MOLACH: The Model for Language Change

Dan Strickland
Dynetics, Inc.
daniel.strickland@dynetics.com
Background

Contractor requests from demonstration for changes in and Validation phase software language to Engineering and Manufacturing Development phase.

MOIACT was developed to provide quantified justification for relative effort in software language change.

Base Model and Language Tax

Example: Ada95: 49 SLOC per Function Point
C++: 53 SLOC per Function Point
Language Tax for a C++ application originally planned in Ada95: (53/49)^1.08 = 8% more code
Language Environmental Factors

- Language specific variables that impact software development throughout the entire life cycle of a project
- Based on COCOMO II Effort Adjustments
- Uses definitions from the COCOMO II Languages and Environment Factors (LTEX) and Use of Software Tools (USST) data sets
- The following new variables are used:
  - Support (SVR)
  - Development (DEV)
  - Environment (ENV)

The adjustment set of over 700 variables was used to adjust the effort estimates for environmental factors. The effort adjustments associated with the environmental factors (LTEX, TOOL, SVR, DEV, ENV) were calculated based on feedback from over 30 software engineers.
Dynetics, Inc.  UNCLASSIFIED

Language Environmental Factors (LTEX)

This is a measure of the level of programming language and software tool experience of the project team developing the software system or subsystem. Software development includes the use of tools that include requirements and design representation and analysis, configuration management, document extraction, library management, program style and formatting, consistency checking, etc. In addition to experience in programming with a specific language the supporting tool set is also considered development time. A low rating is given for experience of less than 2 projects. A rating of 1 is given for experience of 0 or more years.

<table>
<thead>
<tr>
<th>Experience</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rating</td>
<td>1.25</td>
<td>1.55</td>
<td>1.85</td>
<td>2.07</td>
<td>2.14</td>
</tr>
</tbody>
</table>

UNCLASSIFIED

Dynetics, Inc.  UNCLASSIFIED

Language Environmental Factors (TOOL)

These tools have improved significantly since the 1970's projects used to calibrate COCOMO. The tool rating ranges from simple edit and code, very low, to integrated lifecycle management tools, very high.

<table>
<thead>
<tr>
<th>Experience</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rating</td>
<td>2.27</td>
<td>2.89</td>
<td>3.06</td>
<td>3.76</td>
<td>3.92</td>
</tr>
</tbody>
</table>

UNCLASSIFIED
### Language Environmental Factors (VEUF)

Support for software is a present and future concern for number of vendors, number of support tools, availability of support tools and an established corps of support programmers of a given language. The vendor support rating ranges from an internal language with no outside support to widespread proliferation of support vendors and extensive sets of development tools and suites dedicated to a specific language.

<table>
<thead>
<tr>
<th>Language Factor</th>
<th>1.00</th>
<th>1.50</th>
<th>2.00</th>
<th>2.50</th>
<th>3.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vendor Support</td>
<td>1.00</td>
<td>1.50</td>
<td>2.00</td>
<td>2.50</td>
<td>3.00</td>
</tr>
<tr>
<td>Tool Availability</td>
<td>1.00</td>
<td>1.50</td>
<td>2.00</td>
<td>2.50</td>
<td>3.00</td>
</tr>
<tr>
<td>Support Personnel</td>
<td>1.00</td>
<td>1.50</td>
<td>2.00</td>
<td>2.50</td>
<td>3.00</td>
</tr>
</tbody>
</table>

### Language Environmental Factors (DEVF)

Vendor's impact on predicted effort in development. If a language can not be supported by vendors, programmers, or taught in the academic world, the user has to maintain and develop tool sets.

<table>
<thead>
<tr>
<th>Language Factor</th>
<th>1.27</th>
<th>1.32</th>
<th>1.60</th>
<th>2.56</th>
<th>0.74</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vendor Support</td>
<td>1.00</td>
<td>1.50</td>
<td>2.00</td>
<td>2.50</td>
<td>3.00</td>
</tr>
<tr>
<td>Tool Availability</td>
<td>1.00</td>
<td>1.50</td>
<td>2.00</td>
<td>2.50</td>
<td>3.00</td>
</tr>
<tr>
<td>Support Personnel</td>
<td>1.00</td>
<td>1.50</td>
<td>2.00</td>
<td>2.50</td>
<td>3.00</td>
</tr>
</tbody>
</table>
Language Environmental Factors (PORT)

If there is no portability for a language, tools and support costs will be difficult to keep under control. Code developed for one version of a language might not adhere to the rules developed for tools supporting a different version. If there is a high degree of portability, developed source (and sometimes executable) code can be taken from one platform or tool to another without much difference in development.

Complete MOLACH Definition

```
Language SLOG = (NEW SLOG) + (Reuse) + (Language Text) + (LEFT)

```

UNCLASSIFIED
Example 1: Advantage Target

Situation: A program has developed 10000 lines of Ada95 code and has estimated another 10000 lines of new code are needed in the next phase. Design Modification = 25%, Code Modification = 75%, Integration and Retest Required = 80%. Is it cost effective over the life of the program to change language to C++?

\[
\text{LSLOC} = (\text{New Code} + \text{Reuse}) \times \text{Language Tax} \times \text{LEF}
\]

Example 2: Advantage Status Quo

Situation: A program has developed 20000 lines of Ada95 code and has estimated another 10000 lines of new code are needed in the next phase. Design Modification = 25%, Code Modification = 25%, Integration and Retest Required = 50%. Is it cost effective over the life of the program to change language to C++?

\[
\text{LSLOC} = (\text{New Code} + \text{Reuse}) \times \text{Language Tax} \times \text{LEF}
\]
**Actual Results**

All values are in Language SLOC

<table>
<thead>
<tr>
<th>SL</th>
<th>Ada Effort</th>
<th>C Effort</th>
<th>%Delta</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.2%</td>
<td>1.2%</td>
<td>-0.0%</td>
</tr>
<tr>
<td>2</td>
<td>3.3%</td>
<td>3.3%</td>
<td>0.0%</td>
</tr>
<tr>
<td>3</td>
<td>4.0%</td>
<td>4.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>4</td>
<td>1.9%</td>
<td>1.9%</td>
<td>0.0%</td>
</tr>
<tr>
<td>5</td>
<td>7.3%</td>
<td>7.3%</td>
<td>0.0%</td>
</tr>
</tbody>
</table>

* C++ development is a viable alternative for some SL's

**Future Work and Conclusions**

- Future work:
  - Refine model and algorithms
  - Gather feedback on test results
  - Apply model to additional SL's
  - Incorporate user feedback

- Considerations:
  - Measure relative effort involved with a software language change
  - Offers software developers a method of justifying requests for language changes
  - Gives software customers a tool to measure potential average effort.