Software Engineering (MS with Thesis)

Course Requirements

The Master of Science in Software Engineering program consists of at least seven 3 credit hours courses, making a total of 21 credit hours of course work, in accordance with the EMU by-laws. Students accepted in probation status must complete the undergraduate courses assigned to them by the Graduate Committee, and pass these courses, before they can become regular MS students. Graduate courses shall be selected by the student from the Software Engineering graduate curriculum, or from another related graduate curriculum, under the supervision of his/her thesis supervisor, or with the consent of the Graduate program supervisor of the Department.

Thesis Requirements

All MS students are required to prepare and defend an MS thesis under the supervision of a faculty member approved by the Department. Every student must find an MS thesis supervisor, the latest at the end of the first semester of his/her graduate study. The MS thesis should present the results of scholarly investigation of a topic related with the field of Software Engineering. An MS student should complete all the required studies, including courses and thesis work, within the minimum and maximum time set by the EMU by laws.

The courses covered in the program are as follows:

Ref. Code	Course Code	Course Name	Credit
295R0	CMSE500	M.S. Thesis	0
295RS	CMSE598	Seminar	0
295R1	RES1	Restricted Elective Course	3
295R2	RES2	Restricted Elective Course	3
295R3	RES3	Restricted Elective Course	3
295R4	RES4	Restricted Elective Course	3
295R5	REQ1	Elective Course	3
295R6	REQ2	Elective Course	3
295R7	REQ3	Elective Course	3

RESTRICTED ELECTIVE COURSES

CMSE511 Software Architecture

CMSE516 Software Quality Assurance and Reliability

CMSE517 Software System Analysis and Modeling

CMSE519 Contemporary Methods for Software Development

CMPE520 Software Evolution and Maintenance

CMSE522 Formal Specification and Verification of Software Systems

ELECTIVE COURSES**

CMSE512 Database and File Security

CMSE513 User Interface Development

CMSE514 Web Technologies and Services

CMSE515 Testing Web Applications and Services

CMSE518 Big Data Analytics

CMSE521 Software Engineering for Emerging Technologies

** Students may also take Restricted Elective Software Engineering MS courses (CMSE5XX), or Computer Engineering MS courses (CMPE5XX), or other Engineering Faculty MS courses, or at most 1 non-engineering course by the approval of their thesis supervisors.

Course Descriptions

Compulsory Courses:

Course Code: CMSE500 **Course Name:** Master Thesis**Course Description:** Student must submit a thesis based on the results of his/her original investigation of a software problem. The aim of the thesis is either to develop a prototype or an integrated software product using software engineering methodologies, techniques and tools, or to conduct a conceptual/empirical research study on software engineering field (e.g., developing a new Software Architecture & Design Pattern, exploring a new Software Engineering Development Processes & Methodology such as Agile, Waterfall, Spiral model, defining new Software Quality Assurance & Testing Processes and Metrics, developing Software Requirements Eng. & Management Processes, developing the Software Verification & Validation Techniques, developing Software Evolution & Maintenance Techniques, developing Software Project Management & Leadership Processes, etc). The thesis must show a significant style, organization and depth of understanding of the subject. The developed thesis is expected to contribute to the software engineering discipline. Thesis study is conducted with the supervision of a faculty member.

Course Code: CMSE598 **Course Name:** Seminar

Course Description: This course includes seminars which will be given by teaching staff to MS students covering the topics related to research methodologies, thesis writing, presentation, ethics, social responsibility, plagiarism, scientific paper writing and publishing.

Restricted Elective Courses:

Course Code: CMSE511 **Course Name:** Software Architecture

Course Description: Architectural abstraction of software systems. Software elements and how the elements relate to each other. Looking at a software system in terms of its elements, how they are arranged, how they interact, how they are composed, what their functional and behavioral properties are. Describing software in a modular/hierarchical manner. Describing interfaces between modules. Techniques for partitioning modules into submodules. Relationship of hardware architecture and software architecture descriptions of a Software system.

