OMO: Software cost estimation

Tim Menzies

Lane Department of Computer Science, West Virginia University, PO Box 6109, Morgantown, WV, 26506-6109, USA;

http://tim.menzies.us; tim@menzies.us


Abstract  COCOMO is a software effort estimation tool. OMO is COCOMO written in SWI-Prolog [5]

1  Warning!!!

Contains deliberate error. Output should be:

COCOMO.ga says 1223.53 months (total);
28 staff over 45 months

But show how that is all broken.

COCOMO.ga says 134588.0 months (total);
547 staff over 247 months

What is wrong?

2  What is COCOMO?

The COCOMO project aims at developing an open-source, public-domain software effort estimation model. The project has collected information on 161 projects from commercial, aerospace, government, and non-profit organizations [1, 4]. As of 1998, the projects represented in the database were of size 20 to 2000 KSLOC (thousands of lines of code) and took between 100 to 10000 person months to build.

COCOMO measures effort in calendar months where one month is 152 hours (and includes development and management hours). The core intuition behind COCOMO-based estimation is that as systems grow in size, the effort required to create them grows exponentially, i.e. \( ef(fort) \propto KSLOC^a \). More precisely:

\[
\text{months} = a * \left( KSLOC^{(0.91 + \sum_{i=1}^{n} SF_i)} \right) \times \left( \prod_{j=1}^{17} EM_j \right)
\]

where \( a \) is a domain-specific parameter, and KSLOC is estimated directly or computed from a function point analysis. \( SF_i \) are the scale factors (e.g. factors such as “have we built this kind of system before?”) and \( EM_j \) are the cost drivers (e.g. required level of reliability). Figure 1 lists the scale drivers and effort multipliers.

Software effort-estimation models like COCOMO-II should be tuned to their local domain. Off-the-shelf “untuned” models have been up to 600% inaccurate in their estimates, e.g. [3, p165] and [2]. However, tuned models can be far more accurate. For example, [1] reports a study with a bayesian
<table>
<thead>
<tr>
<th>Type</th>
<th>Acronym</th>
<th>Definition</th>
<th>Low-end</th>
<th>Medium</th>
<th>High-end</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM</td>
<td>acap</td>
<td>analyst capability</td>
<td>worst 15%</td>
<td>55%</td>
<td>best 10%</td>
</tr>
<tr>
<td>EM</td>
<td>exp</td>
<td>applications experience</td>
<td>2 months</td>
<td>1 year</td>
<td>6 years</td>
</tr>
<tr>
<td>SF</td>
<td>arch</td>
<td>architecture or risk resolution</td>
<td>few interfaces defined or few risk eliminated</td>
<td>most interfaces defined or most risks eliminated</td>
<td>all interfaces defined or all risks eliminated</td>
</tr>
<tr>
<td>EM</td>
<td>cplx</td>
<td>product complexity</td>
<td>e.g. simple read/write statements</td>
<td>e.g. use of simple interface widgets</td>
<td>e.g. performance-critical embedded systems</td>
</tr>
<tr>
<td>EM</td>
<td>data</td>
<td>database size</td>
<td>10</td>
<td>100</td>
<td>1000</td>
</tr>
<tr>
<td>EM</td>
<td>docu</td>
<td>documentation</td>
<td>many life-cycle phases not documented</td>
<td>extensive reporting for each life-cycle phase</td>
<td></td>
</tr>
<tr>
<td>SF</td>
<td>flex</td>
<td>development flexibility</td>
<td>development process rigorously defined</td>
<td>some guidelines, which can be relaxed</td>
<td>only general goals defined</td>
</tr>
<tr>
<td>EM</td>
<td>ltx</td>
<td>language and tool-set experience</td>
<td>2 months</td>
<td>1 year</td>
<td>6 years</td>
</tr>
<tr>
<td>EM</td>
<td>pcap</td>
<td>programmer capability</td>
<td>worst 15%</td>
<td>55%</td>
<td>best 10%</td>
</tr>
<tr>
<td>EM</td>
<td>pcon</td>
<td>personnel continuity</td>
<td>48%</td>
<td>12%</td>
<td>3%</td>
</tr>
<tr>
<td>EM</td>
<td>pexp</td>
<td>platform experience</td>
<td>2 months</td>
<td>1 year</td>
<td>6 years</td>
</tr>
<tr>
<td>SF</td>
<td>prec</td>
<td>precededness</td>
<td>we have never built this kind of software before</td>
<td>somewhat new</td>
<td>thoroughly familiar</td>
</tr>
<tr>
<td>EM</td>
<td>pvol</td>
<td>platform volatility</td>
<td>12 months</td>
<td>6 months</td>
<td>2 weeks</td>
</tr>
<tr>
<td>EM</td>
<td>rely</td>
<td>required reliability</td>
<td>errors mean slight inconvenience</td>
<td>errors are easily recoverable</td>
<td>errors can risk human life</td>
</tr>
<tr>
<td>EM</td>
<td>ruse</td>
<td>required reuse</td>
<td>none</td>
<td>across program</td>
<td>across multiple product lines</td>
</tr>
<tr>
<td>EM</td>
<td>sced</td>
<td>dictated development schedule</td>
<td>deadlines moved closer to 75% of the original estimate</td>
<td>no change</td>
<td>deadlines moved back to 160% of the original estimate</td>
</tr>
<tr>
<td>EM</td>
<td>site</td>
<td>multi-site development</td>
<td>some contact: phone, mail</td>
<td>some email</td>
<td>interactive multi-media</td>
</tr>
<tr>
<td>EM</td>
<td>stor</td>
<td>main storage constraints (% of available RAM)</td>
<td>N/A</td>
<td>50%</td>
<td>95%</td>
</tr>
<tr>
<td>SF</td>
<td>team</td>
<td>team cohesion</td>
<td>very difficult interactions</td>
<td>basically co-operative</td>
<td>seamless interactions</td>
</tr>
<tr>
<td>EM</td>
<td>time</td>
<td>execution time constraints (% of available CPU)</td>
<td>N/A</td>
<td>50%</td>
<td>95%</td>
</tr>
<tr>
<td>EM</td>
<td>tool</td>
<td>use of software tools</td>
<td>edit, code, debug</td>
<td>well intergrated with lifecycle</td>
<td></td>
</tr>
</tbody>
</table>

