

Template of Course Specification

Faculty: CIT
Department: CS
Program: B.Sc(Software Engineering)

General information about the course :						
1.	Course Title:	Software maintenance and evolution				
2.	Course Code and Number :	SE421				
3.	Credit Hours: 3	Lecture	Seminar/Tutorial	Practical	Training	Total
		3	0	0		3
4.	Study Level and Semester:	Level 4 / Semester 8				
5.	Pre-requisites (if any):	Software engineering (CS325)				
6.	Co-requisites (if any) :	Differential equations(CS221) and Data structures and algorithms (CIT07),Calculus (CIT02)				
7.	Program in which the course is offered	B.Sc(Software engineering)				
8.	Teaching Language:	English				
9.	Study System :	Semester Based				
0.	Prepared by :	Course Facilitator (Bridget Shakesy)				
1.	Approval date :					
2.	Approved by:	Department				

II. Course Description:

Organizations across all industries have huge investments in their software systems. These systems must be continually adapted to the changing needs of the organizations. Software maintenance and evolution refers to the process of modifying existing software systems to maintain their usefulness. Some studies have estimated that up to 75% of all technical software professionals are involved in some form of software maintenance activity. This course builds on basic software engineering concepts by expanding the discussion of the issues involved in maintaining and evolving software systems.

III. Course Aims

- Motivate why it is important to build maintainability in a system
- Plan for the whole software lifecycle with emphasis on risk analysis and maintainability in the initial phases of software development
- Identify problems within the industry and research
- Choose appropriate problem management method. The choice should be based on software criticality and problem urgency.
- Explain differences among all the types of maintenance categories
- Make decisions and manage legacy systems
- Identify challenges associated with software evolution.
- Evaluate agile methods-extreme programming from the lifecycle perspective

VI. Course Intended Learning Outcomes (CILOs) :

Knowledge and Understanding:

Alignment of CILOs (Course Intended Learning Outcomes) to PILOs (Program Intended Learning Outcomes)

Knowledge and Understanding PILOs	Knowledge and Understanding CILOs
After completing this program, students would be able to:	After completing this course, students would be able to:
A1. Understand the principles and theories of mathematics, programming, algorithms and its complexities.	a2. Understanding the similarities and complexities of the software
A2. Distinguish between selected canonical forms of operating systems and computer hardware, architecture and organization.	--
A3. Understand the use and impact of Information and Communication Technology (ICT) by recognizing the concepts of enterprise information management and networking.	--
A4. Understand the software process life cycles, for small, medium and large software projects, and different approaches of software development, software project management, risk and quality management, and human-computer interaction.	a1. Identify and enhance all support line levels involved in maintenance based on industry standards a2. Understanding the similarities and complexities of the software a3. Understand the concepts of Change

	management and other testing aspects
A5. Recognize the social, professional, legal, ethical, and cultural issues involved in the use of computer technology.	a1. Identify and enhance all support line levels involved in maintenance based on industry standards

Intellectual Skills :	
Alignment of CILOs (Course Intended Learning Outcomes) to PILOs (Program Intended Learning Outcomes)	
Intellectual Skills PILOs	Intellectual Skills CILOs
After completing this program, students would be able to:	After completing this course, students would be able to:
B1. Determine and analyze criteria, specifications and requirements appropriate to specific computing problems and formulate strategies to solve them.	b1. Determine the Challenges of software evolution b2. Discuss Release planning, version control , configuration management
B2. Differentiate the various methods of computational thinking, determining their wide relevance and applicability within other domains in everyday life, and being able to employ them in different conditions.	b2. Discuss Release planning, version control , configuration management
B3. Choose the most appropriate methods and tools including deploying appropriate theory, practices, and tools for the specification, design, implementation, and maintenance as well as the evaluation of computer-based systems.	b2. Discuss Release planning, version control , configuration management
B4. Criticize systems in terms of general quality attributes and possible tradeoffs presented within the given problem.	b3. Choose outsourcing focusing on quality

