ASSIGNMENT 3

**DESIGN AND BUILD**

**A**

**SIZE ESTIMATING SPREADSHEET**

**CSE 7315**

**PLANNING AND MANAGING A SOFTWARE PROJECT**

***Spring, 2015***

Professor Dennis J. Frailey

NAME \_\_***RUTUJA MANDHARE***  LOCATION \_\_\_***SPRING 2015***

THIS COVER SHEET SHOULD APPEAR AS THE FIRST PAGE.

Grading template Student do not write inside this box

\_\_\_\_\_\_\_\_\_\_\_/30 1.0 Total Software Size Estimate Summary

\_\_\_\_\_\_\_\_\_\_/ 2.5 Description of overall SW System

\_\_\_\_\_\_\_\_\_\_/ 2.5 Assumptions and Unknowns about SW System

\_\_\_\_\_\_\_\_\_\_/ 10 Actual Size Table for Complete SW System

\_\_\_\_\_\_\_\_\_\_/ 10 Equivalent Size Table for Complete SW System

\_\_\_\_\_\_\_\_\_\_/ 5 Reuse Graph for Complete SW System

\_\_\_\_\_\_\_\_\_\_\_/60 2.0 Software Item Size Estimates

\_\_\_\_\_\_\_\_\_\_/ 8 Descriptions, Assumptions & Unknowns (2 each)

\_\_\_\_\_\_\_\_\_\_/ 20 Actual Size Table (5 each)

\_\_\_\_\_\_\_\_\_\_/ 20 Equivalent Size Table (5 each)

\_\_\_\_\_\_\_\_\_\_/ 12 Reuse Graphs (3 each)

\_\_\_\_\_\_\_\_\_\_\_/10 Overall Report Format (10)

\_\_\_\_\_\_\_\_\_\_\_/100 Total Assignment Grade

Table of Contents

[1.0 Total Software Size Estimate Summary 4](#_Toc384249673)

[1.1 Description of overall software system 4](#_Toc384249674)

[1.2 Assumptions and other notes, such as rationale for reuse factors 5](#_Toc384249675)

[1.3 Actual Size Table, per P&P 6](#_Toc384249676)

[1.4 Equivalent Size Table, per P&P 7](#_Toc384249677)

[1.5 Reuse Chart(s) for Complete System (all software) 8](#_Toc384249678)

[2.0 Individual Software Item Size Estimate Summary 8](#_Toc384249679)

[2.1 Software Item #1 8](#_Toc384249680)

[2.1.1 Assumptions and other notes, such as rationale for reuse factors 8](#_Toc384249681)

[2.1.2 Actual Size Table 9](#_Toc384249682)

[2.1.3 Equivalent Size Table 11](#_Toc384249683)

[2.1.4 Bar Chart showing new, reused and modified 12](#_Toc384249684)

[2.2 Software Item #2 12](#_Toc384249685)

[2.2.1 Assumptions and other notes, such as rationale for reuse factors 12](#_Toc384249686)

[2.2.2 Actual Size Table 12](#_Toc384249687)

[2.2.3 Equivalent Size Table 13](#_Toc384249688)

[2.2.4 Bar Chart showing new, reused and modified 14](#_Toc384249689)

[2.3 Software Item #3 14](#_Toc384249690)

[2.3.1 Assumptions and other notes, such as rationale for reuse factors 14](#_Toc384249691)

[2.3.2 Actual Size Table 14](#_Toc384249692)

[2.3.3 Equivalent Size Table 16](#_Toc384249693)

[2.3.4 Bar Chart showing new, reused and modified 17](#_Toc384249694)

[2.4 Software Item #4 17](#_Toc384249695)

[2.4.1 Assumptions and other notes, such as rationale for reuse factors 17](#_Toc384249696)

[2.4.2 Actual Size Table 17](#_Toc384249697)

[2.4.3 Equivalent Size Table 1](#_Toc384249698)9

[2.4.4 Bar Chart showing new, reused and modified](#_Toc384249699) 20

**SIZE ESTIMATION REPORT**

# 1.0Total Software Size Estimate Summary

## 1.1 Description of overall software system

"Mr. Television" is an upgrade venture having four Software Processors. Each of the Processors has an individual working framework and Application programming. Operating system is treated as a component and also the application software may have more components. All the processor and their software items are described below:

