SOEN 384
Management, Measurement and Quality Control

http://users.encs.concordia.ca/~s384_2/

Instructor: Dr. Olga Ormandjieva

Lecture 1:

Course Overview. Introduction to Software Management

1. *SWEBOK chapter 7*
2. New: **Managing the Development of Software-Intensive Systems By James McDonald (**John Wiley & Sons, Inc. 2010)

Agenda for today

- Course roadmap
- Basic notions of software project management.
- Course goal and objectives
- Our Software Management Model
- Course assumptions
- SWEBOK

Next?

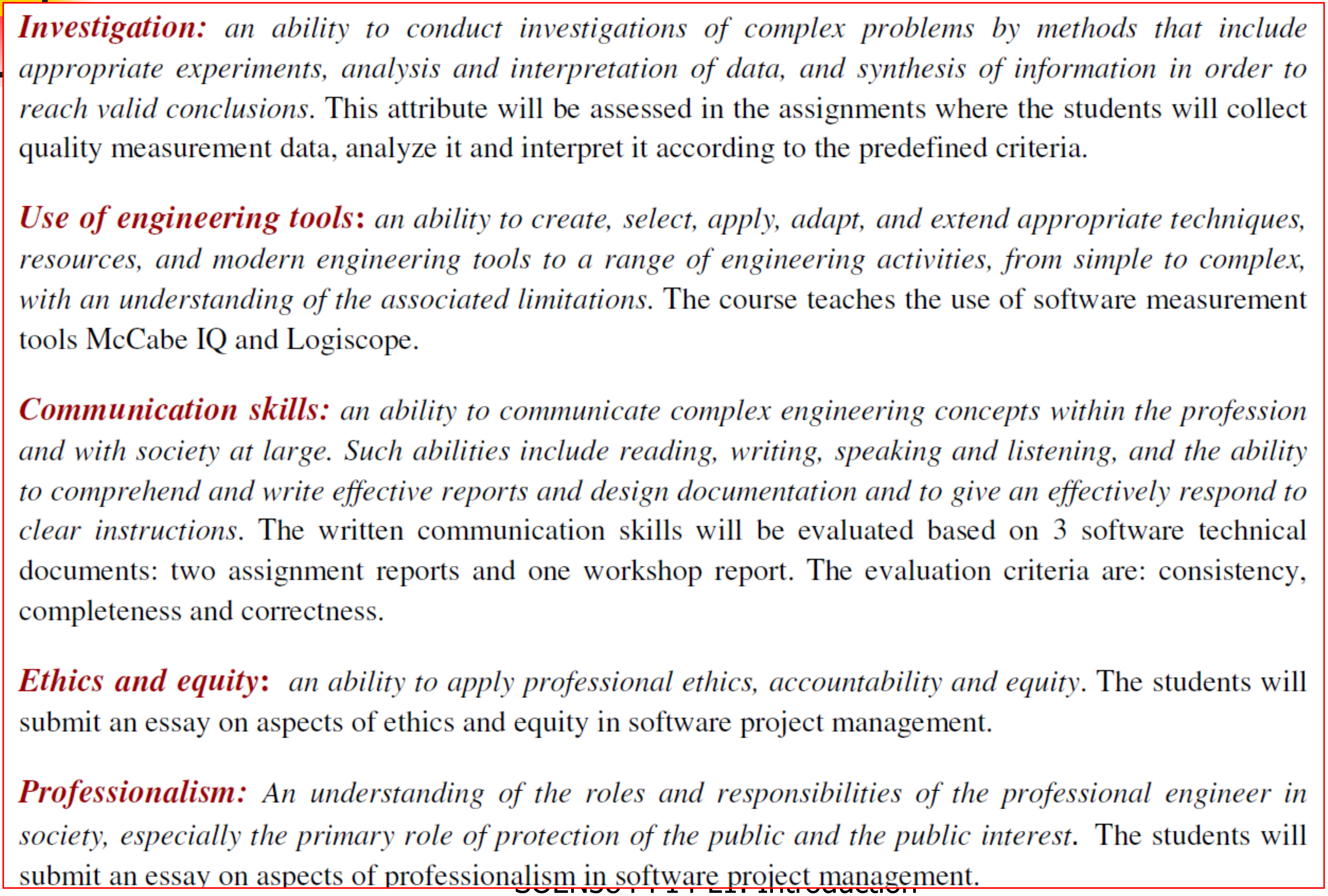
Soen384 – Fall 2014

Description:

