Hotwire PE

WKID Cost Estimation

Document number:
Version: 1.00
Status: Final
CONTENTS

1. Introduction ........................................................................................................... 3
2. References ............................................................................................................. 3
3. Cost Evaluation Technique ................................................................................... 5
4. Assumptions ......................................................................................................... 7
5. WKID software Languages and SLOCs ............................................................... 7
6. WKID software cost assessment ....................................................................... 7
7. Conclusion ............................................................................................................ 10
1. Introduction

Hotwire PE has purchased the WKID package as part of the business strategic direction with the objective of establishing a worldwide BitCoin banking system. It is fundamental to the Roadmap and Vision developed by Hotwire for the next 5 years and is part of the overall vision to provide technologies that drive innovation in the banking and finance industries.

The WKID software suite is comparable to complete banking software currently in use by major banks in Australia and worldwide. The systems are typically large and complex. The development cycle to develop and implement a banking software system similar to the WKID is 10 to 15 years. The development cycle is complex and risk exposure is extremely high. The development requires strong management of a very large highly qualified development team.

Purchasing the WKID software is a mitigating of the extreme high risk exposure associated with banking software and will allow the Hotwire business plan to be executed within the next 5 years. No existing banking software providers are able to provide a banking system based on Bitcoin principles and framework. Hotwire will be able to develop an internationally recognised Bitcoin banking system based on the WKID software.
## 2. References

<table>
<thead>
<tr>
<th>No.</th>
<th>Document Title</th>
<th>Author</th>
<th>Date</th>
</tr>
</thead>
</table>
| 1.  | COCOMO II – Constructive Cost Model  
http://csse.usc.edu/tools/COCOMOII.php |        |          |
| 2.  | COCOMO – Wikipedia  
http://en.wikipedia.org/wiki/COCOMO |        |          |
| 3.  | COCOMO II - Scenario 1 - Development of all software from beginning. | AP     | 2 Apr 14 |
| 4.  | COCOMO II - Scenario 2 - Development of one third of the software form the beginning | AP     | 2 Apr 14 |
| 5.  | COCOMO II - Scenario 3 - Detailed assessment of software being developed from the beginning - Ansi C  
COCOMO II - Scenario 3 - Detailed assessment of software being developed from the beginning – asm  
COCOMO II - Scenario 3 - Detailed assessment of software being developed from the beginning – awk  
COCOMO II - Scenario 3 - Detailed assessment of software being developed from the beginning - C#  
COCOMO II - Scenario 3 - Detailed assessment of software being developed from the beginning - C++  
COCOMO II - Scenario 3 - Detailed assessment of software being developed from the beginning – fortran  
COCOMO II - Scenario 3 - Detailed assessment of software being developed from the beginning – Java  
COCOMO II - Scenario 3 - Detailed assessment of software being developed from the beginning – lex  
COCOMO II - Scenario 3 - Detailed assessment of software being developed from the beginning – Perl  
COCOMO II - Scenario 3 - Detailed assessment of software being developed from the beginning – Python  
COCOMO II - Scenario 3 - Detailed assessment of software being developed from the beginning – sed  
COCOMO II - Scenario 3 - Detailed assessment of software being developed from the beginning – tcl  
COCOMO II - Scenario 3 - Detailed assessment of software being developed from the beginning – yacc | AP     | 2 Apr 14 |
3. Cost Evaluation Technique

The technique used to evaluate the cost of the WKID software pack has been based on the COCOMO II (Constructive Cost Model) software estimation tool (Ref [1] & [2]).

The COCOMO II is an internationally recognized estimation tool and the model has successfully been used since 1990 by established software companies and entities. The model is well tested and provides about 20% cost estimate accuracy (Ref. Code project http://www.codeproject.com/).

The COCOMO II estimation model covers all phases of the software development life time cycle:

- Inception;
- Elaboration;
- Construction; and
- Transition.

The COCOMO II estimation model is also providing a distribution of effort for:

- Management
- Environment/CM
- Requirements
- Design
- Implementation
- Design
- Implementation
- Assessment/verification; and
- Deployment

The evaluation has been based on industry standard for SLOC (software lines of codes) and derived from the WKID software pack at software component language level.

