Table C55 PSP2 Project Plan Summary

Student	Seth Lemanek	Date	3/13/16
Program	List.cpp	Program #	7
Instructor	Arturo Concepcion	Language	C++

Summary	**Plan**	**Actual**	**To Date**
LOC/Hour	14	18.7	16.7
Planned Time	384		384
Actual Time		491	491
CPI(Cost-Performance Index)			0.78 (Planned/Actual)
% Reused	0	0	7.6%
% New Reused	80%	73.8%	48.6%
Test Defects/KLOC	50	20	32
Total Defects/KLOC	100	92	96
Yield %	50%	21.7%	21.7%

Program Size (LOC):	**Plan**	**Actual**	**To Date**
Base(B)	0 (Measured)	0 (Measured)	
Deleted (D)	0 (Estimated)	0 (Counted)	
Modified (M)	0 (Estimated)	0 (Counted)	
Added (A)	86 (N-M)	153 (T-B+D-R)	
Reused (R)	0 (Estimated)	0 (Counted)	13
Total New & Changed (N)	86 (Estimated)	153 (A+M)	503
Total LOC (T)	86 (N+B-M-D+R)	153 (Measured)	671
Total New Reused	0	113	153
Upper Prediction Interval (70%)	106		
Lower Prediction Interval (70%)	66		

Time in Phase (min.)	**Plan**	**Actual**	**To Date**	**To Date %**
Planning	29	53	174	8.5
Design	45	22	207	10.1
Design review	10	48	52	2.5
Code	148	72	685	33.3
Code review	10	42	102	5.0
Compile	27	27	142	6.9
Test	63	170	432	21.0
Postmortem	50	57	264	12.8
Total	382	491	2058	100.0
Total Time UPI (70%)	402			
Total Time LPI (70%)	362			

(continued)

Table C55 PSP2 Project Plan Summary (continued)

Student	Seth Lemanek	Date	3/13/16
Program	List.cpp	Program #	7
Instructor	Arturo Concepcion	Language	C++

Defects Injected	***Plan***	**Actual**	**To Date**	**To Date %**
Planning	0	0	0	0.0
Design	1	0	4	6.9
Design review	0	0	0	0.0
Code	8	13	50	86.2
Code review	0	0	0	0.0
Compile	0	1	3	5.2
Test	0	0	1	1.7
Total Development	9	14	58	100.0

Defects Removed	***Plan***	**Actual**	**To Date**	**To Date %**
Planning	0	0	0	0.0
Design	0	0	0	0.0
Design review	1	0	0	0.0
Code	0	0	5	8.6
Code review	1	0	0	0.0
Compile	4	11	34	58.6
Test	3	3	19	32.8
Total Development	9	14	58	100.0
After Development				

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

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 Seth Lemanek Date 3/8/16

Instructor Arturo Concepcion Program # 7

BASE PROGRAM LOC	ESTIMATE	ACTUAL
BASE SIZE (B) => => => => => => => => =>		
LOC DELETED (D) => => => => => => => =>		
LOC MODIFIED (M) => => => => => => => =>		

OBJECT LOC

BASE ADDITIONS	TYPE[1]	METHODS	REL. SIZE	LOC	LOC
TOTAL BASE ADDITIONS (BA) => => => => => =>					

NEW OBJECTS	TYPE	METHODS	REL. SIZE	LOC (New Reused*)	
TOTAL NEW OBJECTS (NO) => => => => => => =>					

REUSED OBJECTS

REUSED TOTAL (R) => => => => => => =>		

		SIZE	TIME
Estimated Object LOC (E):	E=BA+NO+M		
Regression Parameters:	β_0 (size and time)		
Regression Parameters:	β_1 (size and time)		
Estimated New and Changed LOC (N):	N=β_0+β_1*E		
Estimated Total LOC:	T=N+B−D−M+R		
Estimated Total New Reuse (sum of * LOC):			
Estimated Total Development Time:	Time=β_0+β_1*E		
Prediction Range:	Range		
Upper Prediction Interval:	UPI=N+Range		
Lower Prediction Interval:	LPI=N−Range		
Prediction Interval Percent:			

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