The intuition to be gained from Figure 2 is that some COCOMO parameters are more influential than others. Some are weakly correlated to increasing effort (column 1); some are weakly correlated to decreasing effort (column 2); and some are strongly correlated to decreasing effort (column 3). This will be useful later when we write search engines to control COCOMO. A core heuristic will be “change the influential parameters first”.

The last column of Figure 2 relate to the effort multipliers. While shown here as linear, their influence can be even greater than that since they are used up in an exponential equation.
3 Installation

4 Pre-load actions

4.1 Hooks

Fast assertions of named variables.

Influence of different COCOMO parameters

5 Main System

5.1 Main driver

5.2 Equations

5.2.1 Sizing equations

5.2.2 Schedule Equations

5.2.3 Effort Equations

5.3 Tunings

5.3.1 Constants
5.3.2 Post-architecture scale factors  The COCOMO 2000 scale factors learnt via bayesian tuning.

Some effort multipliers learnt via some genetic algorithms.

5.4.1 General

Some effort multipliers learnt via some genetic algorithms.

5.4 Data dictionary

5.4.2 "project"
5.4.3 "scores"

5.4.3 "scores"

5.4.3 "scores"

5.4.3 "scores"

5.4.3 "scores"

5.4.3 "scores"

5.4.3 "scores"

5.4.3 "scores"

5.4.3 "scores"

5.4.3 "scores"

5.4.3 "scores"

5.4.3 "scores"

5.4.3 "scores"

5.4.3 "scores"

5.4.3 "scores"

5.4.3 "scores"

5.4.3 "scores"

5.4.3 "scores"

5.4.3 "scores"

5.4.3 "scores"

5.4.3 "scores"

5.4.3 "scores"

5.4.3 "scores"

6 Start-up actions

6 Start-up actions

6.1 Multis/2

7 OMO Support code

OMO Support code

7.1 Multis/2

7.2 Fields/3

Poke some values into the named fields.

8 Knowledge base

8 Knowledge base

8 Knowledge base

8 Knowledge base

8 Knowledge base

8 Knowledge base

8 Knowledge base

8 Knowledge base
8.2 LOC per Function points

Also loaded, but not shown due to size, are tables showing productivity in different 482 different programming systems. It tables a lot of code to get anything done in binary, but less code as the language matures. So:

upf2sloc('1st generation default',320).
upf2sloc('2nd generation default',107).
upf2sloc('3rd generation default',80).
upf2sloc('4th generation default',20).
upf2sloc('5th generation default',5).

The units here are lines of code per function point. For more details, see Boehm.

9 Bugs

None known but many suspected.

Acknowledgements his research was conducted at West Virginia University under NASA contract NCC2-0979 and NCC5-685. The work was sponsored by the NASA Office of Safety and Mission Assurance under the Software Assurance Research Program led by the NASA IV&V Facility. Reference herein to specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government.

References


A License

This software is distributed under the GNU General Public License.

A.1 nowarranty.txt

OMO comes with ABSOLUTELY NO WARRANTY: for more details type 'warranty'. This is free software, and you are welcome to redistribute it under certain conditions: for more details, type 'conditions'.

A.2 warranty.txt

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; version 2 (see http://www.gnu.org/copyleft/gpl.html or type 'conditions').

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, US.

A.3 conditions.txt

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The 'Program', below, refers to any such program or work, and a 'work based on the Program' means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term 'modification'.) Each licensee is addressed as 'you'.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also do one of the following:

a) You must cause the modified copies to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, that you provide a warranty) and that users may redistribute the program under these conditions, telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to write works entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above.

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above.

In exercising the rights to copy, distribute and modify the Program, you may also link an unmodified version of it with the Very Bad Library, thus producing a work based on the Program, which is a work not covered by this License.

You must cause all the files accompanying the modified program to carry statements which are identical to this permission notice.

You may not offer or attempt to offer support services for the Program through the free software community.

You may insert a notice that is intended to serve as a legal paradigm for a notice to copyright holders.

The program library is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; version 2 of the License is available online at http://www.gnu.org/copyleft/gpl.html or type 'conditions'.

You should have received a copy of the GNU General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.
The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. As a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License. However, if you are included in a product or a distribution that contravenes the conditions of this License, the owner of the product or distribution assumes the legal risk. You must make available free of charge to the end users of the product or distribution a copy of any such restrictions and conditions.

8. If you distribute the Program, including patches or enhancements to the Program, and these modifications exclude or otherwise alter the conditions of this License, you must do so. Such a distribution is called a “modified version” of the Program.

9. You may distribute in object code or executable form the work according to the terms of Sections 2 and 3 above provided that you accompany it with the complete corresponding source code, which must be distributed under the terms of this License.

10. You may not offer to sell or sell for profit or charge a fee for the Program. The Free Software Foundation accepts no responsibility for the content of the other free programs whose distribution conditions are different, write to the author to ask for permission.

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM ‘AS IS’ WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.