* **Operating System:** The operating systems are identical in each processor. Operating system has very large caches, tables and buffers, and thus uses 64 Megabytes of the internal memory on its respective processor.
* **Processor 1:** It has a Image capture Application which copies a frame of data from the camera to a graphics data memory. All the components are described below:
* Frame Grabber – Copies a frame of data from the camera to one of the data memories. This processor also provides the master timing signal.
* Bit Twiddler – Performs minor data correction functions on the data from the camera.
* Miscellaneous stuff - various minor housekeeping functions. Its main function is to provide the master timing signal to the other processors. Each of the other processors sends a signal indicating that it is complete with its current function.
* **Processor 2:**
* Expert System for Image Enhancement – Improves the image. This is the major function of the “Mr. Television” product.
* Bit Twiddler – Similar to Software Item #1.
* **Processor 3:**
* Frame Putter, which is similar to the Frame Grabber in Software Item #1.
* It also needs an almost identical bit twiddler, but a somewhat different version of the miscellaneous stuff.
* **Processor 4:**
* Implements the sound functions and it processes sound, fully synchronizing it with the video output of processor 3.
* It is fully independent of the other processors.

## 1.2Assumptions and other notes, such as rationale for reuse factors

|  |  |
| --- | --- |
| **Assumptions** | **Notes** |
| Operating System is being implemented for processor#1 and its being concurrently reused in other processors. | The Expected LOC from the actual table is added to the COTS/Reuse in the Equivalent table of Processor #2, Processor #3, and Processor #4. |
| Memory requirements for different languages as per SOW document. They are not accurate as they are based on different scenarios and are just assumed to be approximate. | As per SOW document - Python- 7.5 bytes per line. C- 5.5 bytes per line. Java - 11.5 bytes per line |
| Equivalent effort ratio for modified, reuse and Concurrent reuse components. | Modify-60%\*LOC. Reuse – 30%\*LOC. Concurrent reuse – 2%\*LOC. |

## 

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SOFTWARE ITEM:** | **ACTUAL SIZE** | | | | |
| **SW SYSTEM** | **SOURCE LINES OF CODE** | | | | **BYTES OF TARGET** |
| **OR** | | | |
| **FUNCTION POINTS** | | | |
| ***COMPONENTS*** | ***MINIMUM*** | ***LIKELY*** | ***MAXIMUM*** | ***EXPECTED*** | ***MEMORY*** |
| **PROCESSOR #1** | 60864 | 116583 | 172227 | 116570 | 64091500 |
| **PROCESSOR #2** | 70864 | 128583 | 195500 | 130116 | 114007500 |
| **PROCESSOR #3** | 60864 | 116583 | 172227 | 116570 | 64091500 |  | |
| **PROCESSOR #4** | 52042 | 104728 | 158000 | 104826 | 64040000 |  |
| Total for SW Item | 244634 | 466477 | 697954 | 468082 | 306230500 |

## 1.3 Actual Size Table, per P&P

The Actual size table for the entire system is described below:

## 

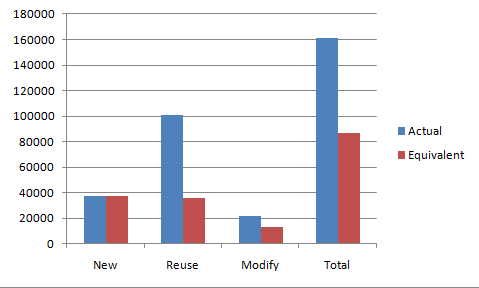
## 1.4 Equivalent Size Table, per P&P

The Equivalent size table for the entire system is described below:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOFTWARE ITEM :** | **NEW** | **COTS/REUSED CODE** | | | **MODIFIED CODE** | | | **NET** |  |  |  |
| **CODE** | **SLOC or Function Pts.** | | | **SLOC or Function Pts.** | | | **EQUIV.** |  |  |  |
| **(SLOC or** |  | | |  | | | **SIZE** | **C** | **JAVA** | **Python** |
| **PROCESSOR #1** | **Function** |  |  |  |  |  |  |  |
| **Points)** | **ORIGINAL** | **EQUIV.** | **EQUIV.** | **ORIGINAL** | **EQUIV.** | **EQUIV.** |  |
| ***COMPONENTS*** |  |  | ***RATIO*** |  |  | ***RATIO*** |  |  |
| **PROCESSOR #1** | 11000 | 93371 | 0.3 | 28011 | 12199 | 0.6 | 7319.4 | 46330 | 46330 | 0 | 0 |
| **PROCESSOR #2** | 21667 | 101250 | 0.02 | 2684 | 5000 | 0.6 | 3000 | 27351 | 2684 | 0 | 24667 |
| **PROCESSOR #3** | 5000 | 106570 | 0.02 | 3621 | 5000 | 0.6 | 3000 | 11621 | 11621 | 0 | 0 |
| **PROCESSOR #4** | 0 | 104826 | 0.02 | 2096 | 0 | 0.6 | 0 | 2096 | 2025 | 0 | 71 |
| Total | 37667 | 101504 |  | 36412 | 22199 |  | 13319 | 87398 |  |  |  |

