A comparison of SLIM and COCOMO estimates versus historical man-power and effort allocation

A. Caccamese (*)
L. Cappello (*)
G. Dodero (*)

Ing. C. Olivetti & C. S.p.A. - Divisione Minicomputers e Reti - Ivrea (Italy)
(**) Universita' di Genova - Dipartimento di Matematica
- Genova (Italy)
1. Introduction

The development of a Planning and Cost Control System is proceeding within Olivetti, where one or more software cost estimation models will be integrated. COCOMO and SLIM have been evaluated using data related to two industrial projects, and one project of the National Research Council.

The results have been compared between themselves and with the distribution of man-power and effort resulting from the historical data.

The conclusion is that those projects have followed a man-power distribution law different from the one considered by both models.
2. General framework

The approach for each case study has been the following:

a) get from the historical data the input parameters for both SLIM and COCOMO models. This has meant collecting values for the number of source code instructions of the product, as well as investigating on the production environment characteristics;

b) for both models, use data to obtain effort and manpower distribution curves, as if an a-priori estimation was carried on. In this phase, an Olivetti proprietary implementation for SLIM (OLISLIM), running on L1 Olivetti computers, has been used, while a proprietary implementation of the University of Genoa of basic COCOMO (PECCOMO), running on DEC computers, has been used;

c) collect historical data to allow drawing of real effort and manpower curves on the same diagram used for the models. In this phase, normalisation has been executed on the time axis to allow consistency for the values given by the two models;

d) compare the curves and try a qualitative interpretation on the basis of the characteristics of the projects.
2. Case studies

The following project case studies have been analysed:

#1 software for a hard disc WREN1 unit;
#2 software for a COBOL programming environment;
#3 software for a multi-micro architecture development environment.

Notice that the above case studies all deal with system software rather than application software.
It can be observed that for such systems (although developed by different organisations) a common behaviour can be extracted, which differs somehow from that of application software systems.
This point will be addressed in detail during case studies analysis and in the conclusions.
1. **Case study #1: WREN1 disc software**

1.1. **Project description**

This case study refers to the development of software for an hard disc WREN1 unit, 27 Megabytes formatted.

The software that has been considered consists in:

a) I/O subsystem, i.e. a logical interface that provides easy access to the peripheral functionalities, encapsulating all details related to interrupts handling, command retry, and specific unit commands. This subsystem is used both by drivers belonging to different operating systems (for different computers including this kind of unit) and by diagnostics programs (see below);

b) diagnostics environment, i.e. a library of programs performing both peripheral diagnosis (defective disc tracks) and peripheral initialisation and formatting;

c) installation and copy utilities for diagnostics environment.

The drivers have not been taken into account because there is a specific driver for each operating system that is developed by an organisation different than that producing the hard disc software, with different schedules and constraints, so it was decided to consider the project restricted to the hard disc software as described before.

The total number of people that were involved in this project amounts to 17.

The duration of the project was 14.2 months.

1.1.2. **Software characteristics**

The software is mainly written in FASCAL, with a minor part written in Z8000 assempler.

It was developed on DEC/UNIX systems, using tools for down-load loading onto the target machine and interactive hexadecimual debugger for debugging.

The number of delivered source instructions is 29736.

1.1.3. **Application of SLIM**

The application of SLIM required, first of all, the knowledge of the amount of the effort spent from the start of the activities to the time of the first external lease (td in SLIM terminology).

It has been measured using historical data and the result was 52.99 man-months (one man-month has been taken as
equivalent to 17.7 man-days, according to the criteria used inside the organisation where the project has been developed).

This data was used as input parameters to SLIM, resulting in the following values for the software project:

\[
C_k: 10239
\]

This high value for the technology constant is in line with the development environment characteristics, since it is in the range of those defined as typical of projects developed using high level languages, and interactive development environments;

\[
D: 3.60
\]

This is a low value for the project difficulty, and is in line with the fact that there have not been any specific staffing or know-how problems in this project. The group has not suffered from turnover problems, and WREN1 was not the first hard disc for which software was going to be developed by the same group;

\[
\text{grad}(D): 7.27
\]

This low value for the project difficulty gradient is in line with the fact that specifications were not affected by major changes during the development, nor were schedule pressures exercised on the group;

\[
\text{PROD: 5877 DSI/man-year}
\]

This is a high value for productivity and is consistent with the values of \(C_k\) and \(D\).