Calendar: "*Organization of large software development. Roles of team members, leaders, managers, stakeholders, and users. Tools for monitoring and controlling a schedule. Financial, organizational, human, and computational resources allocation and control. Project and quality reviews, inspections, and walkthroughs. Risk management. Communication and collaboration. Cause and effects of project failure. Project management via the Internet. Quality assurance and control.*"

Course webpage: http://users.encs.concordia.ca/~s384_2/

Canadian Engineering Accreditation Board's graduate attributes to be evaluated in this course



Investigation: *an ability to conduct investigations of complex problems by methods that include appropriate experiments, analysis and interpretation of data, and synthesis of information in order to reach valid conclusions*. This attribute will be assessed in the assignments where the students will collect quality measurement data, analyze it and interpret it according to the predefined criteria.

Use of engineering tools**:** *an ability to create, select, apply, adapt, and extend appropriate techniques, resources, and modern engineering tools to a range of engineering activities, from simple to complex, with an understanding of the associated limitations*. The course teaches the use of software measurement tools McCabe IQ and Logiscope.

Communication skills: *an ability to communicate complex engineering concepts within the profession and with society at large. Such abilities include reading, writing, speaking and listening, and the ability to comprehend and write effective reports and design documentation and to give an effectively respond to clear instructions*. The written communication skills will be evaluated based on 3 software technical documents: two assignment reports and one workshop report. The evaluation criteria are: consistency, completeness and correctness.

Ethics and equity**:** *an ability to apply professional ethics, accountability and equity*. The students will submit an essay on aspects of ethics and equity in software project management.

Professionalism: *An understanding of the roles and responsibilities of the professional engineer in society, especially the primary role of protection of the public and the public interest*. The students will submit an essay on aspects of professionalism in software project management.

Grading:

- Assignment 1 on software measurement tools (team work, 4-5 students in a team, **10%**)
- Assignment 2 on quality control (team work, 4-5 students in a team, **10%**)
- Midterm (**20%**)
- Final Exam (**35%**)
- Workshop on agile estimation and planning & Retrospective sessions (team work, 8-10 students in a team, **10%**) . Teams will be formed at the beginning of the workshop.
- Participation in class (5-10 min small "surprise" quizzes at the beginning of some lectures, or exercises. Total weight: **10%**)
- Two essays on Ethics and Equity & Professionalism: (team work, 4-5 students in a team, 2 x **2.5%** each)

All deliverables for this course have to be submitted through the **Electronic Assignment Submission** system**:** https://fis.encs.concordia.ca/eas/

Tutorials

- First tutorial is on Friday, September 12
  - 9:15 – 10:05, H929

communication

- **Course mailing list:**

  soen384-f14 at cse.concordia.ca

- **Email all questions related to teaching and organization of the course to:**

  priv-soen384-f14 at cse.concordia.ca

More on the course

- Password-protected material
- How to study for this course
  - Slides are giving the structure of the material
  - Read the corresponding reference material
    - Written on the first page of the lecture notes.
  - Do exercises, assignments
- Office hours
  - Wednesdays 2pm-3pm o by appointment

Agenda for today

- Course roadmap
- Basic notions of software project management.
- Course goal and objectives
- Our Software Management Model
- Course assumptions
- SWEBOK

Next?

What is a Software-based Development Project?

- A set of tasks performed by people with assistance from a variety of tools, computers, compilers, software development environments, test equipment, etc.
- a set of processes that connect those tasks in order in which the tasks must be accomplished
- Constraints: scope, schedule, budget, quality
- Specific characteristics:
  - Has definite starting and end points
  - Creates a unique product or service
  - It is extremely difficult to look at a piece of software and to know the status of the work involved in developing it.

What software development project managers typically do?