## 

## 1.5Reuse Chart(s) for Complete System (all software)



## 

# 2.1 Software Item #1

## 2.1.1 Assumptions and other notes, such as rationale for reuse factors

Frame Grabber- As per the system design summary document, the system engineers believe that it will take the same size estimates as they implemented in BASIC previously.

Miscellaneous stuff - There are many minor housekeeping functions but the details are unknown. Other products have required a small amount of functionality in this category. The estimation is taking into account the involvement with different items.

Operating System - It is assumed that estimation for OS will be calculated for processor #1 and would be considered concurrent reuse in other processors.

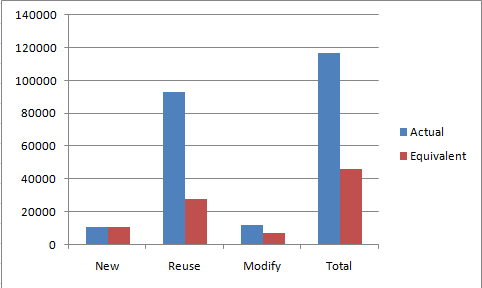
## 2.1.2 Actual Size Table

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Software Item:  Processor#1 | Components or Description | WBS &SOW #S: | Actual Size | | | | | Bytes of Target memory |
| Source lines of code | | | | |  |
| Components | Comments | Language | Minimum | Likely | Maximum | Expected | Comment |  |
| Frame Grabber | Modified- Already implemented in BASIC and needs to be modified in C | C | 8000 | 10000 | 12727 | 10121.16 | Minimum-5000 (new)+60% \*5000(m0dified)+ 0%\*5000(reused).Reused is free because ” Dee Zign” rewrote that part for another project. The above estimation has been taken from Results of Initial Size Estimate Preparation Meeting  Document.  Maximum- For C language, 5.5 bytes of memory per line is needed and as per system design summary, it takes 70000 bytes of memory. 70000/5.5=12,727  Likely- From the System design summary, sit is believed to be about 10000 LOC | 70000 |
| Bit Twiddler | Modified- already implemented in assembly language by Harry Hacker. | C | 1364 | 2333 | 2500 | 2199.33 | Minimum- For C language, 5.5 bytes of memory per line is needed and as per system design summary, it takes 7500 bytes of memory. 7500/5.5=1364.  Maximum- Estimation has been taken from Results of Initial Size Estimate Preparation Meeting as per Harry Hacker discussion.  Likely- As per Harry Hacker, it takes 3-2 ratio of assembly lines to ‘C’ lines in previous efforts. 3500 lines of assembly code is converted into C code. | 7500 |
| Miscellaneous Stuff | New-Considered as new even though there it was previously implemented in assembly language because there are no comments, requirements ,design, information, test, etc | C | 2000 | 3000 | 4000 | 3000 | Minimum-System design summary, Estimated as 2000 LOC.  Maximum- Result of initial Size estimate preparation meeting. Harry estimates to be 4000.  Likely- Mean of max and minimum. | 14000 |
| Operating System | New/Modified/Reuse- There is already an existing OS having some bugs and therefore it needs some minimal modification | C | 49500 | 101250 | 153000 | 101250 | Minimum-New code+0.6 times the modified code0.3%+0.3 times the reuse component  Maximum- Since the operating system has some bugs assuming the worst case it will be the total of modified, reused and new.  Likely- Mean of max and minimum. | 64000000 |
| Total for software item |  |  | 60864 | 116583 | 172227 | 116570 |  | 64091500 |
|  |  |  |  |  |  |  |  |  |

