Objectives
Software projects have many stakeholders – usually with differing criteria for success. The users want appropriate functionality and a simple user interface. The client looks at time and cost. The development manager wants a profitable project and a long-term reputation for excellence. Perhaps the programming staff just want to have interesting tasks. All these parties’ needs must be met. Most of all, the system has to meet specifications. Thus software engineers must understand methods for reducing the numbers of defects. This lecture will look at the interaction of all these factors.

Key Points
• Students have two critical objectives: a successful project and high marks. The two are (mostly) compatible.
• Software development is an intensely human endeavour. The interactions between the various parties cause much greater variation and problems than issues with the process, techniques or tools. Work to build a strong team and assist each other.
• Quality assurance is about meeting the needs of the customer. In many books, this is defined as minimising defects. However there are other definitions which emphasise the “ilities” and a predictable and consistent project team performance.
• A defect is defined as any divergence from the needs of the customer. Note that this is not necessarily divergence from the specifications.
• Thus time spent on getting the requirements correct and fully understood is seldom wasted. Use as many tools as possible to communicate with the client. Prototyping is probably the most effective.
• Note that client sign-off for the specifications does not guarantee that they understand what you have written. There will be assumptions made and misunderstandings. Be prepared for changes and scope creep. It is your job to minimise mis-communication.
• There are complex (non-linear) trade-offs between the needs of the various stakeholders (see objectives above). The hydraulic model of these interactions. A software project is particularly dynamic – the interactions are continuously shifting.
• It is possible that you will not have enough time to do all the requirements. Therefore an ability to plan and to discuss and agree priorities will the client is essential.
• Risk management is an important part of planning.
• Finally: it is important, given any process we choose, that the plan and priorities are regularly reviewed.

Core reading
Pressman. Chapters on "Project Management" and “Quality Management”
Sommerville. Chapters on "Project Management" and “Quality Management”

Further reading
One of the most easily readable books on the dynamics of software teams is DeMarco, T & Lister, T, Peopleware, Dorset House (2nd ed), 1999.