In his seminal work on coding, Brookes (1995) specifies "10 lines per developer per day" as the tested, debugged output from a standard programmer over the project life cycle ("The Mythical Man-Month, Essays on Software Engineering"). The greater the output, the lower and more conservative the valuation. For the sake of comparison, this assessment will also provide a highly theoretical evaluation for 50 lines per developer per day. This is 5 times that proposed by Brookes and is likely not possible except for software development by the top 1% of software developers using the latest and most modern development tools on the market.

THE COCOMO II is using SLOC as matrix for the cost calculation.
 WKID Cost Assessment

The WKID software cost evaluation has been based on the following rules:

1. Cost estimation model: COCOMO II
2. SLOC established at software language level
3. Software analysis and complexity has been established at software language level
4. Overall cost has been calibrated in-line with 10 years development lifecycle (overall cost reduced by 27% - assuming 2.8% pay rise per annum).
5. Monthly Engineering rates are based on existing rate data from personnel hired by Hotwire PE at 2013/2014 market rates ($9,558/Monthly).
6. Software language assessment technique is based on an extremely conservative approach, assessed as standard complexity (defined as Nominal) and not taking into account security aspects associated with banking software. The following complexity levels have been applied:
   - Driver/Conditional Complexity level 1 (see below)
   - Driver/Conditional Complexity level 2 (see below)
   - Driver/Conditional Complexity level 3 (see below)
7. Standard software engineering profile (10 lines of code/day) over the life time of the project.
8. No software maintenance provided by seller.

Driver/Conditional Complexity level 1
Required Software Reliability                      Very High
Data Base Size                                      Very High
Product Complexity                                  Extra High
Documentation Match to Lifecycle Needs             High
Analyst Capability                                  High
All Others                                          Nominal

Driver/Conditional Complexity level 2
Required Software Reliability                      Very High
Product Complexity                                  Extra High
Documentation Match to Lifecycle Needs             High
Analyst Capability                                  High
All Others                                          Nominal

Driver/Conditional Complexity level 3
All                                                  Nominal
4. Assumptions

The WKID software purchase is a one-off license between seller and HotwirePE.

5. WKID software Languages and SLOCs

<table>
<thead>
<tr>
<th>Language</th>
<th>SLOCs</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>C#</td>
<td>2,321,655</td>
<td>38.18%</td>
</tr>
<tr>
<td>C++</td>
<td>1,454,400</td>
<td>23.92%</td>
</tr>
<tr>
<td>java</td>
<td>851,123</td>
<td>14.00%</td>
</tr>
<tr>
<td>Perl</td>
<td>754,958</td>
<td>12.42%</td>
</tr>
<tr>
<td>python</td>
<td>355,220</td>
<td>5.84%</td>
</tr>
<tr>
<td>asm</td>
<td>158,002</td>
<td>2.60%</td>
</tr>
<tr>
<td>Ansi C</td>
<td>156,564</td>
<td>2.57%</td>
</tr>
<tr>
<td>awk</td>
<td>11,452</td>
<td>0.19%</td>
</tr>
<tr>
<td>lex</td>
<td>9,521</td>
<td>0.16%</td>
</tr>
<tr>
<td>tcl</td>
<td>3,017</td>
<td>0.05%</td>
</tr>
<tr>
<td>yacc</td>
<td>2,015</td>
<td>0.03%</td>
</tr>
<tr>
<td>sed</td>
<td>1,526</td>
<td>0.03%</td>
</tr>
<tr>
<td>fortran</td>
<td>1,102</td>
<td>0.02%</td>
</tr>
</tbody>
</table>

Total SLOCs: 6,080,555

6. WKID software cost assessment

The following three cost scenarios have been established in order to evaluate the WKID software cost:

- Scenario 1: Total WKID software is new development.
- Scenario 2: WKID software is developed as 33% new, 33% reuse and 33% modified.
- Scenario 3: WKID software is developed with new/reuse/modified parameters estimated for each language.
All scenarios are costed in-line with evaluation rules as established in above (Section 3).

6.1 Scenario 1

This scenario is based on total WKID software is new development (Ref. COCOMO II Calculation [3]).

<table>
<thead>
<tr>
<th>language</th>
<th>SLOC</th>
<th>New</th>
<th>Reuse</th>
<th>Modified</th>
<th>Maintenance</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>All WKID SW</td>
<td>6,080,555</td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
<td>NO</td>
<td>Complexity 1</td>
</tr>
</tbody>
</table>

Effort (man-Months): 112829
Cost (using 2013/2014 rates): $1,078,421,119
Cost (reduced 27%): $787,247,417

6.2 Scenario 2

This scenario is based on WKID software is developed as 33% new, 33% reuse and 33% modified (Ref. COCOMO II Calculation [4]).