**2.1.3Equivalent Size Table**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOFTWARE ITEM :** | **WBS & SOW #s:** | **NEW** | **COTS/REUSED CODE** | | | **MODIFIED CODE** | | | **NET** |
| **CODE** | **SLOC or Function Pts.** | | | **SLOC or Function Pts.** | | | **EQUIV.** |
| **(SLOC or** |  | | |  | | | **SIZE** |
| **PROCESSOR #1** | **SOW** | **Function** |  |  |  |  |  |  |  |
| **WBS** | **Points)** | **ORIGINAL** | **EQUIV.** | **EQUIV.** | **ORIGINAL** | **EQUIV.** | **EQUIV.** |  |
| ***COMPONENTS*** | ***LANGUAGE*** |  |  | ***RATIO*** |  |  | ***RATIO*** |  |  |
| Frame Grabber | C | 5000 | 121 | 0.3 | 36.3 | 5000 | 0.6 | 3000 | 8036.3 |
| Bit Twiddler | C | - | - | 0.3 | - | 2199.33 | 0.6 | 1319.59 | 1319.59 |
| Miscellaneous Stuff | C | 3000 | - | 0.3 | - | - | 0.6 | - | 3000 |
| Operating System | C | 3000 | 93250 | 0.3 | 27975 | 5000 | 0.6 | 3000 | 33975 |
| Total for Python | Python | 0 | - |  | 0 | 0 |  | 0 | 0 |
| Total for C | C | 11000 | 93371 | 0.3 | 28011 | 12199 | 0.6 | 7319 | 46330 |
| Total for Java | JAVA | 0 |  |  | 0 | 0 |  | 0 | 0 |
| Total |  | 11000 | 93371 | 0.3 | 28011.3 | 12199.33 | 0.6 | 7319.59 | 46330 |

## 2.1.4 Bar Chart showing new, reused and modified



# 2.2 Software Item #2

## 2.2.1 Assumptions and other notes, such as rationale for reuse factors

Expert System - Based on Size estimation preparation memo, the estimates are also based on the prototype.

Operating System - It is a concurrent reuse from first.

## 2.2.2 Actual Size Table

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Software Item:  Processor#2 | Components or Description | WBS &SOW #S: | Actual Size | | | | | Bytes of Target memory |
| Source lines of code | | | | |  |
| Components | Comments | Language | Minimum | Likely | Maximum | Expected | Comment |  |
| Expert System | New-There are risk in the estimates | Python | 20000 | 25000 | 40000 | 26667 | Minimum-As per 02 mrtvsystemdesignsummary(1)  document  Maximum- Theo suggested to double the initial estimate of 20000  Likely- Mean of max and minimum. | 50000000 |
| Bit Twiddler | Modified- already implemented in assembly language by Harry Hacker | C | 1364 | 2333 | 2500 | 2199.33 | Minimum- For C language, 5.5 bytes of memory per line is needed and as per system design summary, it takes 7500 bytes of memory. 7500/5.5=1364.  Maximum- Estimation has been taken from Results of Initial Size Estimate Preparation Meeting as per Harry Hacker discussion.  Likely- As per Harry Hacker, it takes 3-2 ratio of assembly lines to ‘C’ lines in previous efforts. 3500 lines of assembly code is converted into C code. | 7500 |
| Operating System | New/Modified/Reuse- There is already an existing OS having some bugs and therefore it needs some minimal modification | C | 49500 | 101250 | 153000 | 101250 | Minimum-New code+0.6 times the modified code0.3%+0.3 times the reuse component  Maximum- Since the operating system has some bugs assuming the worst case it will be the total of modified, reused and new.  Likely- Mean of max and minimum. | 64000000 |
| Total for software item |  |  | 70864 | 128583 | 195500 | 130116 |  | 114007500 |
|  |  |  |  |  |  |  |  |  |

