Axel van Lamsweerd, Wiley, 2009.

## **Software Project Management - 721713**

### **Aims:**

Software developers moving into a team leader or project management role. Software project managers who want to move from traditional approaches to agile processes. Software project managers preparing a PMP certification. Experienced project managers with little or no experience in software development.

### **Keywords:**

Project Evaluation, Project Organization, Software Estimation, Activity Planning, Management issues.

### **Textbooks:**

- 1- Agile Project Management: Managing For Success, By James A. Crowder, Springer International, 2015.
- 2- Software Project Effort Estimation, By Trendowicz, Jeffery, 2014.
- 3- Effective Project Management: Traditional, Adaptive, Extreme, 3d Edition by Robert K. Wysocki and Rudd McGary.
- 4- Practical Project Initiation: A Handbook with Tools (Microsoft Press, 2007)

## **Software Testing and Quality Assurance – 721714**

### **Aims:**

Every software development organization needs to be focused on the delivery of quality. The software engineering discipline responds by calling for a managed process for the construction and testing of software, and for the improvement of that process. This course explains the necessary concepts within the frameworks provided by three important international standards.

### **Keywords:**

Pre-project software quality components, SQA components in the project life cycle, Management components of software Quality, Standards, certification and assessment, Organizing for quality assurance.

### **Textbooks**

- 1- Wanger, Stefen, Software product Quality Control, 2013
- 2- Murali Chemuturi, Mastering Software Quality Assurance: Best Practice, Tools and Techniques for Software Developers, 2010
- 3- Daniel Galin. Software Quality Assurance: From theory to implementation. Pearson Wesley. 2004
- 4- [G. Gordon Schulmeyer](#) (Editor). Handbook of Software Quality Assurance. Artech house, 2008

### **Useful Link**

<http://users.encs.concordia.ca/~bentahar/inse6260.html>

## Software Evolution and Maintenance - 721715

### Aims:

This course aims to offer advanced skills in Software Maintenance: **Software Change** (Introduction to software maintenance methods and tools: Corrective, adaptive, perfective, and preventive maintenance methods and tools), **Change Management** (Software Configurations Management SCM: Concepts, Repository, Changes set, Workspace, System/product model, Composition, Database, Long transaction, Versioning, tools, researches, ...), **Program Comprehension – Reverse Engineering** (Code reading: Code reading, top down reading, unfamiliar PL code reading; Code Understanding: Software architecture, software understanding tools), **Evolution of Legacy Systems** (Fundamentals and re-engineering by migration: Characteristics of legacy systems, challenges for their evolution, the re-technologies: reverse-engineering, re-structuring, and re-engineering by migration; Restructuring methods and tools: Error avoidance tools, Architectural tools, reuse tools, Application; Change impact analysis methods and tools: Static Analysis, Dependency Graph, Slicing).

### Keywords:

Software Change, Program Comprehension (Reverse engineering), Evolution of Legacy Systems, Look to the future.

### Textbooks:

- 1- Priyadarshi Tripathy, Kshirasagar Naik. Software Evolution and Maintenance. Wiley, 2015
- 2- Tom Mens, Serge Demeyer. Software evolution, Computers, 2008

### Useful Link

[http://intranet.cs.man.ac.uk/Study\\_subweb/Ugrad/coursenotes/CS3331//Manchester maintenance course](http://intranet.cs.man.ac.uk/Study_subweb/Ugrad/coursenotes/CS3331//Manchester%20maintenance%20course)

## **Component-Based Software Engineering - 721716**

### **Aims:**

Component-based software engineering (CBSE) (also known as component-based development (CBD)) is a branch of software engineering that emphasizes the separation of concerns in respect of the wide-ranging functionality available throughout a given software system. It is a reuse-based approach to defining, implementing and composing loosely coupled independent components into systems. This practice aims to bring about an equally wide-ranging degree of benefits in both the short-term and the long-term for the software itself and for organizations that sponsor such software.

Software engineering practitioners regard components as part of the starting platform for service-orientation. Components play this role, for example, in web services, and more recently, in service-oriented architectures (SOA), whereby a component is converted by the web service into a *service* and subsequently inherits further characteristics beyond that of an ordinary component.

### **Keywords:**

Introduction to Component-based SW development, CBSE and reuse, component models, CBSE and object technology, Component Development Lifecycle, CBSE & Software Productivity, CBSE & System Quality

### **Textbooks:**

Component Software: Beyond Object-Oriented Programming (2nd Edition) (Addison-Wesley Component Software), 2011.

