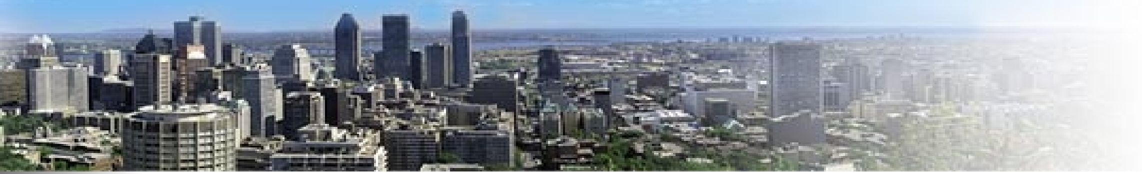


ENGINEERING

Software Estimation Models & Quality Criteria

Alain Abran & Luigi Buglione

QUALITA 2009 BESANÇON

Goals

✓ **G1.**Discuss and analyze the quality of estimates in software projects by examining the estimation models used

✓ **G2.**Evaluate such models in order to determine their reliability of use for a quality-driven process for improving estimates over time

Agenda

- **Introduction**
- **Verification of Direct Inputs**
- **Verification of Derived Inputs to Estimation Models**
- **Analysis of the Outputs of the Estimation Models**
- **Evaluation of Estimation Models**
  - ✓ Evaluation by model builders
  - ✓ Independent reviews
- **Conclusion**

Introduction
State-of-the-art

- **Software Estimation:**
  - ✓ All software projects need to be estimated
  - ✓ All estimations expected to be accurate even if based on fuzzy requirements:
    - **The search for the single magic number**!

- **Multi-variable estimation tools & models available from…**
  - ✓ Books and research papers
  - ✓ Vendors (i.e. black-box approach)
  - ✓ Web (software)

But:

Q: What is the **quality** of such estimation tools and techniques?

Introduction
State-of-the-art

- **Quality of Estimation Models**
  - ✓ Day-to-day life → the quality of products/services is a major concern
    - ❖ i.e. Consumers' Reports or specialized magazines comparing prices, characteristics, etc., before buying something
  - ✓ Work life → for software estimation models, very little is done (even if significant financial impacts will derive from such analysis)
    - ❖ i.e. the most used estimation techniques are often 'Experience & Analogy'

•Do software managers and practitioners carry out the same process for estimating a software project?

➢Why not?

•Is the software industry better at software estimation than 30 years ago?

Verification of the Direct Inputs
Main steps

- **1st step: V&V of the quality of input data**
  - ✓ The implicit assumption is that the inputs are well-defined, accurate and reliable
  - ✓ **Q**: is it true? or are we in a Garbage-in, Garbage-out situation?
- **Some examples of such V&V activities**
  - ✓ Verification of the data definitions
    - ❖ Clear definition, including scale types and related statistical techniques
  - ✓ Verification of the quality of the data collected
    - ❖ i.e. ISO/IEC 25012:2008
  - ✓ Verification of the uncertainty about the data collected
    - ❖ Complete, unambiguous, coherent, stable
    - ❖ **Evaluate the impact of uncertainty** and how to mitigate eventual risks
- **When using statistical techniques, input data need to meet conditions:**
  - ✓ A normal distribution in regression techniques
  - ✓ Identification and removal of significant outliers

Verification of Derived Inputs to Estimation Models
LOC & Functional Size…

2nd step: V&V of the quality of the derived inputs

- ✓ In literature most software estimation models take as inputs:
  - ✓ Lines of Code (LOC)
  - ✓ Function Points (FP)
  - ✓ Other derived inputs → i.e. Cost Drivers (i.e. from COCOMO or other parametric cost models)

- **LOC**
  - ✓ Typically not derived from measurement (software yet to be built) but estimated →
    - ❖ Introduction of additional uncertainty into the estimation process
    - ❖ Quality of outputs highly dependent on the quality of inputs (estimated LOCs)

- **Functional Size**
  - ✓ Functional Size largely recognized as a valid input data, but few tools
  - ✓ 'Backfiring' practice: will 'backfire'!
    - ❖ No support from a statistical viewpoint (i.e. unknown info on original data and the way they were treated to derive such conversion rates)
    - ❖ → little valuable added value for decision-making purposes

Verification of Derived Inputs to Estimation Models
…and other derived cost inputs

