

SCM in 90 minutes – Exercises:

In the solutions you discuss, you might want to refer to concepts and principles from the lecture (either from the lecture slides or from the lecture note).

Each group is required to produce a short summary of the main results of their discussions. **A few (1-4) sentences** for each exercise are sufficient. The summary must be mailed to **bendix@cs.lth.se** (in a reasonable format) **at the end of the exercise session**. Do **not** spend time on things like “nice formatting or spelling”. If you have figures/drawings you want to include, then remember the 4P-principle (pen, paper, photo, paste). You can answer in English – or Danish/Italian/Norwegian ;-)

Since there are 11 questions to discuss, you have **an average of 8 min. per question**. It is OK (in fact wise) to spend a lot less time on some questions (maybe with a one-sentence summary) in order to have more time for other questions.

1. In your own words how would you characterize Wayne Babich’ “*Shared Data*” problem and what are the root causes of the problem? Why is it impossible to eliminate (even if you are only one person)?
2. In your own words how would you characterize Wayne Babich’ “*Double Maintenance*” problem and what are the causes of the problem? Why is it impossible to eliminate (even if you are only one person)?
3. Discuss how the analysis of coordination problems from the lecture notes relates to your own personal experience from working in group(s).
4. Find places where the *web-pages* and the *slides* of this course “suffer” from Wayne Babich’ problems of “*Shared Data*” and “*Double Maintenance*”. Discuss different strategies for handling the problems and their advantages and drawbacks.
5. We created the concept of a *workspace* to protect ourselves from other people’s changes. But can the workspace always protect us from other people’s changes? Creating the workspaces, we get the “double maintenance” problem. Can we always handle that problem by merging?
6. We use the workspace to protect ourselves from other people’s changes, but how can we protect the *repository* from people’s bad changes?
7. List three situations from past experience where you have had problems that might have been caused by bad or missing SCM. Pick one and discuss the root causes for the problem and possible SCM solutions.
8. Discuss how the analysis of problems in *handling changes* from the lecture and lecture notes relates to your own personal experience from working in group (or alone).
9. Brainstorm the following questions (in the context of a specific project you have been on):
 - a. What are the goals regarding configuration management?
 - b. What do you need to do to accomplish them (tools, money, time, people)?
 - c. What do you need to do first? Next? If you are a manager? If you are a developer?
 - d. How will you know when you have met your goals?
10. Discuss what kind of coordination problems there may occur between the different groups of people on a project – programmers, testers, designers, requirements engineers, QA, Discuss to what degree the concepts of baseline and traceability could be (partial) solutions to (some of these) problems – and how it should be handled.

Compulsory exercise for all groups: Do a quick brainstorm of what improvements could be made to the lecture, slides, lecture note, and exercises. List at least three things and pick the most important/pressing improvement for a further short discussion. Write a Change Request (CR) for it and email that CR – together with the list – to Lars Bendix (bendix@cs.lth.se). When you write your CR, you might want to discuss what information is needed on the CR form – and on a CR form in a programming context. What would you think of a colleague that sends you a problem report saying: “there is a problem”? I will think the same about you if you send me a CR that says: “it sucks”! I want to know *where* it sucks, *what* it is that sucks, *why* you think it sucks, and – if you know it – an idea for *how* to correct it. What information would you like to have on the CR form as a developer if you have to implement a feature or fix a bug? What information would you like to have on the CR form if you are a product owner or a project manager who must decide whether or not to implement a feature or fix a bug?

After the exercise session you might want to return to the lecture note and re-read and comment the parts that you now find particularly relevant and interesting to you.