

Table C55 PSP2 Project Plan Summary

Student	<u>Seth Lemanek</u>	Date	<u>3/6/16</u>	
Program	<u>Assign 6 Range Finder</u>	Program #	<u>6</u>	
Instructor	<u>Arturo Concepcion</u>	Language	<u>C++</u>	
Summary	Plan	Actual	To Date	
LOC/Hour	<u>15</u>	<u>12.4</u>	<u>13.9</u>	
Planned Time	<u>340</u>		<u>340</u>	
Actual Time		<u>494</u>	<u>494</u>	
CPI(Cost-Performance Index)			<u>0.64</u> (Planned/Actual)	
% Reused	<u>15.4%</u>	<u>7.6%</u>	<u>7.6%</u>	
% New Reused	<u>40.5%</u>	<u>23.4%</u>	<u>23.4%</u>	
<i>Test Defects/KLOC</i>	<u>50</u>	<u>43</u>	<u>43</u>	
<i>Total Defects/KLOC</i>	<u>101</u>	<u>128</u>	<u>128</u>	
<i>Yield %</i>	<u>0</u>			
Program Size (LOC):	Plan	Actual	To Date	
Base(B)	<u>75</u> (Measured)	<u>75</u> (Measured)		
Deleted (D)	<u>3</u> (Estimated)	<u>14</u> (Counted)		
Modified (M)	<u>1</u> (Estimated)	<u>4</u> (Counted)		
Added (A)	<u>79</u> (N-M)	<u>97</u> (T-B+D-R)		
Reused (R)	<u>30</u> (Estimated)	<u>13</u> (Counted)	<u>13</u>	
Total New & Changed (N)	<u>94</u> (Estimated)	<u>101</u> (A+M)	<u>350</u>	
Total LOC (T)	<u>195</u> (N+B-M-D+R)	<u>171</u> (Measured)	<u>518</u>	
Total New Reused	<u>79</u>	<u>40</u>	<u>40</u>	
<i>Upper Prediction Interval (70%)</i>	<u>114</u>			
<i>Lower Prediction Interval (70%)</i>	<u>74</u>			
Time in Phase (min.)	Plan	Actual	To Date	To Date %
Planning	<u>64</u>	<u>53</u>	<u>121</u>	<u>7.7</u>
Design	<u>50</u>	<u>116</u>	<u>185</u>	<u>11.8</u>
<i>Design review</i>	<u>20</u>	<u>6</u>	<u>6</u>	<u>0.4</u>
Code	<u>108</u>	<u>121</u>	<u>613</u>	<u>39.1</u>
<i>Code review</i>	<u>30</u>	<u>60</u>	<u>60</u>	<u>3.8</u>
Compile	<u>10</u>	<u>40</u>	<u>115</u>	<u>7.3</u>
Test	<u>30</u>	<u>40</u>	<u>262</u>	<u>16.7</u>
Postmortem	<u>32</u>	<u>58</u>	<u>207</u>	<u>13.2</u>
Total	<u>344</u>	<u>494</u>	<u>1569</u>	<u>100.0</u>
<i>Total Time UPI (70%)</i>	<u>384</u>			
<i>Total Time LPI (70%)</i>	<u>304</u>			

(continued)

Table C55 PSP2 Project Plan Summary (continued)

Student	<u>Seth Lemanek</u>	Date	<u>3/6/16</u>
Program	<u>Assign 6 Range Finder</u>	Program #	<u>6</u>
Instructor	<u>Arturo Concepcion</u>	Language	<u>C++</u>

Defects Injected	<i>Plan</i>	<i>Actual</i>	<i>To Date</i>	<i>To Date %</i>
Planning	<u>0</u>	<u>0</u>	<u>0</u>	<u>0.0</u>
Design	<u>1</u>	<u>0</u>	<u>4</u>	<u>9.1</u>
<i>Design review</i>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0.0</u>
Code	<u>8</u>	<u>9</u>	<u>37</u>	<u>84.1</u>
<i>Code review</i>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0.0</u>
Compile	<u>0</u>	<u>2</u>	<u>2</u>	<u>4.5</u>
Test	<u>0</u>	<u>1</u>	<u>1</u>	<u>2.3</u>
Total Development	<u>9</u>	<u>12</u>	<u>44</u>	<u>100.0</u>

Defects Removed	<i>Plan</i>	<i>Actual</i>	<i>To Date</i>	<i>To Date %</i>
Planning	<u>0</u>	<u>0</u>	<u>0</u>	<u>0.0</u>
Design	<u>0</u>	<u>0</u>	<u>0</u>	<u>0.0</u>
<i>Design review</i>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0.0</u>
Code	<u>1</u>	<u>0</u>	<u>5</u>	<u>11.4</u>
<i>Code review</i>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0.0</u>
Compile	<u>5</u>	<u>8</u>	<u>23</u>	<u>52.3</u>
Test	<u>3</u>	<u>4</u>	<u>16</u>	<u>36.4</u>
Total Development	<u>9</u>	<u>12</u>	<u>44</u>	<u>100.0</u>
After Development				