## 2.2.3 Equivalent Size Table

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOFTWARE ITEM :** | **WBS & SOW #s:** | **NEW** | **COTS/REUSED CODE** | | | **MODIFIED CODE** | | | **NET** |
| **CODE** | **SLOC or Function Pts.** | | | **SLOC or Function Pts.** | | | **EQUIV.** |
| **(SLOC or** |  | | |  | | | **SIZE** |
| **PROCESSOR #2** | **SOW** | **Function** |  |  |  |  |  |  |  |
| **WBS** | **Points)** | **ORIGINAL** | **EQUIV.** | **EQUIV.** | **ORIGINAL** | **EQUIV.** | **EQUIV.** |  |
| ***COMPONENTS*** | ***LANGUAGE*** |  |  | ***RATIO*** |  |  | ***RATIO*** |  |  |
| Expert System | Python | 21667 | - | - | - | 5000 | 0.6 | 3000 | 24667 |
| Bit Twiddler | C | - | 2199.33 | 0.3 | 659.799 | - | 0.6 | - | 659.799 |
| Operating System | C | - | 101250 | 0.02 | 2025 | - | - | - | 2025 |
| Total for Python | Python | 21667 | - |  | 0 | 5000 | 0.6 | 3000 | 24667 |
| Total for C | C | - | 103449 | 0.02 | 2684 | - | - | - | 2684 |
| Total for Java | JAVA | 0 | 0 | 0 | 0 | 0 | - | 0 | 0 |
| Total |  | 21667 | 103449 | 0.02 | 2684 | 5000 | 0.6 | 3000 | 27351 |

## 2.2.4 Bar Chart showing new, reused and modified

# 

# 2.3 Software Item #3

## 2.3.1 Assumptions and other notes, such as rationale for reuse factors

Assumed the same size estimates of Software Item#1.

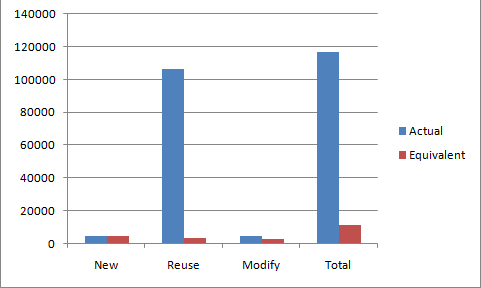
## 2.3.2 Actual Size Table

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Software Item:  Processor#3 | Components or Description | WBS &SOW #S: | Actual Size | | | | | Bytes of Target memory |
| Source lines of code | | | | |  |
| Components | Comments | Language | Minimum | Likely | Maximum | Expected | Comment |  |
| Frame Putter | Modified- Already implemented in BASIC and needs to be modified in C | C | 8000 | 10000 | 12727 | 10121.16 | Minimum-5000 (new)+60% \*5000(m0dified)+ 0%\*5000(reused).Reused is free because ” Dee Zign” rewrote that part for another project. The above estimation has been taken from Results of Initial Size Estimate Preparation Meeting  Document.  Maximum- For C language, 5.5 bytes of memory per line is needed and as per system design summary, it takes 70000 bytes of memory. 70000/5.5=12,727  Likely- From the System design summary, sit is believed to be about 10000 LOC | 70000 |
| Bit Twiddler | Reuse – from processor 1 | C | 1364 | 2333 | 2500 | 2199.33 | Minimum- For C language, 5.5 bytes of memory per line is needed and as per system design summary, it takes 7500 bytes of memory. 7500/5.5=1364.  Maximum- Estimation has been taken from Results of Initial Size Estimate Preparation Meeting as per Harry Hacker discussion.  Likely- As per Harry Hacker, it takes 3-2 ratio of assembly lines to ‘C’ lines in previous efforts. 3500 lines of assembly code is converted into C code. | 7500 |
| Miscellaneous Stuff | Reuse-From processor 1 | C | 2000 | 3000 | 4000 | 3000 | Minimum-System design summary, Estimated as 2000 LOC.  Maximum- Result of initial Size estimate preparation meeting. Harry estimates to be 4000.  Likely- Mean of max and minimum. | 14000 |
| Operating System |  | C | 49500 | 101250 | 153000 | 101250 | Minimum-New code+0.6 times the modified code0.3%+0.3 times the reuse component  Maximum- Since the operating system has some bugs assuming the worst case it will be the total of modified, reused and new.  Likely- Mean of max and minimum. | 64000000 |
| Total for software item |  |  | 60864 | 116583 | 172227 | 116570 |  | 64091500 |
|  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOFTWARE ITEM :** | **WBS & SOW #s:** | **NEW** | **COTS/REUSED CODE** | | | **MODIFIED CODE** | | | **NET** |
| **CODE** | **SLOC or Function Pts.** | | | **SLOC or Function Pts.** | | | **EQUIV.** |
| **(SLOC or** |  | | |  | | | **SIZE** |
| **PROCESSOR #3** | **SOW** | **Function** |  |  |  |  |  |  |  |
| **WBS** | **Points)** | **ORIGINAL** | **EQUIV.** | **EQUIV.** | **ORIGINAL** | **EQUIV.** | **EQUIV.** |  |
| ***COMPONENTS*** | ***LANGUAGE*** |  |  | ***RATIO*** |  |  | ***RATIO*** |  |  |
| Frame Putter | C | 5000 | 121 | 0.3 | 36.3 | 5000 | 0.6 | 3000 | 8036.3 |
| Bit Twiddler | C | - | 2199 | 0.3 | 660 | - | - | - | 660 |
| Miscellaneous Stuff | C | - | 3000 | 0.3 | 900 | - | - | - | 900 |
| Operating System | C | - | 101250 | 0.02 | 2025 | - | - | - | 2025 |
| Total for Python | Python | 0 | - |  | 0 | 0 | - | 0 | 0 |
| Total for C | C | 5000 | 106570 | 0.03 | 3621.3 | 5000 | - | 3000 | 11621 |
| Total for Java | JAVA | 0 | - | 0 | 0 | 0 | - | 0 | 0 |
| Total |  | 5000 | 106570 | 0.3 | 3621.3 | 5000 | 0.6 | 3000 | 11621 |