## **Research Methods in Software Engineering - 721751**

### **Aims:**

The goal is to introduce the participants to research methods, approaches and processes in Software engineering. It consists of lecture introducing software engineering research and study circle meeting in which the concept if validity is discussed in a small groups based on the part of the course book.

### **Keywords:**

Design science, semantic literature review, case studies, action research.

### **Textbooks:**

- 1- Design Science Methodology in Information Systems and Software Engineering, By Roel J. Wiering, 2014.
- 2- Case Study in Software E, by Per Rouneson, Martin Host, Austen Rainer, Bjorn Regnell. 2012.
- 3- Research Methods in Software Engineering, by Wilhelm Hasselbring, Simon Giesecke, 2006.



## **Project - 721752**

In addition to the taught part, you must complete an extended project. Although the project needs to be an original demonstration of ability and understanding, there is no requirement to advance the state of the art in the field. You need only choose and apply an appropriate selection of existing ideas and techniques—provided that their choice, the process of application, and any outcomes are properly explained. New ideas and techniques are welcome, but they are not expected. The project component involves compulsory attendance at a project module in Oxford, at which you will present and refine your proposal, and attend teaching sessions on research skills, engineering in context, and social, legal and ethical issues.

The results of the project work are presented in a short dissertation of 15,000-20,000 words, or 45-60 pages. This forms the basis for formal assessment of the project, just as the written assignments form the basis for assessment of the taught modules. The dissertation can be submitted at any time during the allowed period of study, although it is usually the last piece of work you will submit before being examined.

You will be assigned a supervisor at the beginning of your period of study. Supervisors can provide advice on all academic matters, including course selection and examination entry. Supervisors will also provide guidance on the choice of a suitable MSc project, and on the preparation of a dissertation.

## **Mobile Software Engineering - 721721**

### **Aims:**

The course aims to provide an in depth knowledge on software development life cycles, methods, techniques, tools and technologies used for the development of current and emergent mobile applications. It spans all software development phases and focuses on phases that need to be tailored to mobile applications. It also covers topics related to up to date mobile and hand held devices, and includes a typology of mobile applications.

### **Keywords:**

Mobile/hand held devices, mobile application type (standalone mobile applications, web based mobile applications, context awareness mobile applications, cloud based mobile applications), Agile processes for mobile applications, mobile UI patterns, mobile programming platforms (Android, ...), mobile application testing.

### **Textbooks:**

- 1- Handbook of Research on Mobile Software Engineering: Design, Implementation, and Emergent Applications , Paulo Alencar (Author, Editor), Donald Cowan (Editor), 2012
- 2- Modern Software Engineering Methodologies for Mobile and Cloud Environments António Miguel Rosado da Cruz, Sara Paiva, (To be published by IGI Global 2015)

## **Web Software Engineering – 721722**

### **Aims:**

As an emerging discipline, Web engineering actively promotes systematic, disciplined and quantifiable approaches towards successful development of high-quality, ubiquitously usable Web-based systems and applications.<sup>[3]</sup> In particular, Web engineering focuses on the methodologies, techniques and tools that are the foundation of Web application development and which support their design, development, evolution, and evaluation. Web application development has certain characteristics that make it different from traditional software, information system, or computer application development. Web engineering is multidisciplinary and encompasses contributions from diverse areas: systems analysis and design, software engineering, hypermedia/hypertext engineering, requirements engineering, human-computer interaction, user interface, information engineering, information indexing and retrieval, testing, modelling and simulation, project management, and graphic design and presentation. Web engineering is neither a clone, nor a subset of software engineering, although both involve programming and software development. While Web Engineering uses software engineering principles, it encompasses new approaches, methodologies, tools, techniques, and guidelines to meet the unique requirements of Web-based applications.

### **Keywords:**

Design and construction of web, Using cutting-edge technologies including(Distributed computing, XML, web services, JSP, ASP.NET, MVC frameworks, AJAX;

### **Textbooks:**

- 1- Kappel, Proll, Reich, retschitzegger 2006, Web Engineering, Wiley [ISBN: 0-470-01554-3]
- 2- Casteleyn, S., Daniel, F., Dolog, P., Matera, M.: Engineering Web Applications, Springer 2009,
- 2- Ivan Marsic, Software Engineering, Rutgers, Book website: <http://www.ece.rutgers.edu/~marsic/books/SE/>, 2012.
- 4- Pan, J.Z., Zhao, Y. Pub. Date, Semantic Web Enabled Software Engineering, July 2014.

