Block 3 Instructor Notes

Personal Software Process for Engineers

# Sequence of Lectures

* Lesson 9: Controlling Quality: Review Your Code / Workshop 9: Defect Analysis
* Lesson 10: Reviewing with a Checklist / 10a: Why Checklists Work / Workshop 10: Building a Checklist for Code Review
* Workshop 10a: Assessing Your Review / Lesson 11: Improving Your Review

# Block 3 Sequence Rationale

Block 3 elaborates the essentials of reviewing code with a personalized checklist. The personalization of the checklist comes from analyzing one’s own defect history in order to populate the checklist with effective items. Thus there is a supposition of at least keeping track of one’s own defects and, to a lesser extent, of knowing how long finding and fixing those defects takes individually and with respect to the overall time spent on development. Both of these skills are introduced in block 1, and nothing in this block is dependent upon skills introduced in block 2. Therefore, if an instructor/coach feels that a team would best be served by learning personal code reviews before learning PROBE, there is no logical dependency between those skills to prevent that.

Lesson 9, “Controlling quality: Review Your Code,” introduces the review as a response to the problem “I’m still finding a lot of defects in test”, with the alternative or concurrent problem of having very imprecise or widely varying effort estimates, even after implement PROBE. This is the very question the instructor-coach should ask about the team (after block 1 skills are established) when deciding whether to introduce block 2 skills or block 3 skills first, namely, which is the bigger problem: estimates or quality? If estimates, do block 2 first; if quality, do block 3 first.

If both estimates and quality are problems, there is a small but measurable quality improvement with improved estimating (i.e. block 2 skills). Many long-time instructors believe that the action of breaking a problem into recognizable parts (i.e. the planning model or conceptual design) is essentially an act of design, a known tactic for improving quality and reducing defects. Block 4 addresses module-level design, but any good planning process requires some minimum of design thinking in order to be effective, and should therefore improve quality to some extent.

The workshop, “Defect Analysis,” is a fundamental skill for the prospective PSP developer. The idea is, based on one’s own data, where in the process are defects injected and removed? This is the fundamental personal quality question, and having the answer to it enables everything that follows in the module.

A key non-programming exercise on actually performing a code review, “EX Code Review”, should be introduced here. For many developers, this will be the first time that they attempt to do a checklist-driven review, and it will provide valuable exposure to the skills involved before they attempt to review their own code.

Lesson 10, “Reviewing with a Checklist,” and the optional-but-highly-recommended 10a, “Why Checklists Work,” provide the basics of what reviews are and how to do them. The workshop, “Building a Checklist for Code Review,” describes a minimal defect analysis and review approach.

The next module begins not with lesson 11 but another workshop, “Assessing Your Review,” that addresses issues that the student-developer may have with performing reviews and introduces the basic review measures. Lesson 11, “Improving Your Reviews,” takes this information and uses it to drive a retrospective analysis intended to make personalized modifications to the review process and the checklist to make them more effective.

The only real option in this block is lesson 10a, “Why Checklists Work,” which as noted is highly recommended. Pushback on the idea of using a checklist is fairly common, so this brief sidebar (only 5 slides in addition to the title and markings slides) should be worthwhile more often than not.

# Block 3 Programming (and Other) Exercises

Because of the workshops in each lesson of this block, it may be necessary to spread the work over multiple sessions. In lesson 9, this would mean doing the lecture and workshop in one session, and the programming exercise in the next.

In lesson 10, there is also the need to make way for the code review exercise, so lesson 10 has the potential to overrun. This is the main reason why the “Why Checklists Work” lesson is optional. One strategy to address and still ensure till to deliver 10a is to simply spread lesson 10 over three sessions, with 10/10a and the code review exercise in one session, workshop 10 and actually building a personal checklist in another session, and the third session to introduce a programming assignment on which to use the checklist for the first time.

Lesson 11 begins with workshop 10a, “Assessing Your Review” which is closing the loop on workshop 10, “Building the Checklist.” “Improving Your Review” then leads into the “Make Change” exercise which tends to be large enough to generate a lot of potential for defects that could be caught by a well-executed review.

| **Lesson / (W)orkshop** | **E Assignment** | **F Assignment** |
| --- | --- | --- |
| 9: Controlling Quality: Review Your Code | T-Distribution | Project Data Actual |
| W9: Defect Analysis |
|  |  |  |
| 10: Reviewing with a Checklist | Code Review Exercise | |
| 10a: Why Checklists Work |
| W10: Building a Checklist for Code Review |  |  |
| W10a: Assessing Your Review |  | Make Change |
| 11: Improving Your Review |

Document Markings

Copyright 2020 Carnegie Mellon University. All rights reserved.  
  
This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.  
  
Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.  
  
NO WARRANTY. THIS MATERIAL IS FURNISHED ON AN “AS-IS” BASIS WITH NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, ANY WARRANTY WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT, OR THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.  
  
[Distribution Statement A] This material has been approved for public release and unlimited distribution. The United States Government has Unlimited Rights in this material as defined by DFARS 252.227-7013.

The text and illustrations in this material are licensed by Carnegie Mellon University under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).

The Creative Commons license does not extend to logos, trade marks, or service marks of Carnegie Mellon University.