Instructor Notes for L01

Personal Software Process for Engineers

# Recommended Order of Presentation

1. Follow the Script (PowerPoint), Script INITIAL (Word)
2. Record Defect Data (PowerPoint), Defect Recording Log (Word), Defect Recording Log Instructions (Word), Defect Type Standard (Word)
3. WS\_E\_01 (PowerPoint), ASGKIT\_E\_Sort (Word), Table D14 p761-62 (Word)

## Follow the Script

Beginning with this lesson, and continuing through the course, we will pose a Problem (in this case, “How do I complete a nontrivial task with minimal error?”) and propose a Solution (“Use a process sScript”), which the rest of the lesson elaborates. The Problem–Solution motif is one way to look at how Humphrey elaborated the original PSP0 scripts into the full-blown PSP3 scripts. The instructor who wishes to navigate a unique path through the class can set the Problem–Solution pairs down in a proposed order to verify that the order makes sense.

The new PSP script “INITIAL” is the proposed starting point for “real PSP” in this module. INITIAL reflects modern software engineering practice arguably better than Humphrey’s original PSP0 scripts, and is shown in snapshot on Slide 6, with the graphic on Slide 7 matching INITIAL phases, although even the word “Phases” could elicit pushback from the hardcore agilist.

The instructor should *not* feel bound to using the INITIAL script. After INITIAL, the instructor will be responsible for enhancing this script appropriately for the materials that follow, since there is no definitive path through these new PSP materials as there was for Humphrey’s PSP. On the other hand, Humphrey’s original scripts are still in use today, especially in academic settings, and are still an exemplar of progressive elaboration of a personal process. They can certainly be used, but the instructor will then have to change or otherwise deal with Slides 6 and 7 as well as the quarter-century of distance in Humphrey’s terminology.

In is strongly recommended that, when enhancing the INITIAL script, the instructor does so in concert with the students in the class offering. This is true whether the class is being presented using the provided exercises or being delivered on the job, that is, PSP data is being collected live by developers as they write production code.

While it might make more obvious sense to introduce Gawande’s *The Checklist Manifesto* when discussing review checklists, Slide 4 is included here because *The Checklist Manifesto* is probably the best popular introduction to the idea of checklists and, by extension, PSP-like scripts. The book is also perhaps the best available refutation of “I don’t need a script/checklist” arguments that extremely intelligent, highly trained (not to say “arrogant”) software developers often make when encountering PSP. There are numerous stories of the use or origin of checklists, including ones where order matters like the PSP process scripts, many from the medical field (Gawande is a world-class surgeon) and many from other fields, such as the original 1935 “we need a checklist” story about the B-17 bomber. Finally, it is well written, easy to read, and a valuable resource for any PSP instructor or practitioner.

## Record Defect Data

This is a description of the Defect Log and related Defect Type Standard. The Word versions of the Defect Log, with Instructions, and the default Defect Type Standard are included with this lesson. A similar module for the Time Log was deemed not necessary because the Time Log is relatively simple. The Defect Log and Defect Type Standard are collectively complex enough to warrant a separate few slides of explanation, and it is possible that a development team, taking the course together, would not want to adapt the existing type standard to their work or would want to use one that already exists. The emphasis here should be what data the PSP requires, especially for later uses in analyzing and using that data.

For the Program Summary, we assume that the instructor will use whatever the chosen local tool uses, nominally the Process Dashboard, and that from this point forward, the tool forms take over. While it worked in previous versions of the course to first show the paper PSP forms, then introduce the electronic version of whatever flavor, that is not to say that it worked well. So we are assuming that the instructor chooses the tool to use and how to introduce it to the students. The Defect Log/Type Standard is the exception that proves the rule.

## WS\_E\_01 (aka Program 1, Sort)

We decided to make the sort program the first activity because it seems to work as a kickoff program regardless of whether the instructor chooses easy math problems (the E series) or hard math problems (the F series). Also, “easy” really means “shorter,” and “hard” means “probably longer than the ‘easy’ ones” since the “hard” are sometimes, though not always, the same as the “easy” with a few additional functions required. The first few programming assignments will have E only. Starting in Block 2, there will be a choice of E (easy) or F (choose your mnemonic for hard) math programs for most lessons.

Table D14 from Humphrey’s *A Discipline for Software Engineering* is included here so that the instructor doesn’t have to find a copy of the text, but any set of paired data can be used. The requirements are to sort on either member of the pair, so you do need data pairs.

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.