week 3: Major Issues in SE   
Why are these problem areas for SE: requirments, architecture, change, complexity?

What can go wrong in a software project due to requirments, architecture, change, complexity?

Requirements

Too often we build the “wrong” system Communication between technical and business people is not easy Need precise, accurate description of customer needs Customer does not know what they need

Architecture Where all the demands of requirements “hit home” at once Most non-functional requirements are “holistic” Foundation for (managed) change, maintenance, and evolution

Change The world keeps changing, and we like it that way Software is not easy to change, despite what managers & customers think

Complexity Always trying to build bigger, better, faster, smaller, cheaper Challenge to do things that we have never done before   
What is the impact or consequences of these things going wrong?

* Requirements going wrong: Not delivering the right product
* Over budget
* Not on schedule
* Project rejection & failure

How do we try to prevent these things going wrong? Or lessen their impact?

**Reference: Pressman Chapter 28. Risk management**

* **A risk is potential problem. May or may not happen.**
* **Everyone involved in project should participate in risk analysis and management.**
* **Regardless不管. It is important to identify potential risks: (probability, estimate possible impact, establishes a contingency plan should the problem occur.**
* **Taking measures to avoid the risks and managing them if they occur.**
* **A software team needs to understand and manage uncertainty 不确定性.**
* **Steps: Identify, analyze (probability, impact), have a plan (mitigation).**