- Understand the business needs
- Understand the constraints (budget, schedule, scope, volatility of requirements ...)
- Negotiate a commitment (scope, schedule, budget, resources)
- Lead in planning
  - Identify every task, define the tasks, allocate resources to tasks, schedule tasks, and assign responsibility for completion of tasks for the overall project.
  - Clearly define and manage the definition of interfaces (technical and interpersonal communication)
- Execute the plan by monitoring the status and indentifying discrepancies between the plan and the status, and addressing issues
- Communicate goals, plans, project status, and changes to all of the stakeholders.

Project Management Process

(Managing the Development of Software-Intensive Systems)

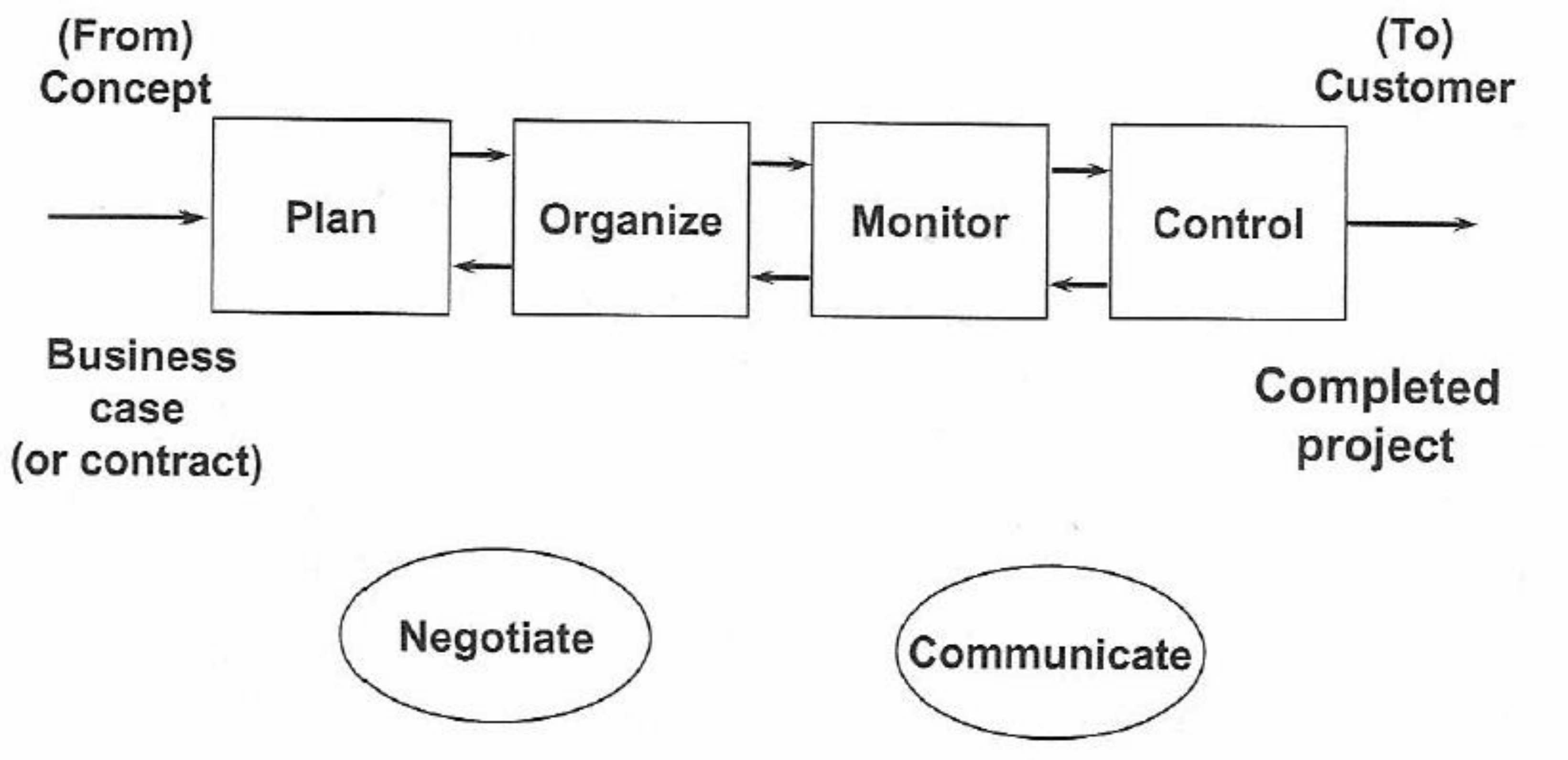


Figure 1.1 Project management processes.