As a matter of fact, in this case study, SLIM parameters are consistent between themselves and project characteristics.

Next step consisted in using the technology constant, DSI and effort up to the development time values to get the estimated effort and man-power curves, that have been plotted in tables 1.1 and 1.2 respectively.

3.1.4. Application of COCOMO

For the application of COCOMO the value of 29736 DSI was used, and the analysis of the characteristics of the group and of its relationships with the outside world suggested to use the organic mode (in this case familiar word fits very well).

In fact, organic mode refers to a skilled organisation, a reliable development environment, and to a poor amount of
effort to be spent in innovations and search for new techniques or algorithms.

Using these parameters, effort and man-power curves have been obtained, and are shown in tables 1.1 and 1.2 respectively.

3.1.5. Comparison between models and historical data

Historical data has been used to obtain real effort and man-power curves that have been plotted in tables 1.1 and 1.2 respectively. This allows to compare the results of the models to the real behaviour of the organisation.

From this comparison, the following observations are made:

**effort:** both models underestimate effort distribution up to the development time, apart from a small region where COCOMO overestimates. The reason for this is due to the fact that the project places in the early phases activities that are usually performed later as estimated by models.

From the development time on, both models suggest a growing trend of the effort, while the experience gained in the same organisation suggests a stabilisation of the effort on constant values in a period of few months. Again, the reason for this is due to the fact that the nature of this software does not involve any specific software modifications for end-user interface improvements or for performance or memory occupancy reasons; all of these problems are generally solved during the development taking place inside the organisation, and the end-user has usually no way to address his performance or memory availability issues to the disc management software, so the feedback is really very poor;

**man-power:** here COCOMO overestimates while SLIM partially overestimates and partially underestimates. In particular, SLIM underestimated applies to approximately 65% of the development time, while approaching development time causes SLIM overestimation.

SLIM estimate is found to be closer to reality than COCOMO, and the phenomenon of early and medium underestimate may be explained with the fact that in these phases of the project activities are carried on that are usually performed
after the time of first external availability, i.e., performance measurements, benchmarks and whatever else may lead to tuning and adjustment of the product.

On the other hand, this explains the phenomenon of SLIM overestimates near the development time, because activities usually performed in this interval have been, in this specific case, already executed.

Both models are completely unreliable after the time of the first external availability of the product, in what they estimate a need for man-power allocation that is not confirmed by reality.

A correlation of this fact with the above effort considerations is apparent.

The nature of this software does not require either the effort or the man-power considered by the models, because the feedback from the field is very poor.
3.2. Case study #2: COBOL programming environment software

3.2.1. Project description

The second case study refers to the development of the software for a COBOL programming environment, consisting of a compiler and an interpreter for COBOL source programs. A remarkable feature of this case study is that compiler and interpreter data were collected as if they were two totally independent projects, but both analysis of historical data and models response suggested to treat them as data referring to a unique project.

This idea was discussed with people in charge of the project and they explained that the compiler translates COBOL source programs in a meta-language that is in turn interpreted by the interpreter. So the analysis was carried considering compiler and interpreter like two components of a unique system, and the data that is presented refer to this approach.

The total number of people that were involved on this project amounts to at least 14.

The duration of the project was 21 months.

3.2.2. Software characteristics

The software is 65% in PL7/sys and 35% in PL7/asm and LIMO (this is a low-level interpretive language).

This is a clear indication for a non-homogeneous production environment with the associated problems (language interfaces, calling conventions, parameter passing), that will be taken into account below.

System architecture and design are very complex.

It was developed on an interactive system using tools for structured programming techniques.

The number of delivered source instructions is 26000.

3.2.3. Application of SLIM

Application of SLIM required first of all the knowledge of the amount of effort spent from the start-up of the activities to the time of the first external release.