<table>
<thead>
<tr>
<th>language</th>
<th>SLOC</th>
<th>New</th>
<th>Reuse</th>
<th>Modified</th>
<th>Maintenance</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>All WKID SW</td>
<td>6,080,555</td>
<td>33%</td>
<td>33%</td>
<td>33%</td>
<td>NO</td>
<td>Complexity 1</td>
</tr>
</tbody>
</table>

Effort (man-Months): 33708
Cost (using 2013/2014 rates): $322,179,559
Cost (reduced 27%): $235,191,078
6.3 Scenario 3

This scenario is based on WKID software WKID software complexity, development share and cost are assessed and evaluated against each software components level (Ref. COCOMO II Calculation [5]).

<table>
<thead>
<tr>
<th>language</th>
<th>SLOC</th>
<th>New</th>
<th>Reuse</th>
<th>Modified</th>
<th>Maintenance</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>C#</td>
<td>2,321,655</td>
<td>50%</td>
<td>25%</td>
<td>25%</td>
<td>NO</td>
<td>Complexity 1</td>
</tr>
<tr>
<td>C++</td>
<td>1,454,400</td>
<td>50%</td>
<td>25%</td>
<td>25%</td>
<td>NO</td>
<td>Complexity 1</td>
</tr>
<tr>
<td>java</td>
<td>851,123</td>
<td>50%</td>
<td>25%</td>
<td>25%</td>
<td>NO</td>
<td>Complexity 1</td>
</tr>
<tr>
<td>Perl</td>
<td>754,958</td>
<td>25%</td>
<td>50%</td>
<td>25%</td>
<td>NO</td>
<td>Complexity 3</td>
</tr>
<tr>
<td>python</td>
<td>355,220</td>
<td>50%</td>
<td>25%</td>
<td>25%</td>
<td>NO</td>
<td>Complexity 1</td>
</tr>
<tr>
<td>asm</td>
<td>158,002</td>
<td>25%</td>
<td>50%</td>
<td>25%</td>
<td>NO</td>
<td>Complexity 2</td>
</tr>
<tr>
<td>Ansi C</td>
<td>156,564</td>
<td>50%</td>
<td>25%</td>
<td>25%</td>
<td>NO</td>
<td>Complexity 2</td>
</tr>
<tr>
<td>awk</td>
<td>11,452</td>
<td>25%</td>
<td>50%</td>
<td>25%</td>
<td>NO</td>
<td>Complexity 3</td>
</tr>
<tr>
<td>lex</td>
<td>9,521</td>
<td>25%</td>
<td>50%</td>
<td>25%</td>
<td>NO</td>
<td>Complexity 3</td>
</tr>
<tr>
<td>tcl</td>
<td>3,017</td>
<td>25%</td>
<td>50%</td>
<td>25%</td>
<td>NO</td>
<td>Complexity 3</td>
</tr>
<tr>
<td>yacc</td>
<td>2,015</td>
<td>25%</td>
<td>50%</td>
<td>25%</td>
<td>NO</td>
<td>Complexity 3</td>
</tr>
<tr>
<td>sed</td>
<td>1,526</td>
<td>25%</td>
<td>50%</td>
<td>25%</td>
<td>NO</td>
<td>Complexity 3</td>
</tr>
<tr>
<td>fortran</td>
<td>1,102</td>
<td>25%</td>
<td>50%</td>
<td>25%</td>
<td>NO</td>
<td>Complexity 3</td>
</tr>
</tbody>
</table>

Estimated New Development (SLOCs); 2,804,879

Effort (man-Months): 39595

Cost (using 2013/2014 rates): $378,450,135

Cost (reduced 27%): $276,268,599
7. Conclusion

An extremely conservative cost evaluation (Ref Scenario 3) using industrial standards is assessing minimum cost for the WKID SW ($AUD): $276,268,599.

WKID Software at sell assuming 10% margin ($AUD): $303,895,458.

Increasing the efficiency factor 5 times (from 10 to 50 Software Lines of Code per day) will reduce the WKID Software cost to around $55Mill.

The purchase price of $55Mill is considered extremely low for a product of this size and complexity.