11 Conclusion of software change

• The last phase of software change
• The activities depend on the specific software process
Steps of conclusion

• Commit
  – programmers return their updated code to the configuration management repository
  – resolving conflicts

• New baseline

• New release
New baseline

• A thorough testing is required
  – this testing will guarantee that the new baseline is as bug free as possible
  – the new baseline represents a progress of the project, not a regression

• The baseline testing can take a significant time
  – often done overnight or over the weekend
  – a specialized testing team conducts testing
Baseline testing

• The frequency depends on the size of the program and required quality
• To postpone the baseline testing for too long means that there could be a large accumulation of the bugs
  – it will make the task of the testing difficult
  – too frequent baseline testing can represent an unnecessary overhead
Baseline as a deadline

• Deadline to commit
  – time when the testing of the baseline starts

• Miss the deadline
  – submit by the next deadline
    • additional work needed for the new baseline
    • can be a significant additional work
  – management knows how often a particular programmer missed the deadline
    • may require an explanation
Bugs in baseline

• Programmers commit a faulty code
  – bugs are discovered during the baseline testing

• If the bugs are minor, the testing team can still certify the updated code as the new baseline
  – add correction of these bugs into the stack of the bug reports
  – they will be fixed as a part of the future changes.
Broken baseline

• More serious bugs
  – the testing team can reject the buggy commits

• Sometimes the whole work done on the new baseline has to be rejected
  – no new baseline is created
    • all work that was done is invalidated or postponed
    • a significant cost on large projects
  – testing team can identify the programmers who committed buggy files
  – reputation of these programmers suffers
Stakeholder’s role

• They conduct *acceptance testing*
  – a functional testing done by the stakeholders
  – thoroughly tests software functionalities
• It gives them information about progress of the project
  – stakeholders approve software for the release
New release

• From time to time, the programmers release the baseline code to the users.
  – a substantial extra work is required
  – the frequency of the releases is both a technical and a business decision.

• Less frequent large releases + more frequent small releases
  – application AwesomeApp 4.2
  – “versioned model of software lifespan”
Large and small releases

• Download and install large releases
• Small releases are delivered as patches
  – incorporated into the users program by the tool “merge”