Professional and Practical Skills	
Alignment of CILOs (Course Intended Learning Outcomes) to PILOs (Program Intended Learning Outcomes)	
Professional and Practical Skills PILOs	Professional and Practical Skills CILOs
After completing this program, students would be able to:	After completing this course, students would be able to:
C1. Apply fundamental concepts of computer science, software engineering, human-computer interaction, science and mathematics in the modeling and design of computer systems.	c1 .Apply and demonstrate Agile methods in maintenance, assess Reused based software maintenance
C2. Develop a wide range of software through all stages of their life cycle, namely problem analysis, requirements specification, design, construction; verification and evolution using appropriate methods and tools.	c1 .Apply and demonstrate Agile methods in maintenance, assess Reused based software maintenance c2. Formulate a plan of strategic maintenance to transform and better align software with the business goals of a company c3. Evaluate and demonstrate model-based tool design for analysis of maintenance and evolution
C3. Operate computing equipment and software systems effectively and efficiently.	c2. Formulate a plan of strategic maintenance to transform and better align software with the business goals of a company
C4. Evaluate software process and artifacts in terms of software quality and models.	c3. Evaluate and demonstrate model-based tool design for analysis of maintenance and evolution c4. Estimate the maintenance and evolution by applying metrics
C5. Plan, undertake, and monitor software projects.	c2. Formulate a plan of strategic maintenance to transform and better align software with the business goals of a company c3. Evaluate and demonstrate model-based tool design for analysis of maintenance and

	<p>evolution</p> <p>c4. Estimate the maintenance and evolution by applying metrics</p> <p>c5. Managing the maintenance process</p>
--	--

Transferable (General) Skills :	
Alignment of CILOs (Course Intended Learning Outcomes) to PILOs (Program Intended Learning Outcomes)	
Transferable (General) Skills PILOs	Transferable (General) Skills CILOs
After completing this program, students would be able to:	After completing this course, students would be able to:
D1. Communication skills: Make succinct presentations to a range of audiences about technical problems and their solutions. This may involve face-to-face, written communication or electronic communication.	d1. Develop and present a project outcome that demonstrates understanding of, and engagement with, the key representation of mixed strategy approach and agile methods with focus on extreme programming
D2. Teamwork: Be able to work effectively as a member of a development team.	d2. Communicate effectively well-informed and well-reasoned positions on these issues, verbally (oral analysis) and in writing (project report) by applying critical thinking reasoning skills, technical skills on a goals of the maintenance and evolution of the software
D3. Numeracy skills: Understand and explain the quantitative dimensions of a problem.	d1. Develop and present a project outcome that demonstrates understanding of, and engagement with, the key representation of mixed strategy approach and agile methods with focus on extreme programming
D4. Self-management skills: Manage one's own learning and development, including time management and organizational skills.	d2. Communicate effectively well-informed and well-reasoned positions on these issues, verbally (oral analysis) and in writing (project report) by applying critical thinking reasoning skills, technical skills on a goals of the maintenance and evolution of the software
D5. Professional development skills: Keep abreast of current developments in the discipline to continue one's own professional development.	d2. Communicate effectively well-informed and well-reasoned positions on these issues, verbally (oral analysis) and in writing (project report) by applying critical thinking reasoning skills, technical skills on a goals of the maintenance and

	evolution of the software
--	---------------------------

VI. Alignment of CILOs to Teaching and Assessment Strategies

First: Alignment of Knowledge and Understanding CILOs

Knowledge and Understanding CILOs	Teaching Strategies	Assessment Strategies
a1. Identify and enhance all support line levels involved in maintenance based on industry standards	Direct teaching (lectures, tutorials, problem solving, groups discussions)	Observation, Assignments , quizzes , problem sheets and exams , exams
a2. Understanding the similarities and complexities of the software	Interactive learning (, forums, cooperative learning)	presentation, tech reports, reports Sheets, problem formulation, Case study , Presentation, Viva, Projects, Seminars, Report,
a3. Understand the concepts of Change management and other testing aspects.	Research project Assignments Experimental learning Presentation E-learning, M-learning Self learning Indirect learning (web resources)	

Second: Alignment of Intellectual Skills CILOs

Intellectual Skills CILOs	Teaching Strategies	Assessment Strategies
b1. Determine the Challenges of		Written test ,Quizzes, Report Assignments ,problem sheets and

software evolution	Direct teaching (lectures, tutorials, problem solving)	exams, tech reports, problem formulation
b2. Discuss Release planning, version control , configuration management		
b3. Choose outsourcing focusing on quality	Interactive learning (, forums, cooperative learning) Assignments Experimental learning Presentation E-learning, M-learning Self learning Indirect learning (web resources)	