It has been measured using historical data, and it amounts to 134.816 man-month (one man-month has been taken as equivalent to 20.5 man-days, according to the criteria used inside the company that developed the project).

Applying SLIM using these input parameters, the following values have been obtained for the software project:

Ek: 3940
This is a low value among those considered in SLIM for interactive, high-level-language systems. No clear explanation has been found for this fact;

D: 10.01

This low level of difficulty does not fit with the poor experience of the project team, with the large amount of turn-over that disturbed the team and with the tight constraints of memory occupation and of efficiency that were imposed on the project. No clear explanation has been found for this fact;

grad(D): 5.72

Difficulty gradient is low, so suggesting a development without high schedule constraints which was not the case. No clear explanation has been found for this fact;

PROD: 2045 DSI/man-year

Productivity value is high. No clear explanation has been found for this fact. Note that its value is the smallest in the case studies.

As a matter of fact, SLIM parameters in this case study do not fit with the real characteristics of the project.

The next step consisted in using development time and effort up to the development time values to get the estimated effort and man-power curves, that have been plotted in tables 2.1 and 2.2 respectively.

3.2.4. Application of COCOMO

For the application of COCOMO, semidetached mode has been chosen, since this seemed the more appropriate to fit project characteristics as poor experience, extended turn-over and non-homogeneous production environment.

Using this mode and 26000 DSI value, effort and man power curves have been obtained and plotted in tables 2.1 and 2.2 respectively.

3.2.5. Comparison between models and historical data

Historical data in this case study are rather poor, in the sense that average values have been collected instead of detailed values as in the other case studies. Nevertheless, plotting these average values on tables 2.1 and 2.2 respectively, an analysis has been performed of the correlation of the model results to the behaviour of the
organisation.
In particular, this is the case study where real values are available on the effort spent after the development time.

From the comparison between real data and models output, the following observations may come:

- effort: both COCOMO and SLIM underestimate up until the development time. Overestimation is common for the two models in the next 17 months after the development time.

The reason for this may be found in the fact that certain activities usually performed after the project is made available to the field have been anticipated, with the creation of beta-sites during last phases of the project, in which the product was used by a client organisation. This involves a major amount of effort in the phase that is usually devoted to development, and a minor effort in the period after development has been completed;

- man-power: notwithstanding that man-power real curve is very rough, the conclusion may be reached that both SLIM and COCOMO underestimate in the early period of the project, while overestimation is common near and after the development time. Also in this case it seems that the organisation allocated more resources in early phases of the project than model previsions, this due to activities that are anticipated on the schedules as estimated by the models. From this point of view, this case study gives a common understanding from the comparisons of both the two set of curves.

3.3. Case study #3: Multi-Micro programming software

3.3.1. Project description

This case study refers to the development of software for a complete development environment designed for multi-micro computer architectures. The software that has been considered consists in:

a) programming subsystem, i.e. a compiler, an interpreter and code generators for different target machines.

In this subsystem functionalities related to consistency checks of source programs (string handlers, program declaration analysers, modules interface checkers) have also been considered;
b) network related functionalities, i.e. target network description analyser, distributed programs linker;

c) other functionalities, like a relational database, plus general utilities programs.

The main feature of this project is that it was a distributed project, i.e. it was carried out in four different geographic sites (two in Milan, one in Genoa and one in Bologna).

The financial support for this project came from the contributions of Universities, Government and private Companies, thus contributing to a general instability of people working on it and to a great amount of turn-over.

Moreover, some of the people involved were students preparing their thesis, while the contribution of most of the experienced people was part-time.

The total number of people that were involved on this project amounts to 16.

The duration of the project was 38.0 months.

### 3.3.2. Software characteristics

The software is 100% PASCAL.

System architecture and design are complex, and their stabilisation required a certain amount of feedback, also after the start of the development phase.

In other words, this was a project that suffered for an unstable requirement specification, besides the financial and turn-over problems already addressed.