<i>Defect Removal Efficiency</i>	<i>Plan</i>	<i>Actual</i>	<i>To Date</i>
<i>Defects/Hour - Design review</i>			
<i>Defects/Hour - Code review</i>			
<i>Defects/Hour - Compile</i>			
<i>Defects/Hour - Test</i>			
<i>DRL(DLDR/UT)</i>			
<i>DRL(CodeReview/UT)</i>			
<i>DRL(Compile/UT)</i>			

Table C57 C++ PSP2 Design Review Checklist

PROGRAM NAME AND #:

Purpose	To guide you in conducting an effective design review				
General	As you complete each review step, check that item in the box to the right. Complete the checklist for one program unit before you start to review the next.				
Complete	Ensure that the requirements, specifications, and high-level design are completely covered by the design: - all specified outputs are produced - all needed inputs are furnished - all required includes are stated				
Logic	Verify that program sequencing is proper: - that stacks, lists, etc. are in the proper order - that recursion unwinds properly Verify that all loops are properly initiated, incremented, and terminated				
Special Cases	Check all special cases: - empty, full, minimum, maximum, negative, zero - out of limits, overflow, underflow - ensure "impossible" conditions are absolutely impossible - handle all incorrect input conditions				
Functional use	Verify that all functions, procedures, or objects are fully understood and properly used Verify that all externally referenced abstractions are precisely defined				
Names	Verify that: - all special names and types are clear or specifically defined - the scopes of all variables and parameters are self-evident or defined - all named objects are used within their declared scopes				
Standards	Review the design for conformance to all applicable design standards				

Table C58 C++ Code Review Checklist

PROGRAM NAME AND #:

Purpose	To guide you in conducting an effective code review.				
General	As you complete each review step, check that item in the box to the right. Complete the checklist for one program unit before you start to review the next.				
Complete	Verify that the code covers all the design.				
Includes	Verify that includes are complete				
Initialization	Check variable and parameter initialization: - at program initiation - at start of every loop - at function/procedure entry				
Calls	Check function call formats: - pointers - parameters - use of '&'				
Names	Check name spelling and use: - is it consistent? - is it within declared scope? - do all structures and classes use '.' reference?				
Strings	Check that all strings are - identified by pointers and - terminated in NULL.				
Pointers	Check that - pointers are initialized NULL - pointers are deleted only after new, and - new pointers are always deleted after use.				
Output Format	Check the output format: - line stepping is proper - spacing is proper				
{ } Pairs	Ensure that the { } are proper and matched				
Logic Operators	Verify the proper use of ==, =, , and so on. Check every logic function for proper ().				
Line by Line Check	Check every LOC for - instruction syntax and - proper punctuation.				
Standards	Ensure that the code conforms to the coding standards.				
File Open and Close	Verify that all files are - properly declared, - opened, and - closed.				

Table C39 Size Estimating Template

Student	<u>Seth Lemanek</u>				Date	<u>3/6/16</u>
Instructor	<u>Assign 6 Range Finder</u>				Program #	<u>6</u>
BASE PROGRAM					LOC	
BASE SIZE (B)	=>	=>	=>	=>	=>	<u>75</u>
LOC DELETED (D)	=>	=>	=>	=>	=>	<u>3</u>
LOC MODIFIED (M)	=>	=>	=>	=>	=>	<u>1</u>
PROJECTED LOC						
BASE ADDITIONS:	TYPE	METHODS	REL. SIZE	LOC		
TOTAL BASE ADDITIONS (BA)					=>	<u>0</u>
NEW OBJECTS:	TYPE ¹	METHODS	REL. SIZE	LOC (NewReuse*)		
Range Predictor*	Calc	7	M	<u>78.75</u>		
TOTAL NEW OBJECTS (NO)					=>	<u>0</u>
REUSED PROGRAMS					LOC	
Program #4					<u>30</u>	
REUSED TOTAL (R)					=>	<u>30</u>
Projected LOC:	P = BA+NO				<u>78.75</u>	
Regression Parameter:	β_0				<u>0</u>	
Regression Parameter:	β_1				<u>1.18</u>	
Estimated New and Changed LOC:	$N = \beta_0 + \beta_1 * (P+M)$				<u>94</u>	
Estimated Total LOC:	$T = N + B - D - M + R$				<u>195</u>	
Estimated Total New Reused (sum of * LOC):					<u>79</u>	
Prediction Range:	Range				<u>20</u>	
Upper Prediction Interval:	UPI = N + Range				<u>114</u>	
Lower Prediction Interval:	LPI = N - Range				<u>74</u>	
Prediction Interval Percent:						

time (min)

0

4.364

344

40

384

304

¹ L-Logic, I-I/O, C-Calculation, T-Text, D-Data, S-Set-up