Project management in reality …

Set of management processes and techniques:

- Planning
- Organizing
- Monitoring
- Controlling

To be integrated with:

- Underlying business processes
- Underlying technical processes
  - Specification, architecture, design, coding, testing, deployment, maintenance …
- External contract organizations

How to integrate them?

Agenda for today

- Course roadmap
- Basic notions of software project management.
- Course goal and objectives
- Our Software Management Model
- Course assumptions
- SWEBOK

Next?

Software Project Management Goal (and soen384 goal)

To integrate software project management processes with the underlying business and technical processes in order to develop **high-quality** products in **predictable** ways

Objectives of the course: to develop …

1. A broad understanding of how project management can be applied to software development projects
2. An understanding of the estimation methods
3. The ability to properly apply project management methods to the requirements, architecture, design, testing, and delivery of software products
4. An understanding of common organizational structures, team building, and conflict resolution
5. The ability to monitor and to report on the status of software development work
6. The ability to use audits, reviews, an assessments effectively
7. The ability to minimize risk
8. The ability to tailor the project management processes to appropriately match the project and its environment
9. The ability to review successes and failures and to lean from our experiences.

Agenda for today

- Course roadmap
- Basic notions of software project management.
- Course goal and objectives
- Our Software Management Model
- Course assumptions
- SWEBOK
- Next?

A broad understanding of how project management can be applied to software development projects: Our Model (Managing the Development of Software-Intensive Systems)

