Radovan: I have moved few small technical issues to GitHub issues close to their respective lessons.

* All material that we use should be reachable for the students, we should not have slides which are only on private computer
* Git configuration should be moved before editing any files
* We should aim to only use lecture rooms with two projector screens installed
* Move DevOps to last day
* focus more on containerization?
* Large conceptual gap between introducing git and DevOps tools
* Show how jupyter can be used in collaborative session, and with version control and git
* Question round asking participants about IDE habits worked very well.
* Editor-off IDE vs. vim/emacs could be fun
* Retrospective in Jupyter session 2 was very good, showed that we adapt and react to questions
* For Jupyter check whether referencing articles with bibtex is possible

-> yes, see https://www.youtube.com/watch?v=m3o1KXA1Rjk

* Documentation lesson should probably have more than one hour
* Timings for our sessions often did not work at all. Some participants were frustrated with us stretching, compressing, sometimes dropping sessions “at will”. We have to respect times. If we have too much for a session, we need to reduce material, we cannot simply expand the session which necessarily takes time away from other sessions and forces them to rush or drop material which frustrates people interested in the session that got dropped or compressed and it also looks chaotic.
* It seems that one cannot copy/paste to/from Git Bash on Windows which makes some exercises tough on Win.
* In testing better explain how the automatic closing of issue really works so that it is not surprising. Explain why local git status claims everything is up to date after pull request was accepted but before pull/fetch was done to update local repo (git diff showed differences but git status was confusing). Need to clarify whether git status communicates with origin.
* Prepare people better before the course starts to not lose time or frustrate those who are prepared.
* It was often only one or two persons debugging red stickers. Perhaps it signals that we don’t know the other lessons well enough to help out.
* Move install instructions to the lessons so that information is not lost or duplicate and diverging (example: Git).
* Generally good to have more real-world examples instead of simplified toy examples, (common feedback from students)