Course Code: CMSE516 **Course Name:** Software Quality Assurance and Reliability

Course Description: Concepts, metrics, and models in software quality assurance. Components of software quality assurance systems at the beginning, during, and after the software development process. Software quality assurance within the stages of the software development process: planning, configuration management, reviews, testing, and maintenance. In these stages, metrics and models for software quality will be discussed. The course will include practical case studies. Students will develop an understanding of how software quality can be assured.

Course Code: CMSE517 **Course Name:** Software System Analysis and Modeling

Course Description: This course is intended for helping students build up an understanding of how to develop a software system by guiding them through the software system analysis and design process. Software systems analysis and design as a problem-solving activity, key elements of analysis and design,

and the place of the analysis and design phases within the system development life cycle will be discussed in detail. Students will be required to apply these techniques in a group design project.

Course Code: CMSE519 **Course Name:** Contemporary Methods for Software Development

Course Description: In software development practices, Agile software development proves to be highly effective when utilized in a collaborative and people-centered organizational culture, especially in scenarios that demand exploratory development efforts, involve complex requirements, and suffer frequent changes. This course explores Agile methods, including Extreme Programming (XP), Scrum, Lean, Crystal, Dynamic Systems Development Method, and Feature-Driven Development, providing an understanding of how software can be developed more rapidly and efficiently. The capabilities of Agile development teams in swiftly creating high-quality software with customer value are examined and compared with teams following more traditional methodologies that emphasize intensive planning and documentation. Students will learn Agile principles and techniques covering the entire software development process from the conception of the software problem to analysis, development, testing, and deployment. Successfully completing this course will equip students to actively participate in Agile software development and effectively manage an Agile team. The course includes practical exercises covering Agile software development project processes alongside theoretical understanding. Additionally, it will touch upon DevOps, Waterfall, and Rapid Application Development in addition to the Agile model.

Course Code: CMSE520 **Course Name:** Software Evolution and Maintenance

Course Description: The course introduces the conceptual definitions of Software Evolution and Maintenance. It delves into fundamental aspects of software maintenance, covering corrective, adaptive, perfective, and preventive maintenance. A detailed examination of the maintenance process and critical concepts like impact analysis and change management is provided. Maintenance process of software developed using different methodologies (ranging from the Waterfall model to more modern methodologies like Agile) is considered. Software aging and decay concepts and techniques for renewing aging software are discussed. This is followed by an exploration of software reengineering, including methodologies related to reverse engineering, forward engineering, and restructuring processes (ideas, processes, techniques, tools). The concept of software refactoring (ideas, processes, and formalisms) is introduced. The course covers configuration management, including the use of version control systems, change

management, and configuration management tools in software evolution. The importance of software maintenance metrics and measurement is discussed, covering key performance indicators. Legacy system migration challenges and techniques are considered. Additionally, the course discusses program comprehension (ideas, terms, cognition models, protocol analysis, and visualization) and reuse (ideas, models). Case studies provide practical insights into software maintenance challenges. The course then explores emerging trends such as DevOps, continuous integration, and microservices architecture, examining their impact on software evolution. Ethical and legal considerations in software maintenance are discussed. The term project allows students to apply the concepts learned.

Course Code: CMSE522 **Course Name:** Formal Specification and Verification of Software Systems

Course Description: Mathematical foundations such as set theory, logic, and discrete mathematics, propositional logic, and first-order logic. Mathematical reasoning and proof techniques. Introduction to formal specification languages such as Z, VDM, and Alloy. Expressing software requirements using formal specifications. Model checking and its applications in software verification. Temporal logics such as LTL (Linear Temporal Logic) and CTL (Computation Tree Logic). Hoare logic and its use in verifying program correctness. Static analysis techniques for program verification. Abstract interpretation and data flow analysis. Verification of safety and security properties in software systems.

Elective Courses:

Course Code: CMSE512 **Course Name:** Database and File Security

Course Description: Security requirements. Confidentiality, integrity, availability, assurance, authenticity, anonymity. Security threats and attacks on security, Protection, Intruders, Access control models. Discretionary and mandatory access control, Authorization. Multilevel security. Malicious software, Trojan horses. Password protection. Symmetric and public key encryption. DES. RSA. Digital signature. Key distribution and authentication. Secure Sockets Layer protocol. Passport: Single Sign-On. Electronic commerce. Secure Electronic Transaction protocol, Goods atomicity, Certified delivery and escrow. Electronic cash. Blind signature, Security in XML-based Web-services. Kerberos. X.509 authentication service. Authentication, procedures. One-time password. Hash functions. Statistical database security,

Functional dependency attack on databases with multilevel security. Flow control, Covert channel, Auditing. Auditing process, Auditing classifications and types.

