

CZ3002 - Advanced Software Engineering

Introduction to Advanced Software Engineering

Faculty : Dr Shen Zhiqi
School : School of Computer Science and Engineering
Email : zqshen@ntu.edu.sg
Office : N4-02B-43

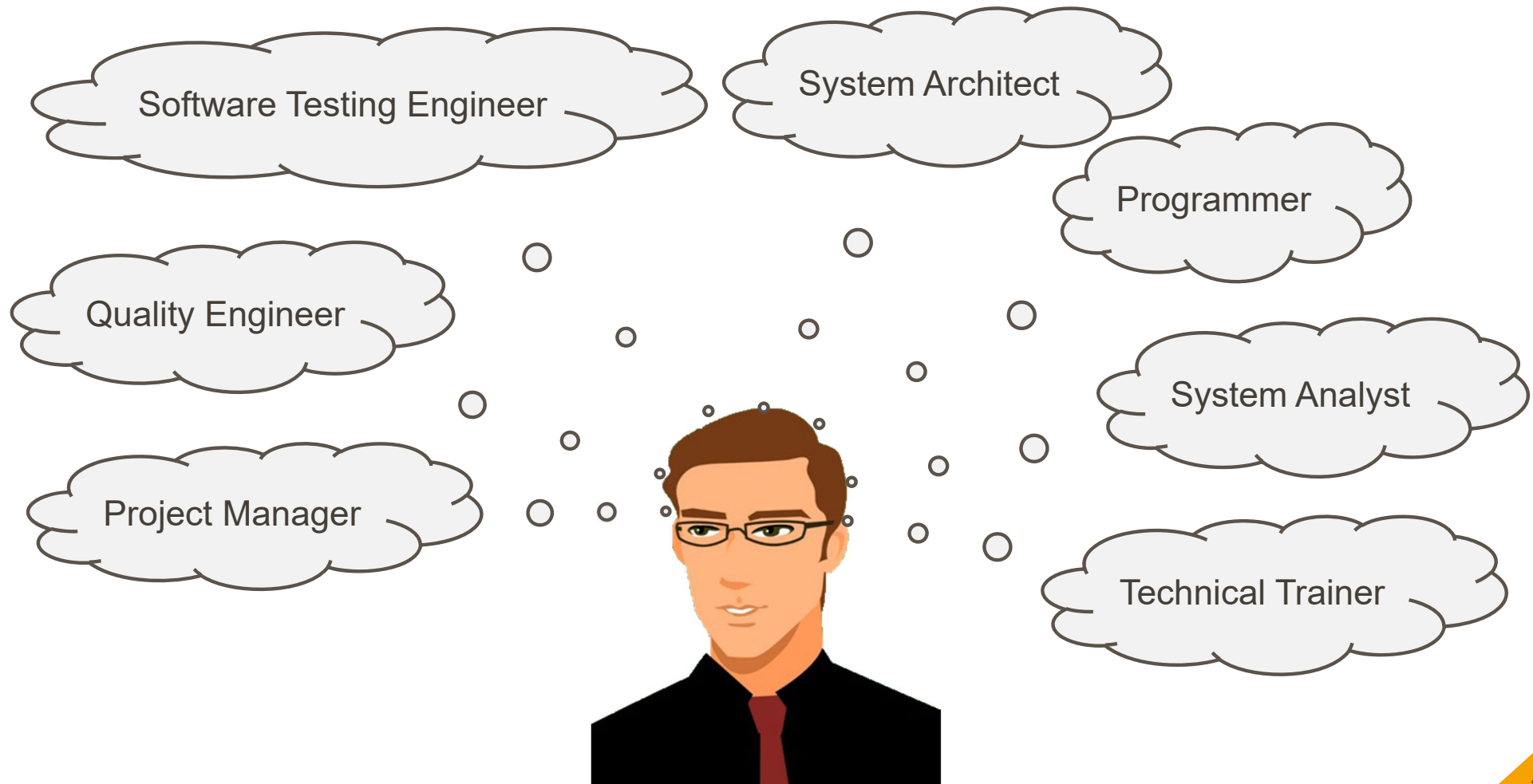
Lesson Objectives

At the end of the lesson, you should be able to:

- ▶ Describe the course requirements
- ▶ Describe the project management lifecycle



Possible Roles in the Industry:



Software Development Components

People



Processes



High Quality Products

Technology



It's Not Just Technological Skills

- ▶ In school, there's a tendency to focus on technology, missing people and process practices.
- ▶ In industry, programming is often what you spend the **least** amount of time doing, overall!

Software Project Management

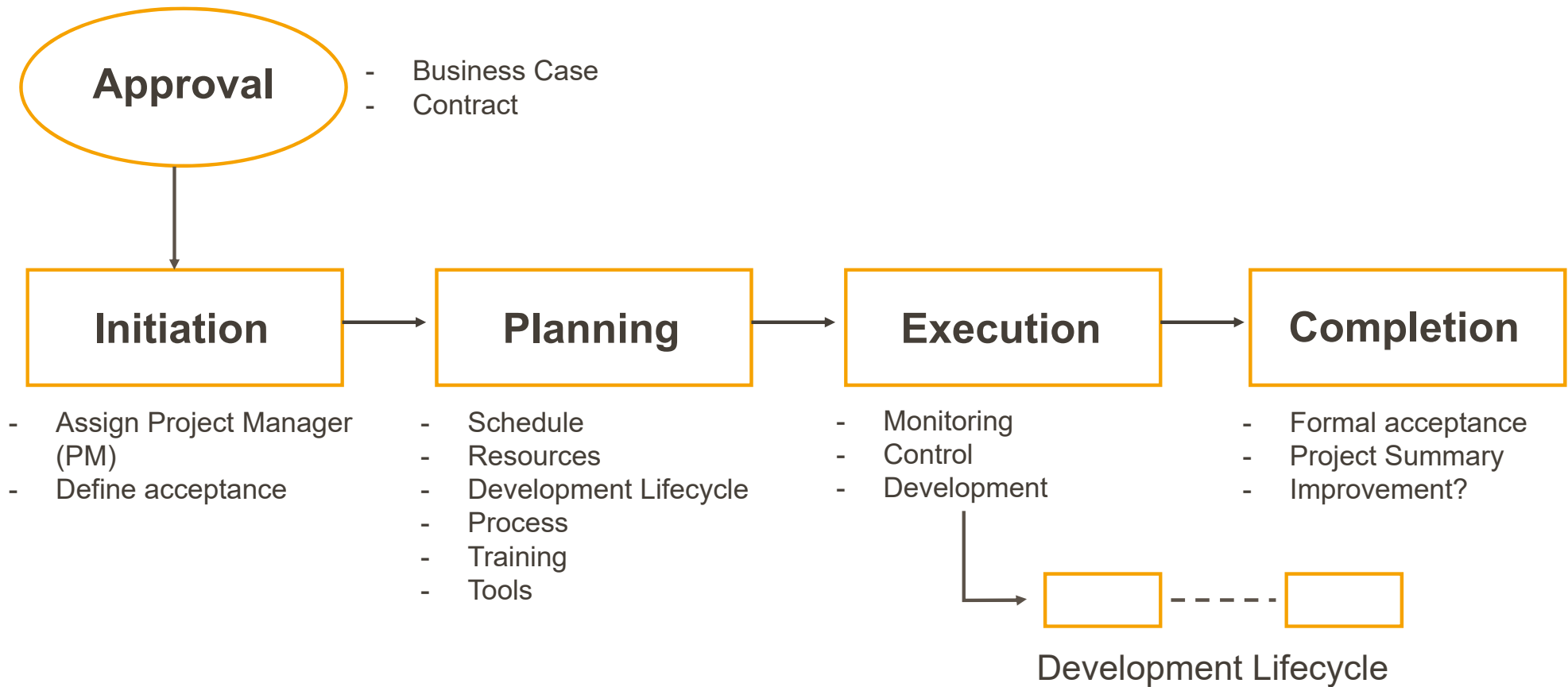
- ▶ Concerned with activities involved in ensuring that software is delivered on time on schedule and within budget.
- ▶ Project management is needed because software development is always subject to budget and schedule constraints that are set by the organisation developing the software.