## **Embedded Software Engineering - 721723**

### **Aims:**

Software engineering practice and methods for embedded systems, focused around state machines as a unifying formalism for understanding software, hardware, and systems. Embedded software requirement, specification, and analysis. Principles of embedded software architecture and design .Design of concurrent systems. Testing and analysis techniques for embedded systems.

### **Keywords:**

Principles of embedded software architecture and design .Design of concurrent systems. Testing and analysis techniques for embedded systems.

### **Textbooks:**

Software Engineering for embedded systems : methods, practical Techniques, and Applications (Expert Guide) Hardcover –May 6, 2013,Ropert Oshana (Editor); Mark Kraaling(Editor)

## **Software Engineering for Secure Software - 721725**

### **Aims:**

This course aims to teach security issues in different phases of software development (requirements, architecture, design, coding, testing). It also covers threats, vulnerabilities and attacks for Web and Mobile applications.

### **Keywords:**

Secure software development life cycle, Threat Modeling, Risk Assessment, Securing Web and Mobile Applications.

### **Textbooks:**

1- Secure Software Design, Theodor Richardson, Charles Thies, Jhones and Barlett Learning Editors, 2013.

2- Core Software Security: Security at the Source, James Ransome, Anmol Misra, Auerbach Publications, 2013

3- Secure and Resilient Software Development , Mark S. Merkow, Lakshmikanth Raghavan, Auerbach Publications, 2010

## **Software Engineering for Intelligent Systems – 721726**

### **Aims:**

In this course MSc. Students will be introduced to modern software engineering and intelligent systems (IS) development practices. Compared to business software projects, IS applications have different challenges. Their complexities around problem solving, heuristics, control structures, knowledge representation, and learning.

### **Keywords:**

Connectionist Paradigms and Machine Learning; Fuzzy Sets, Rough Sets and Approximate Reasoning; Agent Architectures and Distributed Intelligence; Intelligent Web Computing; design of artificial neural networks, evolutionary algorithms, hybrid computing systems.

### **Textbooks:**

1-Intelligent Systems Design and Applications

Editors: Abraham, Ajith, Franke, Katrin, Koeppen, Mario (Eds.) - 2003

2- Soft Computing and Intelligent Systems Design: Theory, Tools and Applications; 2004; by Fakhreddine O. Karray (Author), Clarence W De Silva

3- S.J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach (3rd edition)*, Prentice-Hall, 2010.

## **Development of Enterprise Resource Planning Systems - 721731**

### **Aims:**

The course aims to develop analytical and critical thinking skills for the development of integrative plans for enterprise-wide systems that optimize enterprise performance by analyzing business enterprise activities, workflow and process to identify problems, weaknesses, strengths, threats, opportunities, stakeholders and entities interacting with the enterprise.

### **Keywords:**

Business patterns, Enterprise patterns, Business services, Business modeling, trends in business applications, SOA, emergent development methods and technologies

### **Textbooks:**

1-Future Business Software: Current Trends in Business Software Development, Brunetti, G., Feld, Th., Heuser, L., Schnitter, J., Webel, C. (Eds.), 2014

2-BUSINESS ANALYSIS: Second Edition  
EDITED BY: Debra Paul, Donald Yeates and James Cadle  
2010 British Informatics Society Limited

3-Business Patterns For Software Developers, Alan Kelly, 2012

4-Practical Solutions for Recurring IT-Architecture Problems

Perroud, Thierry, Inversini, Reto-2013

5-Business Process Modeling: Software Engineering, Analysis and Applications- Editors: Jason A. Beckmann, Pub. Date: 2011 3rd Quarter

<http://www.csus.edu/indiv/v/velianitis/171/Syllabus.htm>

## **Development of Game Systems - 721734**

### **Aims:**

This course will teach the differences between modern software engineering and game development practices. Compared to business software projects, digital games pose markedly different challenges. Their complexities around prototyping, testing, and platforms demand more flexibility from developers when it comes to software specifications and functionality. Even seasoned professionals have to modify their familiar approaches to design and team management while also dealing with dramatic changes in user behavior and technology.