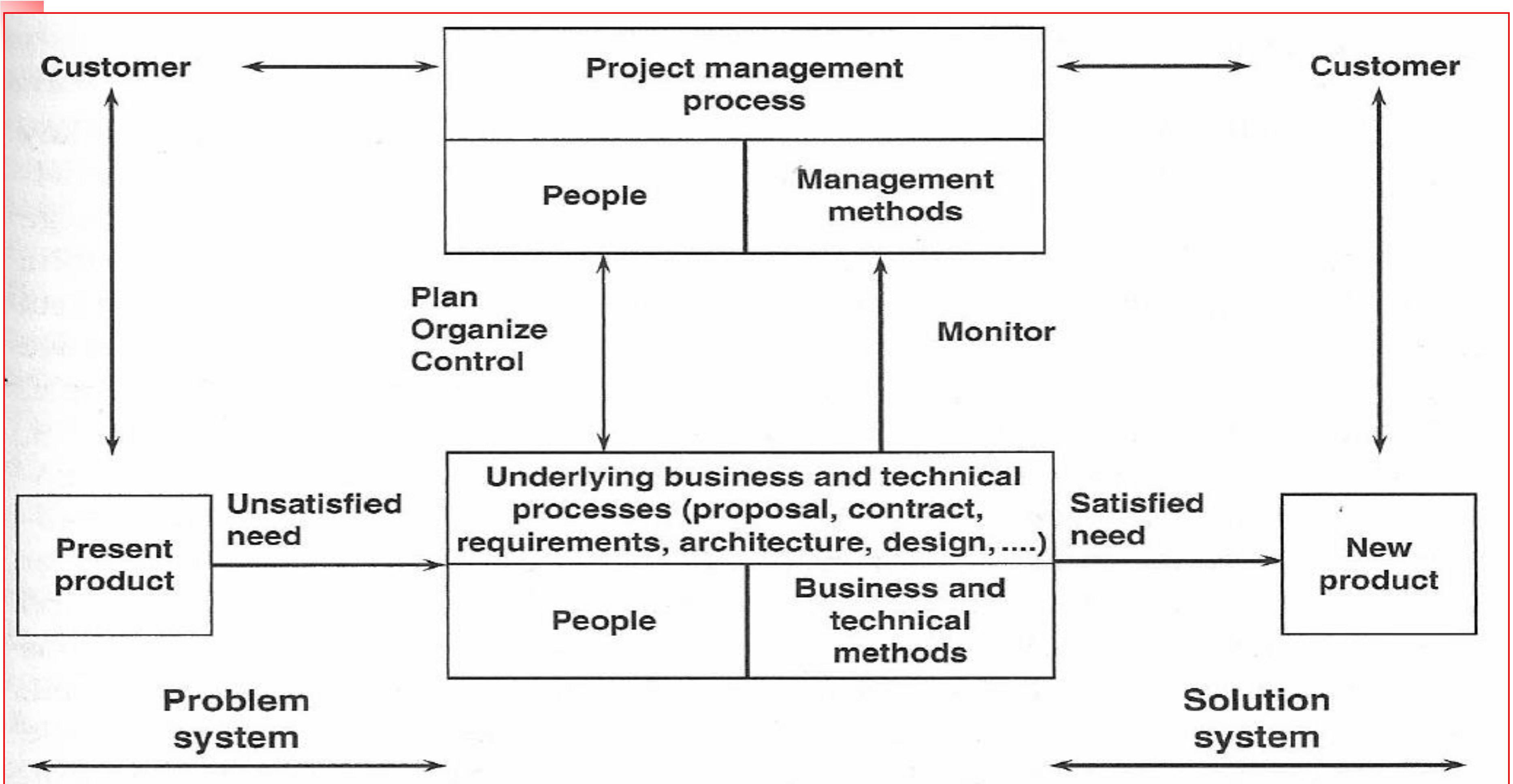

Customer
Project management process
Customer
People
Management methods
Plan
Organize
Control
Monitor
Present product
Unsatisfied need
Underlying business and technical processes (proposal, contract, requirements, architecture, design, ...)
Satisfied need
New product
People
Business and technical methods
Problem system
Solution system

Figure 1.2 Model assumed throughout this book (from Crawford and Fallah 1985, reproduced with permission of IEEE © 1985 IEEE).

Project Management v.s. **Software** Project Management

- Project Management is a set of rigid methods
- Why **Software** Project Management is different?
  - Different development processes
  - Different types of software
    - Web-based applications
    - Large data management systems
    - Real time systems
    - Reactive embedded systems
    - ...
  - Different size of projects defined by:
    - Size of the product to be developed
    - Max size of teams (from 2 to thousands)

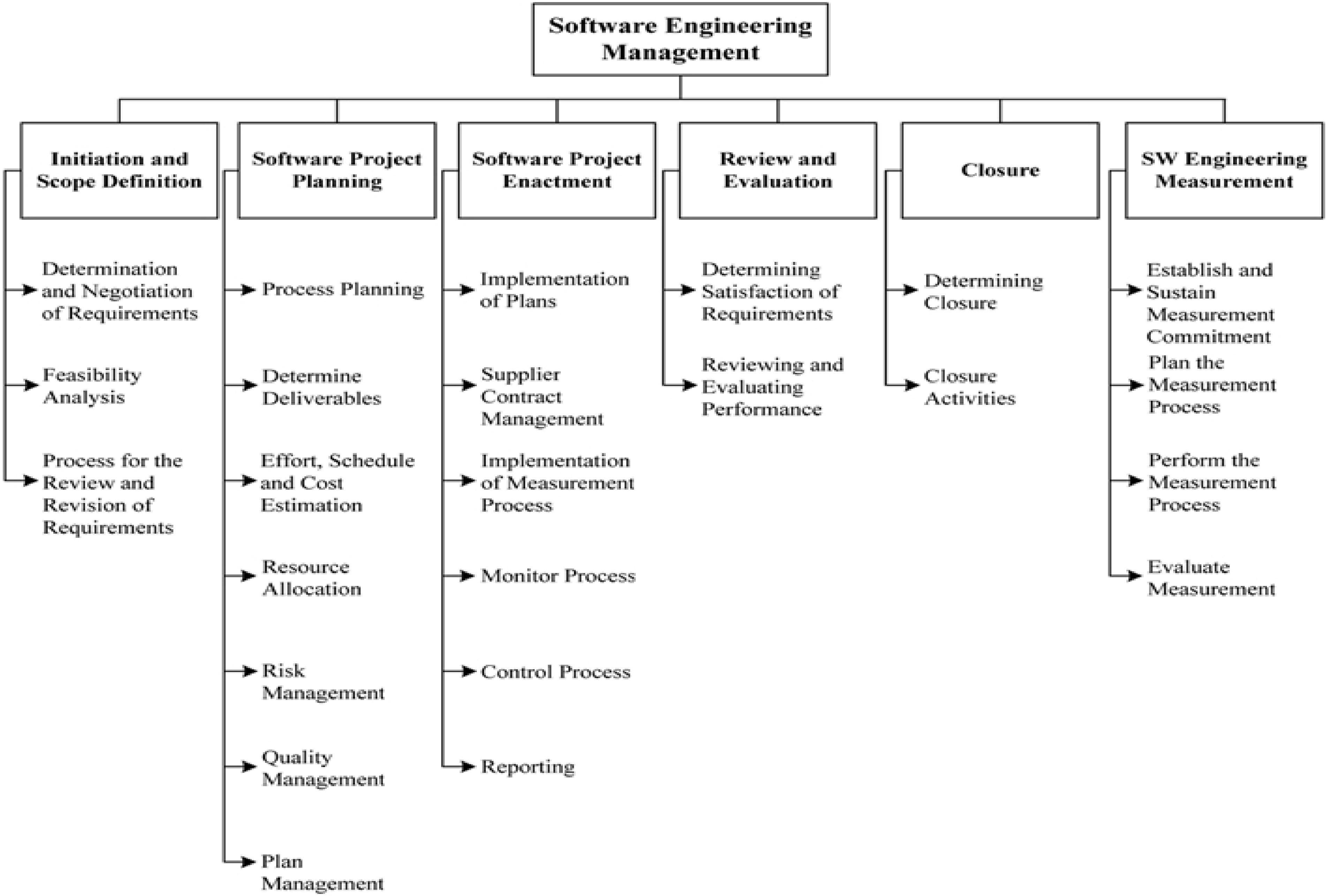
Software Project Management methods need **tailoring** to the environment (*type of the development process, type of software, size of project ...*)