Management Activities

- ▶ Proposal writing
- ▶ Project planning and scheduling
- ▶ Project costing
- ▶ Project monitoring and reviews
- ▶ Personnel selection and evaluation
- ▶ Report writing and presentations

Project (Management) Lifecycle



Project Initiation Phase

- ▶ Project Manager (PM) identified
- ▶ Project “Board” and/ or “Sponsor(s)” identified:
 - ❖ Who the PM reports to?
 - ❖ Who to ask help from?
 - ❖ Who oversees the project?
- ▶ Acceptance criteria (when are we done?) defined
- ▶ PM collects any existing relevant project material (e.g., contract, business case, customer’s Request for Proposal (RFP) and company’s response, etc.)

Project Planning Phase

- ▶ The initial detailed Project Plan is written and approved by management (& customer)
 - ❖ Goals & objectives
 - ❖ Work breakdown structure (WBS)
 - ❖ Estimates & schedule
 - ❖ Resources (people, tools, hardware, office space,...)
- ▶ Lifecycle, procedures, standards etc. are selected/ tailored
- ▶ Some other activities are started (e.g., training, obtaining tools & hardware, hiring, ...)

Project Execution Phase

- ▶ Work is tracked against the project plan.
- ▶ The project plan is modified as needed.
- ▶ Team meetings (“communication”).
- ▶ Reviews (risks, progress, results, ...).
- ▶ Reporting to upper management & customer.
- ▶ Management support of development effort (people issues, resource issues, ...).

Project Close-Out Phase

- ▶ Customer Acceptance is obtained.
- ▶ Project “File” is completed - all information is collected and archived somewhere.
- ▶ Final Project Review.
- ▶ Improvement recommendations.
- ▶ Hand-over to “operations” & “maintenance”.
- ▶ Team member reviews (employees and contractors).

Course Tutorials and Projects

- ▶ Lectures include lab briefing at beginning week of each lab.
- ▶ Tutorials start from the 3rd week.
 - ❖ Tutorial questions should be done before the tutorial session.
- ▶ Projects start from the 2nd and 3rd week respectively according to your group numbers.

Lab Assignments

- ▶ Lab assignments can be found on course web page!
- ▶ Team formation
 - ❖ 6-7 students per team
 - ❖ From the same session group (SSPx)
 - ❖ 5 labs (2 weeks per lab duration)
 - Finish lab assignment for course requirements
 - 2 hours in-class session per lab

How To Work In A Team?

- ▶ Exploring software project management and development within a team – Role play
- ▶ Getting a feeling for the people, process and technology problems
- ▶ Contributing to the deliverables
 - ❖ Documents (at least two documents for each member)
 - ❖ Prototype development, presentation slides
- ▶ Discuss in team for every deliverable
 - ❖ Meeting minutes, backlogs (form can be found at Lab folder of NTULearn)



Course Projects

- ▶ Lab Manual
 - ❖ Assignments for 5 labs
- ▶ Assessments
 - ❖ Meeting minutes of each lab session
 - ❖ Deliverables of each lab
 - ❖ Prototype system demo in lab 4
 - ❖ Final presentation in lab 5
 - ❖ Peer review

Project Deliverables

- ▶ Documents in 5 labs
 - ❖ Templates and references
- ▶ Demonstration of a live prototype system
 - ❖ Held at in-class session of lab 4
- ▶ Project presentation
 - ❖ Held at in-class session of lab 5
- ▶ Backlogs and meeting minutes
 - ❖ One backlog file per lab
 - ❖ At least two meeting minutes per lab
- ▶ Peer review report