### **Keywords:**

Artificial intelligence, Complex systems, Education and pedagogy, Game development tools, Narrative design, and Player research tools.

### **Textbooks:**

1- Software Engineering for Game Developers, November 8, 2004

by Ph.D. John P Flynt (Author), Omar Salem (Author)

2- Mathematics for 3D Game Programming and Computer Graphics, Third Edition, June 2, 2011,  
by Eric Lengyel

3- Computer Games and Software Engineering, Kendra M. L. Cooper, Walt Scacchi , June 4,  
2015



## **Formal Methods in Software Engineering - 721717**

### **Aims:**

In this course MSc students will be seamlessly introduced to the theoretical issues and practical applications of formal methods within the software development process with the broad aim of understanding how the use of such methods even if pragmatically applied might conduct to the construction of maintainable and high quality software systems while keeping acceptable development costs.

### **Keywords:**

Role of formal specification and analysis techniques in the software development cycle, Program assertion languages and analysis approaches, Formal approaches to software modeling and analysis, Model checkers, Model finders, Tools in support of formal methods.

### **Textbooks:**

- 1- Boca, Paul, Bowen, Jonathan P., Siddiqi, Jawed (Eds.), Formal Methods: State of the Art and New Directions, Springer, 2010
- 2- H. Habrias and M. Frappier (Eds), Software Specification Methods: An Overview Using a Case Study, 2006
- 3- Research papers related to algebraic development techniques, Z, Object-Z, Petri nets, CCS and others.
4. V.S. Alagar, K. Periyasamy: Specification of Software Systems, Graduate Texts in Computer Science, Springer-Verlag, 2 nd Edition, 2011.

**Thesis 721753**

9 Credit Hours

See Regulations of Graduate Studies for Further details.

## **Mobile Software Engineering - 721721**

### **Aims:**

The course aims to provide an in depth knowledge on software development life cycles, methods, techniques, tools and technologies used for the development of current and emergent mobile applications. It spans all software development phases and focuses on phases that need to be tailored to mobile applications. It also covers topics related to up to date mobile and hand held devices, and includes a typology of mobile applications.

### **Keywords:**

Mobile/hand held devices, mobile application type (stand alone mobile applications, web based mobile applications, context awareness mobile applications, cloud based mobile applications), Agile processes for mobile applications, mobile UI patterns, mobile programming platforms (Android, ...), mobile application testing.

### **Textbooks:**

- 1- Handbook of Research on Mobile Software Engineering: Design, Implementation, and Emergent Applications , Paulo Alencar (Author, Editor), Donald Cowan (Editor), 2012
- 2- Modern Software Engineering Methodologies for Mobile and Cloud Environments António Miguel Rosado da Cruz, Sara Paiva, (To be published by IGI Global 2015)

## **Model driven Software Engineering - 721741**

### **Aims:**

This course aims to provide a deep knowledge on Model-Driven Software Engineering which is an approach based on models for building and reasoning about software systems. The course will teach the major concepts in model-driven software development, and the state-of-the art for research and practice in Model Driven Engineering approaches.

### **Keywords:**

Model Driven Engineering, Model Driven Architecture, Models, Meta Models, Domain Specific Modeling languages, Model Transformations, Model verification, Model Evolution and Versioning.

### **Textbooks:**

- 1- Model-Driven Software Engineering in Practice, Marco Brambilla, Jordi Cabot, Manuel Wimmer, Morgan & Claypool Publishers, 2012
- 2- Progressions and Innovations in Model-Driven Software Engineering, Diaz Vicente Garcia, IGI Global, 2013

## **Advance Software Architecture - 721718**

### **Aims:**

The goal of architecture-centric development is the effective , efficient, competitive development of software product. The activities of development are based on architecture. The goal of this subject is to provide both the student and professional a comprehensive treatment architecture-centric development, instructing how to develop products and serving as a reference for the issues of techniques, modeling notations, standards, and methods comprising the approach.

### **Keywords:**

Software architecture modeling, Software architecture analysis, Software architecture implementation, Designing for non-functional properties, Security and trust, Software architecture adaptation .

### **Textbooks:**

1- Software architecture : Foundations, Theory, and practice, by Richard N.Taylor, Nenad Medridovic, Eric D. Dashofy - 2010

2- Software Architecture in Practice, by Len bass, Paul Clements, Rick Kazman - 2013