Third: Alignment of Professional and Practical Skills CILOs

Professional and Practical Skills CILOs	Teaching Strategies	Assessment Strategies
c1 .Apply and demonstrate Agile methods in maintenance, assess Reused based software maintenance	Group Projects Tutorials Assignments	Report Worksheets Report + Presentation Observations
عميد الكلية د. عبدالقادر العبادي	رئيس القسم أ. نبيل المخلافي	المراجع الموصف
	مسؤول البرنامج أ. وديع القباطي	

<p>c2. Formulate a plan of strategic maintenance to transform and better align software with the business goals of a company</p>		
<p>c3. Evaluate and demonstrate model-based tool design for analysis of maintenance and evolution</p>		

c4. Estimate the maintenance and evolution by applying metrics		
c5. Managing the maintenance process		

Fourth: Alignment of Transferable (General) Skills CILOs		
Transferable (General) Skills CILOs	Teaching Strategies	Assessment Strategies
d1. Develop and present a project outcome that demonstrates understanding of, and engagement with, the key representation of mixed strategy approach and agile methods with focus on extreme programming	Grouping Tips Notes software projects	Presentation Observation Report
d2. Communicate effectively well-informed and well-reasoned positions on these issues, verbally (oral analysis) and in writing (project report) by applying critical thinking reasoning skills, technical skills on a goals of the maintenance and evolution of the software		

I. Course topics and sub-topics (theoretical and practical) with contact hours and alignment to CILOs					
Topics/Units of Course Contents					
First: Theoretical Aspects					
No.	Course Topics/Units	Sub-topics	No. of Weeks	Contact Hours	CILOs
1	Introduction	Background The Problem of Software Change Change management Version control systems Release planning Regression testing Software Evolution and Its Challenges	1	3	a3,a4,b1,b2
2	Conventional methods of software maintenance and evolution	A Methodology for Software Maintenance -Maintenance Planning - EvolutionandMaintenaneTesting. -Configuration Management	2	6	a1,a3,c1,c2,c3

الموصف المراجع مسؤول البرنامج رئيس القسم عميد الكلية
أ.وديع القباطي د.عبدالقادر العبادي أ.نبيل المخلافي د.عبدالقادر العبادي

		<p>-Problem Management with corrective maintenance -Maintenance supporting tools</p> <p>Static Program Analysis Methods - Reverse Engineering Methods</p>			
3	Model-Based Design for Ease of Maintenance	<p>Evolution of Versions and Configuration Management</p> <p>Impact of agile methods in maintenance -extreme programming on maintenance</p> <p>Limits of Conventional Techniques</p>	1	3	b2,c1
4	Re-used based software – software maintenance and evolution	<p>The Mixed-Strategy Approach</p> <p>Concepts of Reuse-Based Evolution Change-Design versus Design for Change</p> <p>Concepts of the Mixed-Strategy Approach</p> <p>A Preview of the Mixed-Strategy Approach: An Example</p> <p>The Role of Genericity in the Mixed-Strategy Approach</p>	1	3	c1
5	Overview • Step-By-Step Introduction to XVCL	<p>Salient Features of XVCL</p> <p>Flexible Composition of X-Frames</p> <p>Defining Compositions with XVCL</p> <p>Specifying Small-Granularity Changes</p> <p>Changes at Various Abstraction Levels</p> <p>Defining Generic Structures and Generators</p> <p>Capturing Change Traces and Similarity Patterns in</p> <p>Evolutionary Changes</p> <p>Handling Implementation Dependencies</p>	1	3	c1

		among Features			
MIDTERM					
6	Software Similarities:	The Problem of Software Similarities and Cloning Software Similarities and Reuse Software Similarities and Generic Design Software Similarities, Genericity, and Software Complexity Similarity Patterns and Clones in STL Similarity Patterns in Application Programs General Implications Identifying Similarity Patterns and Automated Detection of Clones	2	6	a2
8	Evolution with the Mixed-Strategy Approach	Introduction An Overview of the FRS X-Framework Development of a New System Release Specifying Changes for FRS	1	3	c1
9	Scaling Up the Mixed-Strategy Approach	Introduction Modeling Features Arising during Evolution A Mixed-Strategy Evolution Support Environment (MESE) An FRS Architecture with Connectors Using the Customization Decision Tree (CDT) Using MESE to Customize and Evolve the FRS Assembly Pass	1	3	c1
10	Evaluation of the Mixed-Strategy	The Mixed-Strategy Process Trade-Offs Involved in Applying the Mixed-Strategy Approach.	1	3	c1

	Approach				
11	Maintenance Measurements	Maintenance Metrics Maintenance Cost Estimation	1	3	c4
12	Management and Organizational Issues Teams	Organization Aspect of Maintenance Maintenance Activities and Role Outsourcing IT Maintenance Managing the Maintenance Function Maintenance	1	3	b3,c5
15	FINAL EXAM				
Total number of weeks and hours			14	42	

I. Tasks and Assignments :				
No.	Task/Assignment	CILOs	Week due	Mark
1	Assignment 1	c1,c3,c4	4	10
2	Assignment 2	a1,a2,a3,b2	11	10
3	Project	c2,d1,d2	12	20

I. Learning Assessment:					
No.	Assessment Tasks	Week due	Mark	Proportion of Final Assessment	Aligned CILOs
1	Quizzes	3 , 6, 10	10+10+10	10%	--
2	Problem based Assignments	4 & 11	10+10	10%	
3	Midterm exam	7	25	10%	--
4	Project	8,12	10+10	20%	
5	Final exam	15	100	50%	--
Total			100		

I. Learning Resources :				
(Author, (Year), Book Title, Edition, Publisher, Country of publishing)				
Textbooks-not more than 2				
<p>Effective Software Maintenance and Evolution: A Reuse-Based Approach, Berlin ; New York: Springer Stanislaw Jarzabek ,auerback publications,taylor and francis group,newyork</p> <p>Grubb and Takang, Software Maintenance: Concepts and Practices, 2nd edition, World Scientific, 2003.</p>				
Essential References-not less than 4				
<p>Hallsteinsen, S. and Paci, M,1997, Experiences in software evolution and reuse : twelve real world projects</p> <p>Software engineering handbook Boca Raton: Auerbach. Polo, M,</p> <p>Advances in software maintenance management : technologies and solutions Hershey, PA: Idea Group Pub. West, R., 1993, Improving the maintainability of software London: HMSO</p>				
Electronic Materials and Web Sites				
<ul style="list-style-type: none"> • Selection of papers from technical literature <p>IEEE Standard for Software Maintenance ISO/IEC FDIS 14764:2005(E),</p> <ul style="list-style-type: none"> • 				

عميد الكلية
د. عبدالقادر العبادي

رئيس القسم
أ. نبيل المخلافي

مسؤول البرنامج
أ. وديع القباطي

المراجع

الموصف

Template of Course Syllabus

Faculty : Computing and Information technology

Department : Computer Science

Program : B.Sc(software Engineering)

I. General information about the course instructor :							
Name		Office Hours(3 Hours Weekly)					
Location & phone number		Sat	Sun	Mon	Tue	Wed	Thu
Email							

II. General information about the course :						
3.	Course Title:	Software maintenance and evolution				
4.	Course Code and Number :	SE421				
5.	Credit Hours: 3	Lecture	Seminar/Tutorial	Practical	Training	Total
		3	0	0		3
6.	Study Level and Semester:	Level 4 / Semester 8				
7.	Pre-requisites (if any):	Software engineering (CS325)				
8.	Co-requisites (if any) :	Differential equations(CS221) and Data structures and algorithms (CIT07),Calculus (CIT02)				
9.	Program in which the course is offered	B.Sc(Software engineering)				
10.	Teaching Language:	English				
11.	Study System :	Semester Based				
12.	Prepared by :	Course Facilitator (Bridget Shakesy)				
13.	Approval date :					
14.	Approved by:	Department				

III. Course Description:

Organizations across all industries have huge investments in their software systems. These systems must be continually adapted to the changing needs of the organizations. Software maintenance and evolution refers to the process of modifying existing software systems to maintain their usefulness. Some studies have estimated that up to 75% of all technical software professionals are involved in some form of software maintenance activity. This course builds on basic software engineering concepts by expanding the discussion of the issues involved in maintaining and evolving software systems.

IV. Course Aims

Motivate why it is important to build maintainability in a system

Plan for the whole software lifecycle with emphasis on risk analysis and maintainability in the initial phases of software development

Identify problems within the industry and research

Choose appropriate problem management method. The choice should be based on software criticality and problem urgency.

Explain differences among all the types of maintenance categories

Make decisions and manage legacy systems

Identify challenges associated with software evolution.

Evaluate agile methods-extreme programming from the lifecycle perspective

V.Course topics and sub-topics (theoretical and practical) with contact hours and alignment to CILOs

Topics/Units of Course Contents				
No.	Course Topics/Units	Sub-topics	No. of Weeks	Contact Hours
1	Introduction	Background The Problem of Software Change Change management Version control systems Release planning Regression testing Software Evolution and Its Challenges	1	3
2	Conventional methods of software maintenance and evolution	A Methodology for Software Maintenance -Maintenance Planning -EvolutionandMaintenaneTesting. -Configuration Management -Problem Management with corrective maintenance -Maintenance supporting tools Static Program Analysis Methods - Reverse Engineering Methods	2	6
3	Model-Based Design for Ease of Maintenance	Evolution of Versions and Configuration Management Impact of agile methods in maintenance - extreme programming on maintenance Limits of Conventional Techniques	1	3

4	Re-used based software – software maintenance and evolution	<p>The Mixed-Strategy Approach</p> <p>Concepts of Reuse-Based Evolution</p> <p>Change-Design versus Design for Change</p> <p>Concepts of the Mixed-Strategy Approach</p> <p>A Preview of the Mixed-Strategy Approach: An Example</p> <p>The Role of Genericity in the Mixed-Strategy Approach</p>	1	3
5	Overview • Step-By-Step Introduction to XVCL	<p>Salient Features of XVCL</p> <p>Flexible Composition of X-Frames</p> <p>Defining Compositions with XVCL</p> <p>Specifying Small-Granularity Changes</p> <p>Changes at Various Abstraction Levels</p> <p>Defining Generic Structures and Generators</p> <p>Capturing Change Traces and Similarity Patterns in</p> <p>Evolutionary Changes</p> <p>Handling Implementation Dependencies among Features</p>	1	3
6	Software Similarities:	<p>The Problem of Software Similarities and Cloning</p> <p>Software Similarities and Reuse</p> <p>Software Similarities and Generic Design</p> <p>Software Similarities, Genericity, and Software Complexity</p> <p>Similarity Patterns and Clones in STL</p> <p>Similarity Patterns in Application Programs</p> <p>General Implications</p> <p>Identifying Similarity Patterns and Automated Detection of Clones</p>	2	6
8	Evolution with the Mixed-Strategy	<p>Introduction</p> <p>An Overview of the FRS X-Framework</p>	1	3

	Approach	Development of a New System Release Specifying Changes for FRS		
9	Scaling Up the Mixed-Strategy Approach	Introduction Modeling Features Arising during Evolution A Mixed-Strategy Evolution Support Environment (MESE) An FRS Architecture with Connectors Using the Customization Decision Tree (CDT) Using MESE to Customize and Evolve the FRS Assembly Pass	1	3
10	Evaluation of the Mixed-Strategy Approach	The Mixed-Strategy Process Trade-Offs Involved in Applying the Mixed-Strategy Approach.	1	3
11	Maintenance Measurements	Maintenance Metrics Maintenance Cost Estimation	1	3
12	Management and Organizational Issues Teams	Organization Aspect of Maintenance Maintenance Activities and Role Outsourcing IT Maintenance Managing the Maintenance Function Maintenance	1	3
15				
Total number of weeks and hours			14	

V.Tasks and Assignments :			
No.	Task/Assignment	Week due	Mark
1	Assignment 1	4	10
2	Assignment 2	11	10
3	Project	12	20

VI. Learning Assessment:

No.	Assessment Tasks	Week due	Mark	Proportion of Final Assessment	Aligned CILOs
1	Quizzes	3 , 6, 10	10+10+10	10%	--
2	Problem based Assignments	4 & 11	10+10	10%	
3	Midterm exam	7	25	10%	--
4	Project	8,12	10+10	20%	
5	Final exam	15	100	50%	--
Total			100		

VII. Learning Resources :

(Author, (Year), Book Title, Edition, Publisher, Country of publishing)

Textbooks-not more than 2

Effective Software Maintenance and Evolution: A Reuse-Based Approach, Berlin ; New York: Springer Stanislaw Jarzabek ,auerback publications,taylor and francis group,newyork

Grubb and Takang, Software Maintenance: Concepts and Practices, 2nd edition, World Scientific, 2003.

Essential References-not less than 4

Hallsteinsen, S. and Paci, M,1997,
Experiences in software evolution and reuse : twelve real world projects

Software engineering handbook
Boca Raton: Auerbach.
Polo, M,

Advances in software maintenance management : technologies and solutions
Hershey, PA: Idea Group Pub.
West, R., 1993,
Improving the maintainability of software
London: HMSO

Electronic Materials and Web Sites

- Selection of papers from technical literature

IEEE Standard for Software Maintenance
ISO/IEC FDIS 14764:2005(E),

-