Write down merits and drawbacks of all different service management models

File locking:
No concurrent access possible. Once the current developer “checks out” the file, others can read it and after he “checks in” the updated file it can be changed by another developer. Mertis: Good for making radical changes to many sections of a large file. Drawbacks: If the file is locked for too long developers may tend to clone the file and work on the code locally, leading to more serious problems.

Atomic operations:
Operation that applies a set of distinct changes as a single operation. All the changes commited by an atomic commit will be reversed if the process of commiting the changes is interrupted. Only one atomic commit can be processed at a time. Atomic commits are crucial for maintaining a consistent state in the repository. Without atomic commits some changes a developer has made may be applied but other changes may not.

Version merging:
Multiple developers are allowed work on a file at the same time. The system may provide facilities to merge further changes into the central repository, and preserve the changes from the first developer when other developers check in.

Baselines, labels and tags:
Mostly only one of theses similar terms is used by revision control tools. They refer to identifiying a snapshot (“label the project”) or the record of the snapshot (“trying it with baseline X”).

Go on “paper” through at least 3 different typical merge scenario

- Normal merge without any conflict
- Normal merge with conflict -> try to solve conflict (manually) -> merge with no conflict
- Merge of a Branch with the Master Branch

Use GitHub for your group projects, start here http://rogerdudler.github.io/git-guide/
Setup a git repository and share the checkout url
https://github.com/Dani2004/wbis

Discuss the disadvantages of Git based on the article here, http://stevebennett.me/2012/02/24/10-things-i-hate-about-git/ Are these real arguments against Git?

This article is written by someone who is obviously very sure about his opinion towards SVN. He is not considering the advantages that Git brings along in exchange for being a little bit more complicated. Furthermore some things are just not true. Git has a pretty good documentation nowadays. The information model is more complex - yes, but along comes a better model, with more possibilites. As an example we can take the comparison in his argument number 10, “Simple tasks need so many commands”. He compares SVN and Git, SVN has 3 commands to complete a simple task of doing changes and pushing them to the server.

For Git he lists 5:

1. Make some changes
2. git add [not to be confused with svn add]
3. git commit
4. git push
5. Your changes are still only halfway there. Now login to Github, find your commit, and issue a “pull request” so that someone downstream can merge it.

If he would know about Git and see this task without having a predefined opinion, he would state it like this:

1. Make some changes
2. git add & git commit (git commit -am “message”)
3. git push

He is obviously looking for disadvantages and creating some where he can’t even find them.