- **Other derived cost inputs**
  - ✓ Parametric cost models (i.e. COCOMO) adjust rough estimates by a series of 'cost drivers'

RELY – Req.Sw Reliability	**DOCU – Documentation**	**ACAP – Analyst Capability**	**PEXP – Platform Experience**
DATA – DB Size	**TIME – Exec. Time Constr.**	**PCAP – Program. Capability**	**LTEX – Lang./Tool Exper.**
CPLX – Prod. Complexity	**STOR – Storage Constraints**	**PCON – Personn. Continuity**	**TOOL – Use of Sw Tools**
RUSE – Req. Reusability	**PVOL – Platform Volability**	**AEXP – Applic. Experience**	**SITE – Multisite Devel.**
SITE – Multisite Commun.	**SCED – Req. Dev. Schedule**		

  - ✓ Each cost driver is…
    - Described as a 'nominal variable'
    - Broken down into 5 'ordinal' categories (from 'very low' to 'extra high')
  - ✓ …and is evaluated through an impact factor
    - Transformation of such inputs from cost drivers into fractions of 'days per size unit'
    - **Consequence**: input cost drivers are no longer direct inputs to estimation models **but** rather 'estimation sub-models' themselves"
    - …but such transformations are not documented nor supported by publicly available empirical data…
    - …therefore the quality of such estimation sub-models unknown
    - → weak basis for the estimation models themselves (black-box approach)

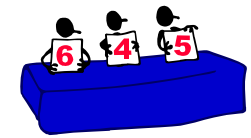
Analysis of the Outputs of the Estimation Models

Main used statistical criteria

- **3rd step: V&V of the quality of the outputs obtained**
  - ✓ Multiple statistical criteria for assessing the capability of an estimation model to properly predict the behaviour of the dependent variable
    - Coefficient of determination (R^2)
      - % of variability explained by the predictive variable; $0 \leq R^2 \leq 1$
    - Error of an estimate
      - MRE, MMRE, RMS, RRMS
    - Predictive quality of the model
      - PRED (l)=k/n; a reference value in Software Engineering: PRED(0.25)=0.75
    - p-value statistical parameter
      - Significance of the coefficient of the independent variables; ref. value → $p \leq 0.05$
  - ✓ Additional conditions
    - Large enough datasets
      - at least 30 data points **for each** independent parameter
    - A normal distribution of input parameters
    - No outlier which unduly influences the model
  - ✓ …otherwise…
    - 15-20 data points → models to use with care - no generalization
    - 4-10 data points → models merely anecdotical with no statistical strength

Evaluation of Estimation Models

Evaluation by Model Builders



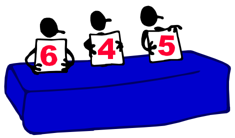
- **A typical evaluation by a model builder (COCOMO I, 1981):**

	MRE	PRED(0.25)
Basic		**25 %**
Intermediate		**68 %**
Detailed		**70 %**

- **COCOMO II (1997)**
  - ✓ Revision of the original COCOMO model (from 63 data points → 161 data points)
    - ✓ Updated the list of cost drivers
    - ✓ Added the usage of backfiring (LOC → FP)
  - ✓ Design of the revised method based mostly on opinions from domain experts rather than on empirical data
- **Conclusions**
  - ✓ COCOMO II model not based on empirical data but on expert opinions
    - ✓ Should be considered as **unproven 'theoretical' models**
      - ✓ quality is still far to be demonstrated

Evaluation of Estimation Models

Independent Evaluations



- **Evaluations by model builders**
  - ✓ Interesting but not necessarily complete

- **Independent evaluations**
  - ✓ COCOMO I – some numbers:
    - ✓ 63 data points
    - ✓ 3 models: Basic, Intermediate, Detailed
    - ✓ Parameters: 4 (basic), 18 (intermediate), 72 (Detailed)
  - ✓ What is required for a meaningful evaluation:
    - ✓ Model statistically significant if verified on:
      - ❖ Basic model: **120** projects (4 independent parameters by 30 data points)
      - ❖ Intermediate model: **540** projects (18 parameters by 30 data points)
      - ❖ Detailed model: **2160** projects (18 parameters by 4 project phases by 30 data points)

- **Summary:**
  - ✓ COCOMO users should not reply on the reported performance of Intermediate & Detailed models

Error Propagation in Estimation Models

- ✓ In **Science**
  - ❖ referable to an inherent uncertainty in all measurements and cannot be eliminated
- ