University of Macau
Faculty of Science and Technology
Department of Computer and Information Science
SFTW497 – Software Project Management
Syllabus
1st Semester 2012/2013
Part A – Course Outline

Compulsory course in Computer Science

Course description:

(2-1) 2 credits. This course introduces project management as it relates to the software life cycle. Different software life cycle models, and the project management activities in each phase of the life cycle, are studied. Project planning activities are introduced, including effort estimation and the use of software metrics; risk analysis; resource allocation; and project scheduling. The course concludes with a study of project monitoring and control, project contracts, and team organization.

Course type:

Theoretical with substantial laboratory/practice content

Prerequisites:

none

Textbook(s) and other required material:

- Bob Hughes & Mike Cotterell, *Software Project Management*, 5th ed., McGraw-Hill, 2010. (Required)

References:

- John J. Rakos, *Software Project Management for Small to Medium Sized Projects*, Prentice Hall, 1990.
- William H. Roetzheim, *Structured Computer Project Management*, Prentice Hall, 1988.
- Roger S. Pressman, *Software Engineering: A Practitioner's Approach*, 7th ed., McGraw-Hill, 2009.
- Tom DeMarco, *Controlling Software Projects: Management, Measurement and Estimates*, Prentice Hall, 1986.

Major prerequisites by topic:

none

Course objectives:

1. Introduce students to fundamental concepts of software projects and their management. [a]
2. Enable students to understand and differentiate software development life cycle models, and select a suitable model for a given project. [a, c]
3. Introduce students to techniques of project planning, monitoring and control. [a, f]
4. Understand concepts of project team composition. [d]
5. Learn to prepare, maintain and communicate a project plan. [c, g]
6. Learn to apply project management tools to software project management. [a, k, l]

Topics covered:

1. **Introduction to software project management (4 hours):** Study the importance of software project management, basic definitions and project characteristics, project management activities in different project life cycle stages, project objectives and measures, management control.
2. **Step Wise project planning method (3 hours):** Give an overview of the Step Wise project planning method, project scope and objectives, project infrastructure and characteristics, products and activities, effort estimation, risk analysis, resource allocation, publicising the plan.
3. **Programme management and project evaluation (4 hours):** Study programme vs project management, benefits management, cost-benefit analysis (net profit, return on investment, payback period, net present value, internal rate of return), risk profile analysis.

4. **Selection of an appropriate project approach (3 hours):** Introduce overall project risks, process models: waterfall, V-process mode, spiral model, prototyping, incremental process, agile methods, extreme programming.
5. **Software effort estimation (3 hours):** Introduce bottom-up estimation, top-down estimation, function point analysis, function points Mk II, COCOMO, estimation by analogy.
6. **Activity planning (3 hours):** Study product breakdown, product flow, work breakdown, PERT charts, CPM charts, lead and lag times, task dependencies, calculating schedule dates, task float, critical path.
7. **Risk management (3 hours):** Introduce definitions of risk, risk cause and effect, risk identification, causal mapping, risk prioritization, risk planning, risk reduction, calculating z value and likelihood of meeting targets, critical chain approach.
8. **Resource allocation (3 hours):** Introduce definition of resources, resource histogram, resource levelling, Burman's priority list, allocating individuals, cost schedules.
9. **Monitoring and control (3 hours):** Study project control cycle, assessing progress, representing progress (red/amber/green reporting, Gantt charts, slip charts, ball charts, timeline), cost monitoring, earned value analysis, change control.
10. **Project contracts (3 hours):** Study software acquisition process, request for proposals, types of contract (fixed price, time and materials, fixed price per unit), tendering process, contract terms.
11. **Managing people and organizing teams (3 hours):** Introduce organizational behaviour, motivation, team formation and team balance, team roles, task type, leadership type, virtual projects, communication methods by project phase, cultural influence.

Class/laboratory schedule:

Timetabled work in hours per week			No of teaching weeks	Total hours	Total credits	No/Duration of exam papers
Lecture	**Tutorial**	**Practice**				
2	1 (10/14)*	1 (4/14) *	14	42	2	1 / 2 hours

* Note: 10 weeks have a 1-hour tutorial, 4 weeks have a 1-hour practice

Student study effort required:

Class contact:	
Lecture	26 hours
Tutorial	9 hours
Practice	5 hours
Mid-term exam	2 hours
Other study effort:	
Self-study	14 hours
Homework assignment	4 hours
Total student study effort	60 hours

Student assessment:

Final assessment will be determined on the basis of:

Homework assignments	20%
Mid-term exam	30%
Final exam	50%

Course assessment:

The assessment of course objectives will be determined on the basis of:

1. Homework assignments and exams
2. Course evaluation

Course outline:

Weeks	Topic	Course work

Weeks	Topic	Course work
1-2	**Introduction to software project management** Importance of software project management, basic definitions and project characteristics, project management activities in different project life cycle stages, project objectives and measures, management control.	
2-3	**Step Wise project planning method** Overview of the Step Wise project planning method, project scope and objectives, project infrastructure and characteristics, products and activities, effort estimation, risk analysis, resource allocation, publicising the plan.	
3-4	**Programme management and project evaluation** Programme vs project management, benefits management, cost-benefit analysis (net profit, return on investment, payback period, net present value, internal rate of return), risk profile analysis.	
5	**Selection of an appropriate project approach** Overall project risks, process models: waterfall, V-process mode, spiral model, prototyping, incremental process, agile methods, extreme programming.	
6	**Software effort estimation** Bottom-up estimation, top-down estimation, function point analysis, function points Mk II, COCOMO, estimation by analogy.	
7	**Activity planning** Product breakdown, product flow, work breakdown, PERT charts, CPM charts, lead and lag times, task dependencies, calculating schedule dates, task float, critical path.	
8	(*exam week*)	Mid-term exam
9	**Risk management** Definitions of risk, risk cause and effect, risk identification, causal mapping, risk prioritization, risk planning, risk reduction, calculating z value and likelihood of meeting targets, critical chain approach.	
10	**Resource allocation** Definition of resources, resource histogram, resource levelling, Burman's priority list, allocating individuals, cost schedules.	
11	**Monitoring and control** Project control cycle, assessing progress, representing progress (red/amber/green reporting, Gantt charts, slip charts, ball charts, timeline), cost monitoring, earned value analysis, change control.	Assignment 1
12	**Project contracts** Software acquisition process, request for proposals, types of contract (fixed price, time and materials, fixed price per unit), tendering process, contract terms.	
13	**Managing people and organizing teams** Organizational behaviour, motivation, team formation and team balance, team roles, task type, leadership type, virtual projects, communication methods by project phase, cultural influence.	Assignment 2

Contribution of course to meet the professional component:

This course prepares students to work professionally in the area of software development.

Relationship to CS program objectives and outcomes:

This course primarily contributes to the Computer Science program outcomes that develop these student abilities:
(a) an ability to apply knowledge of computing, mathematics, science, and engineering.
(c) an ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability.
(k) an ability to use the techniques, skills, and modern computer tools necessary for engineering practice.
(l) an ability to use the computer/IT tools relevant to the discipline along with an understanding of their processes and limitations.

The course secondarily contributes to the Computer Science program outcomes that develop these student abilities:
(d) an ability to function effectively on multi-disciplinary teams.
(f) an understanding of professional, ethical, legal, security and social issues and responsibilities.

(g) an ability to communicate effectively.

Relationship to CS program criteria:

Criterion	DS	PF	AL	AR	OS	NC	PL	HC	GV	IS	IM	SP	SE	CN
Scale: 1 (highest) to 4 (lowest)												4	1	

Discrete Structures (DS), Programming Fundamentals (PF), Algorithms and Complexity (AL), Architecture and Organization (AR), Operating Systems (OS), Net-Centric Computing (NC), Programming Languages (PL), Human-Computer Interaction (HC), Graphics and Visual Computing (GV), Intelligent Systems (IS), Information Management (IM), Social and Professional Issues (SP), Software Engineering (SE), Computational Science (CN).

Course content distribution:

Percentage content for			
Mathematics	Science and engineering subjects	Complementary electives	**Total**
0%	50%	50%	100%

Coordinator:

Prof. Xiaoshan Li

Persons who prepared this description:

Dr. Robert P. Biuk-Aghai
Dr. Fai Wong

Part B General Course Information and Policies

1st semester 2012/2013

Instructor: Dr. Robert P. Biuk-Aghai — Office: N325
Office Hour: Wed 2:30 – 6:30 pm, Thu 2:30 am – 4:30 pm, or by appointment — Phone: 8397 4375
Email: robertb@umac.mo

Time/Venue: Thu 4:30 – 6:30 pm, U107 (lectures)
Fri 4:30 – 5:30 pm, JM22/T104 (tutorial/laboratory)

Grading Distribution:

Percentage Grade	Final Grade	Percentage Grade	Final Grade	Percentage Grade	Final Grade
100 – 93	A	77 – 73	B-	57 – 53	D+
92 – 88	A-	72 – 68	C+	52 – 50	D
87 – 83	B+	67 – 63	C	below 50	F
82 – 78	B	62 – 58	C-		

Comment:
The objectives of the lectures are to explain and to supplement the text material. Students are responsible for the assigned material whether or not it is covered in the lectures. Students who wish to succeed in this course should read the lecture notes prior to the lecture and should do all practice exercises. You are encouraged to look at other sources (other texts, etc.) to complement the lectures and primary text.

Homework Policy:
The completion and correction of the practice exercises is a powerful learning experience; therefore:

- There will be 2 graded practice exercises.
- Practice exercises are due one week after assignment unless otherwise noted, and late submissions will lose points (1% point off per hour late).
- Possible revision of grades may be discussed with the grader within one week from the return of the marked practice exercise.
- The course grade will be based on the average of the practice exercises.

Mid-term Exam:
There will be one mid-term exam held at about the middle of the semester.

Note

- The lecture session is an important part of this course and attendance is compulsory. At most 20% absence without leave is allowed.
- Check UMMoodle (ummoodle.umac.mo) for announcements, practice exercises and lecture notes. Report any mistakes on your grades within one week after posting.
- No make-up exam is given except for CLEAR medical proof.
- No exam is given if you are 30 minutes late in the midterm exam, or 45 minutes late in the final exam. Even if you are late in the exam, you must turn in at the due time.
- Cheating is strictly prohibited by the university and will be severely punished.

Rubric for Program Outcomes (a) to (l)

Rubric for (a)	5 (Excellent)	3 (Average)	1 (Poor)
Understand the theoretic background	Students understand theoretic background and the limitations of the respective applications.	Students have some confusion on some background or do not understand theoretic background completely.	Students do not understand the background or do not study at all.
Use a correct model and formulation correctly	Students choose a model correctly and properly apply correct techniques.	Students choose a wrong model sometime, use a wrong formula, or a different technique.	Students use a wrong model and wrong formula, or do not know how to model.
Compute the problem correctly	Students use correct techniques, analyze the problems, and compute them correctly.	Students sometime solve problem mistakenly using wrong techniques.	Students do not know how to solve problems or use wrong techniques completely.

Rubric for (c)	5 (Excellent)	3 (Average)	1 (Poor)
Design capability and design constraints	Student understands very clearly what needs to be designed and the realistic design constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability.	Student understands what needs to be designed and the design constraints, but may not fully understand the limitations of the design constraints.	Student does not understand what needs to be designed and the design constraints.
Process to meet desired needs	Student understands very clearly the process of the design.	Student understands what the needs of the process design, but may not fully understand the limitations of the design constraints.	Student does not understand the process.

Rubric for (d)	5 (Excellent)	3 (Average)	1 (Poor)
Ability to work in teams	Performance on teams is excellent with clear evidence of equal distribution of tasks and effort as well as frequent meetings of the team members.	Performance on teams is acceptable with one or more members carrying a larger amount of the effort as well as infrequent meetings of the members or one or more members being absent from several meetings.	Performance on teams is poor to unacceptable with one or two members clearly carrying the majority of the effort as well as inadequate team meeting or one or more members missing the majority of the meetings.
Multi-disciplinary teams	Team consists of members from two or more different engineering/science/business fields (this could contain some members not actually enrolled in the course but interacting as part of a competition, collaboration, etc.)	Team consists of members from two or more concentrations within the Department of Computer and Information Science.	Team consists of members from the same concentration within the Department of Computer and Information Science.

Rubric for (f)	5 (Excellent)	3 (Average)	1 (Poor)
Design	Understand how to critique and analyze design tradeoffs and constraints with respect to safety, liability, and integrity of	Have knowledge of safety, liability, and integrity of data, and context of use but cannot analyze thoroughly.	No awareness of importance of safety, liability, and integrity of data, and context of use.

	data, and context of use.		
Professional engineering practice	Understand how to critique and analyze tradeoffs and constraints with respect to research issues of credit and authorship, integrity of data, and informed consent.	Have knowledge of credit and authorship, integrity of data, and informed consent but cannot completely identify ownership in practical.	No awareness of credit and authorship, integrity of data, and informed consent.
Group relations	Understand how to critique and analyze tradeoffs and constraints with respect to conflict of interest, bribery, professional dissent, authorship, and discrimination.	Have partial knowledge of conflict of interest, bribery, professional dissent, authorship, discrimination but cannot apply it in practice correctly.	No awareness of conflict of interest, bribery, professional dissent, authorship, and discrimination.

Rubric for (g)	**5 (Excellent)**	**3 (Average)**	**1 (Poor)**
Professional impact	Student's/Team's/Group's document(s)/presentation(s) is/are considered to be of professional quality.	Student's/Team's/Group's document(s)/presentation(s) is/are considered acceptable for college level work.	Student's/Team's/Group's document(s)/presentation(s) is/are considered unacceptable for college level work.
Written component	Document is nearly error free with sophisticated use of vocabulary, formatted properly, with well-developed concise sentences and paragraphs.	Document contains some errors with a somewhat colloquial vocabulary, minor formatting issues, with some organizational issues that do not interfere with communication.	Document contains many errors, very colloquial vocabulary, with severe organizational issues that interfere with communication. Document would be considered unacceptable.
Oral component	Presentation is consistent, uniform, clear, direct, complete and captivating with very clear fonts and graphics with an excellent layout that clearly presents the technical content.	Presentation is somewhat inconsistent between speakers, occasionally difficult to hear, with an acceptable layout containing acceptable fonts and graphics that adequately presents the technical content.	Presentation is very inconsistent between speakers, difficult to hear with a poor layout containing illegible fonts and graphics that poorly presents the technical content. Would be considered unacceptable.

Rubric for (k)	**5 (Excellent)**	**3 (Average)**	**1 (Poor)**
Use modern principles, skills, and tools in engineering practice	Student applies the principles, skills and tools to correctly model and analyze engineering problems, and understands the limitations.	Student applies the principles, skills and tools to analyze and implement engineering problems.	Student does not apply principles and tools correctly and/or does not correctly interpret the results.

Rubric for (l)	**5 (Excellent)**	**3 (Average)**	**1 (Poor)**
Use modern computer/IT tools relevant to the discipline	Student uses computer/IT tools relevant to the engineering discipline, and understands their limitations.	Student uses computer /IT tools relevant to the engineering discipline.	Student does not use computer/IT tools relevantly, and does not understand their limitations.