Moreover, it being a research project, innovation has been considered a must, risky solutions have been chosen, and there was no specific know-how on the problems to be solved.

It has been developed on DEC interactive system (VMS/RSX11), using tools for structured programming techniques.

The number of delivered source instructions is 55803.

### 3.3.3. Application of SLIM

Application of SLIM required first of all the knowledge of the amount of effort spent from the start-up of the activities to the time of the first external release.

It has been counted using historical data, and it amounts to 125 man-month (one man-month has been taken as equivalent to 21 man-days, according to the media of criteria used inside the different organisations method that contributed to the project).

Applying SLIM using these input parameters, the following
values have been obtained for the software project:

\( Ck = 3882 \)

This is a low value among considered in SLIM as representative of projects developed using interactive, high-level-language.

An explanation for this fact may be found in the limitation of funds, that forced to use development environments inadequate to the people involved, so de facto jeopardising the capabilities offered by the interactive development environments;

\( D = 2.83 \)

This very low level of difficulty does not fit with the poor experience of the project team, and can find some explanation in the fact that de facto no time schedule and constraints were imposed on this project, that was mainly a research project;

\( \text{grad}(D) = 0.89 \)

The difficulty gradient is low, so it suggests a development without schedule constraints/pressures and this fits with the real situation.

On the other hand, this does not fit with the instability of the specifications of the project.

No clear explanation has been found for this fact;

\( \text{PROD} = 4672 \text{ DSI/man-year} \)

Productivity value is high.

No clear explanation has been found for this fact.

As a matter of fact, SLIM parameters for this case study do not fit the real characteristics of the project.

The next step consisted in using development time and effort up to the development time values to get the estimated effort and man-power curves. These have been plotted in tables 3.1 and 3.2 respectively.

### 3.3.4. Application of COCOMO

For the application of COCOMO, semidetached mode was chosen, since this seemed the most appropriate to fit project characteristics as composite experience (skilled people were involved as well as students), meaningful turn-over and financial difficulties that caused in some cases interrupting and resuming activities with the associated overhead for start-up.

Another factor contributing to the choice of semidetached mode came from the geographic distribution of the project
site, which caused communication problems between cooperating groups, as well as problems with alignment of development environments software releases, physical output formats, and so on.

Moreover, the decision to allow innovation in the project and the amount of technological problems faced contributed to the choice of semidetached mode.

Using this mode and 55083 DSI value, effort and man power curves have been obtained and plotted in tables 3.1 and 3.2 respectively.

3.3.5. Comparison between models and historical data

Historical data in this case study is precise, apart from a certain amount of uncertainty in evaluating the exact effort spent by people not tied to a specific company (i.e. students).

The feeling is that, these uncertainties apply to a large number of cases, the evaluation of effort is in the average good enough.

The data has been used to obtain real effort and man-power curves that have been plotted in tables 3.1 and 3.2 respectively. This allows to compare the results of the models to the real behaviour of this composite organisation.

From the comparison between real data and models output, the following observations are made:

**effort:**
here the behaviour is rather surprising.
SLIM underestimates for the whole period up until the development time, and the amount of this underestimation seems rather small.
The trend of the two curves is the same, apart from regions where a decrease in the trend is apparent, probably tied to vacations period where most of the people did not work, and less effort was applied.
After the development time SLIM underestimates, and this is common to the previous case studies.

**COCOMO,** on the contrary, apart from the start-up period, overestimates and the amount of this overestimation is large.
Also the trend of the curve is different from the real one.

It is apparent that in this case study, as far as the effort is concerned, SLIM estimations may be taken into account, while COCOMO's not.

**man-power:**
the shape of the real curve deserves in this case a bit of explanation.
The flat region at the beginning may be explained with a difficult start-up period in the project, where a lot of people were immediately allocated, whom did not have completely defined interfaces, roles and so on.

Next, there are regions where a crack in the man-power allocation is apparent. These regions correspond to the regions of decrease in the effort applied, and are tied to bureaucratic problems limiting funds allocated.

Taking into account these considerations, the shape of the man-power curve may be reduced to be regular, and similar to the one obtained in the previous case studies.

