Report for Iteration 3

1. Chosen Methods

White box critical tests:

linkNodeForward (In PertNetwork.java class)

Black box critical tests:

getActivityScheduleValue (In EarnedValue.java class)

areValidPercentages (In Project.java class)

areValidValues (In Project.java class)

areValidTimes (In Activity.java class)

areValidPercentAndTimes (In Activity.java class)

2. Justification and Ranges

White box tests:

Black box tests:

Function: getActivityScheduleValue

getActivityScheduleValue is an important function for calculating the planned value of an activity during an Earned Value Analysis procedure. Given its nature of taking 5 parameters of the type Double, and also its importance in the application, it is a perfect candidate for boundary box testing.

Ranges:

Function: areValidPercentages

areValidPercentages is an input validity checking function for use in the constructor of Project. It is necessary to ensure data integrity and proper program functionality. It checks that the Project’s percentages are within suitable ranges (not smaller than 0 and not larger than 1).

Ranges:

0.0 ≤ percentage1 ≤ 1

0.0 ≤ percentage2 ≤ 1

Function: areValidValues

areValidValues is an input validity checking function for use in the constructor of Project. It is necessary to ensure data integrity and proper program functionality. It checks that the Project’s values (budgetAtCompletion, actualCost and earnedValue) are within suitable ranges (not smaller than 0 and not larger than MAX\_COST).

Ranges:

0.0 ≤ value1 ≤ MAX\_COST (constant, current value is: 10,000,000)

0.0 ≤ value2 ≤ MAX\_COST (10,000,000)

0.0 ≤ value3 ≤ MAX\_COST (10,000,000)

Function: areValidTimes

areValidValues is an input validity checking function for use in the constructor of Activity. It is necessary to ensure data integrity and proper program functionality. It checks that the Activities’ projected times (mostLikely, optimistic, pessimistic, targetCompletionDate) are within suitable ranges (not smaller than 0 and not larger than MAX\_DURATION), and also that the values are appropriately set relative to one another (For example, mostLikely should be smaller than or equal to pessimistic, etc.).

Ranges:

0.0 ≤ mostLikely ≤ Activity.MAX\_DURATION (constant, current value is: 731).

0.0 ≤ optimistic ≤ Activity.MAX\_DURATION (731).

0.0 ≤ pessimistic ≤ Activity.MAX\_DURATION (731).

Function: areValidPercentAndCost

areValidPercentAndCost is an input validity checking function for use in the constructor of Activity. It is necessary to ensure data integrity and proper program functionality. It checks that the Activities’ percent and cost values are within suitable ranges (like that percentComplete is larger than or equal to 0 but smaller than or equal to 1, and that actual cost is larger than or equal to 0 but smaller than MAX\_COST).

Ranges:

0.0 ≤ minPercent ≤ 1.0

0 ≤ minCost ≤ 1,000,000

3. White Box Testing

Function: getActivityScheduleValue

Cyclomatic complexity: 4

Basis Path Coverage:

<1,2,7>, <1,3,4,5,6,7>, <1,3,7>, <1,3,4,7>



Functions: areValidPercentages, areValidValues, areValidCosts

(These functions have identical CFG’s)

Cyclomatic complexity (of each): 3

Basis Path Coverage:

<1,2,5>, <1,3,2,5>, <1,3,4,5>



Function: areValidTimes

Cyclomatic complexity: 4

Basis Path Coverage:

<1,2,6>, <1,3,2,6>, <1,3,4,2,6>, <1,3,4,5,6>



4. Black Box Testing

5. Implement, Run and Test Reports

See code for implementation.

Report:

6. Follow-up