Project Tools/ Templates

- ▶ Templates / References
 - ❖ Based on standard or industry practice
 - ❖ Big coverage (not all from the lecture notes)
 - ❖ Adapt to your project
- ▶ Wiki – Document Management
- ▶ SVN – Version Control
- ▶ Others – from Software Engineering and other courses

Contact

- ▶ Course instructors
- ▶ Lab supervisors
- ▶ Tutorial tutors
- ▶ Lab technicians
- ▶ Teaching assistants

Check the course site on NTULearn for details

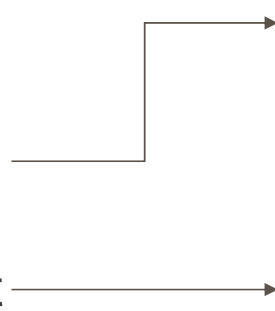
Course Projects

- ▶ Recommendations but not Limited to:
 - ❖ Theme: Activities of Daily Living (ADL)
 - Health care services for elderly daily living
 - System for measuring ADL
 - Data analytics for monitoring and recommendation, etc.
 - ❖ Formats
 - System on Mobile/ Smart phone/ iPad
 - Web based systems
 - Virtual worlds, games, social networks, etc.

Project Proposal

- ▶ Highlight the problem you are going to solve in this project
- ▶ Identify what work is to be done
- ▶ Explain why this work needs to be done
- ▶ Persuade the readers/reviewers/investors to approve/support the project; show that you have a plausible management plan and technical approach, have the resources needed and qualified team for the work to complete the task within the stated time and cost constraints
- ▶ A proposal writing guideline and a template are given under the lab folder of NTULearn
- ▶ You can add/delete/modify the headings/chapters according to your own project.

Project Proposal

- ▶ Executive Summary
 - ▶ Statement of Problem
 - ▶ Objectives
 - ▶ Technical Approach
 - ▶ Project Management
 - ▶ Appendix - Resumes
- ❖ Brief details on the functionalities
 - ❖ Proposed design and architecture
- ❖ Schedule (major milestones without details)
 - ❖ Budget (salary, hardware, software, etc.)
 - ❖ Team members (competency)
- 
- The diagram shows two boxes on the right containing detailed components. The top box, which includes 'Brief details on the functionalities' and 'Proposed design and architecture', is connected by a line to the 'Objectives' and 'Technical Approach' sections. The bottom box, which includes 'Schedule (major milestones without details)', 'Budget (salary, hardware, software, etc.)', and 'Team members (competency)', is connected by a line to the 'Project Management' section.

System Requirement Specifications (SRS)

- ▶ High level requirements of the whole project including requirements of software, hardware, network, database, installation, operation, etc.
- ▶ Different from software requirement specifications which is just for the software part of the project
- ▶ A complete example about university student registration system project is given under the lab folder of NTULearn
- ▶ You can learn from it and/or regard it as a template to work out your one for your project. You can add/delete/modify the headings/chapters according to your own project.
- ▶ Detailed requirement analysis and design of the software using UML are not necessary for this document except for the use case diagram

System Requirement Specifications (SRS)

- ▶ Introduction to the proposed system
- ▶ Settings and constraints
- ▶ System requirements
 - ❖ Operational/ Process requirements
 - ❖ Functional/ Non-Functional requirements
 - ❖ Input/ Output requirements
 - ❖ Hardware/ Software/ Database requirements
 - ❖ Platform/ Network requirements
 - ❖ Deployment/ Installation requirements

- ▶ Act in public interest
- ▶ Act in interest of client and employer, consistent with public interest
- ▶ Ensure quality products
- ▶ Maintain integrity in judgement
- ▶ Manage software development ethically
- ▶ Advance reputation of profession
- ▶ Support colleagues
- ▶ Participate in lifelong learning