Course Code: CMSE513 **Course Name:** User Interface Development

Course Description: This course will provide a general introduction to the theory and practice of computer user interface design. In this respect, principles of human factors in computing, cognitive modeling and usability engineering will be covered in association with practical design issues. User interface design phases task analysis, user-centered design, and prototyping will be covered in detail. Design of windows, menus, and commands will be preented based on the design of 2D graphical user interfaces in three environments: standalone, Web and mobile devices. The study of several important paradigms and principles of design and how these can be applied to the user Interfaces will be explored.

Course Code: CMSE514 **Course Name:** Web Technologies and Services

Course Description: WEB 2.0 technologies. Influence of WEB 2.0 over business and society. Web 3.0 and semantic web concepts and technologies. Web 3.0 applications and management of web data. Web services overview. Service-Oriented Architecture (SOA). Web Services Description Language (WSDL). Universal Description, Discovery & Integration (UDDI). Simple Object Access Protocol (SOAP). Service-Oriented Software Reengineering (SOSR). XML technologies. Web services interaction protocol and description with J2EE technologies. Web service discovery and composition. Future trend in web technologies and services. For the programming part of this course, students will use different tools: Client Side (e.g. HTML, CSS, JavaScript, Ajax, jQuery and JSON), Server Side (e.g. Servlets, JSP, Java Beans, JAX-RS for RESTful services), Database (e.g. MySQL) and Knowledgebase (e.g. OWL 2.0).

Course Code: CMSE515 **Course Name:** Testing Web Applications and Services

Course Description: This course will start with a discussion on how modern web applications work. Secure coding practices to develop better web applications will be introduced. White/grey/black-box testing techniques and tools for assessing web applications will be mentioned. Common pitfalls and security

vulnerabilities that affect web applications will be discussed. Mobile application security concerns will be explained.

Course Code: CMSE518 **Course Name:** Big Data Analytics

Course Description: This course examines the basic concepts and practices of big data analytics. Learning objectives of this course involve; understanding of the MapReduce paradigm and Hadoop, developing data analysis skills with Hive and Pig, analyzing temporal, geospatial, text, and graph data via Spark, and also performing supervised/unsupervised machine learning algorithms on large datasets using Mahout (Hadoop) or MLlib (Spark). R, Java or Python language examples will be used for illustration.

Course Code: CMSE521 **Course Name:** Software Engineering for Emerging Technologies

Course Description: This course focuses on how software projects involving emerging technologies are managed based on principles of software engineering and project management methods. It aims to provide students with a broad perspective on various new solutions to overcome challenges that may be encountered in managing such projects.

The course aims to equip students with the necessary knowledge and skills for appropriate software development such as SDLC methodologies, effective management planning, and development completion for efficient development in software projects related to emerging technologies. Thus, students will acquire the fundamental skills required to develop innovative software solutions for real-world problems.

In the initial stage of the course, building upon the foundational knowledge of software engineering principles, processes, development methodologies, and techniques acquired in previous courses, current topics such as artificial intelligence, GenAI tools (e.g., ChatGPT), machine learning, neural networks, Internet of Things, block chain technology, cybersecurity, big data, edge and cloud computing, among others, will be covered. Topics will include planning, execution, and coordination of software projects covering new technologies. Additionally, various software projects involving emerging technologies from the industry and literature will be discussed, providing students with insights into important projects in these areas.

Contact Information

Prof. Dr. Zeki Bayram (Chairman)
Tel: +90 392 630 1484
Fax: +90 392 365 0711
E-mail: cmpe.info@emu.edu.tr
Web: http://cmpe.emu.edu.tr
Program website:
https://cmpe.emu.edu.tr/en/graduate-programs/software-engineering-ms-with-thesis
EMU website: https://www.emu.edu.tr/en