Some course assumptions (the material has to be tailored to other environments)

- Life cycle models and development processes
  - Waterfall
  - Spiral
  - Agile
- Team size 10-20 people in average
- New development

Agenda for today

- Course roadmap
- Basic notions of software project management.
- Course goal and objectives
- Our Software Management Model
- Course assumptions
- SWEBOK
- Next?



Software Engineering Management

- Initiation and Scope Definition
  - Determination and Negotiation of Requirements
  - Feasibility Analysis
  - Process for the Review and Revision of Requirements
- Software Project Planning
  - Process Planning
  - Determine Deliverables
  - Effort, Schedule and Cost Estimation
  - Resource Allocation
  - Risk Management
  - Quality Management
  - Plan Management
- Software Project Enactment
  - Implementation of Plans
  - Supplier Contract Management
  - Implementation of Measurement Process
  - Monitor Process
  - Control Process
  - Reporting
- Review and Evaluation
  - Determining Satisfaction of Requirements
  - Reviewing and Evaluating Performance
- Closure
  - Determining Closure
  - Closure Activities
- SW Engineering Measurement
  - Establish and Sustain Measurement Commitment
  - Plan the Measurement Process
  - Perform the Measurement Process
  - Evaluate Measurement

Software Engineering Management Subareas (SWEBOK)

- Software project management process (5 subareas)
  - *Management process refers to the* activities that are undertaken in order to ensure that the software engineering processes are performed in a manner consistent with the organization’s policies, goals, and standards.
- Software engineering measurement (1 subarea)
  - *Measurement* refers to the assignment of values and labels to aspects of software engineering (products, processes, and resources) and the models that are derived from them, whether these models are developed using statistical or other techniques.
- Selection and use of tools to manage a software engineering project (1 subarea)

Software Engineering Management Subareas (SWEBOK)

- Software project management process (5 subareas)
- Software engineering measurement (1 subarea)
- Why two different subjects (management and measurement) are combined?
  - Management without measurement, qualitative and quantitative, suggests a lack of rigor
  - measurement without management suggests a lack of purpose or context.

Course Topics by Areas
(tentative schedule posted)

- **Part I:** Software Measurement
- **Part II:** Software Project Management
- **Part III**: Agile Estimating and Planning.

To Do List:

- Read chapter 1 of James McDonald’s book (posted)
- Read SWEBOK Chapter 7 (the link is in the syllabus)

Next?

- Lecture 2: Goal-driven software measurement

Questions?

- …