## 2.3.3 Equivalent Size Table

## 2.3.4 Bar Chart showing new, reused and modified



# 2.4 Software Item #4

## 2.4.1 Assumptions and other notes, such as rationale for reuse factors

Considered the proportional ratio for Sound capacity part to be 0.02 as the effort is just to actualize the product in the software which is being conveyed by FANCYMIKES.

## 2.4.2 Actual Size Table

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Software Item:  Processor#4 | Components or Description | WBS &SOW #S: | Actual Size | | | | | Bytes of Target memory |
| Source lines of code | | | | |  |
| Components | Comments | Language | Minimum | Likely | Maximum | Expected | Comment |  |
| Sound Functions | COTS- No significant change is needed except for the synchronization | Java | 2542 | 3478 | 5000 | 3576 | Minimum- As per system design summary.  Maximum-the initial size meeting estimate says that it would take 5000 LOC.  Likely- As per system design summary, it takes 40000 bytes of memory and written in JAVA. It takes 11.5 bytes per line and calculations give 3478. | 40000 |
| Operating System |  | C | 49500 | 101250 | 153000 | 101250 | Minimum-New code+0.6 times the modified code0.3%+0.3 times the reuse component  Maximum- Since the operating system has some bugs assuming the worst case it will be the total of modified, reused and new.  Likely- Mean of max and minimum. | 64000000 |
| Total for software item |  |  | 52042 | 104728 | 158000 | 104826 |  | 64040000 |
|  |  |  |  |  |  |  |  |  |

## 

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SOFTWARE ITEM :** | **WBS & SOW #s:** | **NEW** | **COTS/REUSED CODE** | | | **MODIFIED CODE** | | | **NET** |
| **CODE** | **SLOC or Function Pts.** | | | **SLOC or Function Pts.** | | | **EQUIV.** |
| **(SLOC or** |  | | |  | | | **SIZE** |
| **PROCESSOR #4** | **SOW** | **Function** |  |  |  |  |  |  |  |
| **WBS** | **Points)** | **ORIGINAL** | **EQUIV.** | **EQUIV.** | **ORIGINAL** | **EQUIV.** | **EQUIV.** |  |
| ***COMPONENTS*** | ***LANGUAGE*** |  |  | ***RATIO*** |  |  | ***RATIO*** |  |  |
| Sound Functions | JAVA | - | 3576 | 0.02 | 71 | - | - | - | 71 |
| Operating System | C | - | 101250 | 0.02 | 2025 | - | - | - | 2025 |
| Total for Python | Python | 0 | - |  | 0 | 0 | - | 0 | 0 |
| Total for C | C | 0 | 101250 | 0.02 | 2025 | 0 | - | 0 | 2025 |
| Total for Java | JAVA | 0 | 3576 | 0.01 | 71 | 0 | - | 0 | 71 |
| Total |  | 0 | 104826 | 0.02 | 2096 | 0 | - | 0 | 2096 |

## 2.4.3 Equivalent Size Table

## 2.2.4 Bar Chart showing new, reused and modified