SLIM underestimates, in the period between activities start-up and near to the development time, while near the development time overestimation takes place.

This behaviour is very close to that of case study #1, thus indicating a situation in which activities are executed in an earlier stage than that considered by the model, and activities usually considered by the model after the development time are not really executed (recycle due to user interfaces modifications and similar).

On the contrary, COCOMO's overestimation is apparent and spans all over the period of the project.

4. Conclusions

The main conclusion that can be drawn from quantitative analysis is that the projects that have been studied followed an effort and man-power distribution different than those estimated by either model.

The law by which they differ from the models seems to be unique for all the case studies, in which people and effort have been allocated or spent in advance with respect to model estimates.

Standard SLIM analysis of project cost distribution between its phases suggest that effort and man-power curves may result from the cumulation of different curves, one for each main phase of the project, placed in subsequent time intervals.

Case study #1 suggests that the project reaches the peak for man-power allocation in a region where the peak for the development activity is reached in SLIM estimation, and this behaviour may be observed, although not as evident, in the other two case studies.
Moreover, a common characteristic for these projects is that they represent system software rather than application software.

This can be used to interpret the comparisons as a result of the kind of projects that have been analysed, in that software life cycle models are usually built on the hypothesis of application software. This implies an effort and man-power distribution different from those involved in the system software that has been analysed.

A contribution to this conclusion is also given by the observation of the complete disagreement of models after the development time, in that models estimate man-power and effort that are spent only in a minimum part inside the projects that have been analysed. This can be easily explained by the fact that, since we are dealing with system software, poor feedback from the field is usually received.

What can be concluded is that system software seems to follow a software life cycle where activities are compressed in comparison with the cycle of application software on which models are based.

Another common characteristic of the case studies is that all of them present an initial step for man-power allocation that may be explained with a minor flexibility in man-power handling in Italy as opposed to other countries. Managers are forced to allocate people in advance in order to avoid staffing problems that can arise immediately after, when the needs of the organisation may not be satisfied due to tight constraints in people finding.

A further observation may come from quantitative analysis of SLIM parameters. While D and grad(D) lie in the low range, PROD lies in the middle and high ranges, with reference to SLIM definitions. We feel that the intrinsic characteristics of the case studies are homogeneous between themselves and different from those of the projects on which SLIM has been validated, so a redefinition of ranges should be necessary in our environment. This is reinforced by further analysis on other project currently being carried on.
# TABLE OF CONTENTS

(No Reference Information)

2 1. Introduction

3 2. General framework

4 3. Case studies

5 3.1. Case study #1: WREN1 disc software

6 3.1.1. Project description

7 3.1.2. Software characteristics

8 3.1.3. Application of SLIM

9 3.1.4. Application of COCOMO

10 3.1.5. Comparison between models and historical data

9 3.2. Case study #2: COBOL programming environment software

10 3.2.1. Project description

11 3.2.2. Software characteristics

12 3.2.3. Application of SLIM

13 3.2.4. Application of COCOMO

14 3.2.5. Comparison between models and historical data

11 3.3. Case study #3: Multi-Micro programming software

12 3.3.1. Project description

13 3.3.2. Software characteristics

14 3.3.3. Application of SLIM

15 3.3.4. Application of COCOMO

16 3.3.5. Comparison between models and historical data

15 4. Conclusions
WREN 1 disc software

DSI = 29376
mode = organic
CK = 10239
COBOL programming environment software

Table 2.1

DSI=26000
mode=semidetached
Ck=3940

reality
-- smoothed COCOMO
-. SLIM
COBOL programming environment software

DSI=26000
mode=semidetached
Ck=3940

--- reality
--- smoothed COCOMO
--- SLIM

Table 2.2
Multi-Micro programming software

DSI=55083
mode=semidetached
Ck=3882

--- reality
-- smoothed COCOMO
-. SLIM

Table 3.1
Multi-Micro programming software

DSI=55083
mode=semidetached
CK=3882

--- reality
-- smoothed COCOMO
-- SLIM

Table 3.2