Block 4 Instructor Notes

Personal Software Process for Engineers

# Sequence of Lectures

* Lesson 12: Quality by Design
* Lesson 13: Describe the Problem / Operational Scenario / Operational Scenario Exercise (sort)
* Lesson 14: State the Problem / State Template
* Lesson 15: Structure the Solution / Functional Template
* Lesson 16: Implement the Logic / Logic Template
* Lesson 17: Scaling Up

# Block 4 Lessons

Block 4 has the most flexibility of all the blocks in terms of order of delivery. Lesson 12, “Quality by Design,” begins by talking about common design problems that are difficult to find in code mainly because they reflect issues of incorrect or incomplete design. By the end of the module, the concept of design views is introduced. The four distinct pairings of internal vs. external and static vs. dynamic views are presented in table-matrix. Each element of the matrix points to one of the following four lessons and associated view templates.

Lesson 13, “Describe the Problem,” focuses on the external-dynamic design view with the UML use case (external to software system) or sequence diagram (external to individual software modules). The PSP “Operational Scenario” is described in its own slide deck of the same name, with the “Operational Scenario” exercise intended to provide practice for capturing this view of design when necessary.

Lesson 14, “State the Problem,” looks at the internal-dynamic views captured by state charts (sometimes called state diagrams) or state transition tables. In general, if the same input can cause different reactions based on the previous history of inputs, a state machine is probably inside. “State Template” introduces the PSP version of this view, the State Specification Template (SST), and illustrates it with a limited discussion of the notorious “GoToFail” bug.

Lesson 15, “Structure the Problem,” introduces the external-static view of design. In UML this would be one or more class diagrams, or something like a CRC (Class-Responsibilities-Collaborators) card view; in PSP, this would be one or more Functional Specification Templates (FST). This is also the view that is congruent with the planning model/conceptual design introduced in block 2. The “Functional Template” describes the PSP Functional Template and then directs the student to use it to document part of a previous assignment.

Lesson 16, “Implement the Logic,” speaks to the internal-static design view. Many developers are familiar with pseudo-code, a limited-vocabulary “plain-speak” version of the program logic that is often encouraged when one is learning how to code, or when a specific, even formal, implementation of an algorithm or approach is necessary. The “Logic Template” slides describe the PSP Logic Specification Template (LST), but English-like free-form pseudo-code also works.

Lesson 17, “Scaling Up,” strictly speaking does not belong with the design modules, but is included here only because it is the only extant member of block 5 which is intended to cover topics relevant to implementing PSP principles in the large. The current lesson talks about scaling up the individual development process from single-day projects like the programming assignments in this class, to real-world work that spans multiple days or weeks.

# Block 4 Sequencing

As noted above, block 4 should probably always begin with lesson 12, “Quality by Design,” but everything after that is at the discretion of the instructor and the class. If logic problems are common, jump to lesson 16 on “Implementing the Logic.” If state issues abound, try lesson 14, “State the Problem”. If there are struggles with creating a planning model for larger assignments, lesson 15 “Structure the Problem” may be the right choice. If use cases or user stories cause problems, lesson 13 “Describe the Problem” may be an answer.

Lesson 17, as noted above, is not part of design discussions (block 4) but is included here only because it is the only representative of block 5 currently in existence. The subject is standalone in nature. However, “Scaling Up” issues are probably best understood once blocks 1 and 2, and perhaps block 4, have been introduced.

# Block 4 Exercises and Programs

The design block includes modules for

• introduction

• operational scenario

• state modeling

• functional decomposition

• logic description

There workshops include practice exercises applying the tools to prior programs and programming exercises to apply the skills on new problems.

The expected outcome is that developers will be able to apply these techniques and will demonstrate that capability on the programming exercises.

Exercises include the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Module** | **Program Set E** | **Program Set F** | **Other** |
| Quality by Design |  |  | Defect analysis |
| Design review checklist |
| Describe the Problem | Sudoku verification | Sudoku verification |  |
| State the Problem |  |  | State analysis for sort |
| Structure a Solution |  |  |  |
| Implement the Logic |  |  |  |

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.