

Week 1: Software Engineering

Reading: Pressman Chapters 1 (Software Engineering); Pressman Chapter 2 (Process Models)

1. Develop your own answers to Pressman's five questions about software.
 - 1) Why does it take so long to get software finished?
 - 2) Why are software development costs high?
 - 3) Why can't we find all the errors before software is delivered?
 - 4) Why do we spend so much time and effort to maintain existing software?
 - 5) Why is it so difficult to measure progress as software is developed and maintained?
2. A common problem during communication occurs when you encounter two stakeholders who have conflicting ideas about what the software should be. That is, they have mutually conflicting requirements. Develop a process pattern that addresses this problem and suggest an effective approach to it.
3. Cockburn argues "All organisations have a methodology - it is simply how they do business." and "Your methodology is everything you regularly do to get your software out . . . the conventions your group agrees to."

Consider the 12 items in Cockburn's Elements of a Methodology figure (slide 8 lecture 1):

Process, Milestones, Values
Activities, Techniques, Tools
Teams, Roles, Skills
Quality, Products, Standards

Choose a software project that you have worked on, select a few of the given methodology concepts and describe your methodology. Identify any lessons learned from that project and how you might improve it for future projects.

4. S and J Robertson propose the "brown cow" model as a way of capturing multiple viewpoints, and capturing the "essence" of a problem. Click on the links to read [How Now Brown Cow \(pdf\)](#) and/or watch the [\(video\)](#).

Consider the problem of providing material for learners and teachers in a university course. Identify several different view points for this problem (stakeholders) and describe each one briefly. In groups, each student should role play one of these view points. Then, brainstorm specific items for each of the quadrants: what now; how now; what future and how future. For example, put in the how future quadrant, and then work backwards to discover the what now or what future quadrant requirements. For example, a how-future requirement could be “Access to learning materials should be password protected for enrolled students only”. Identify some “what” requirements for this problem. Concentrate on distinguishing between the what (essence) and how (solution), and on recognising different viewpoints.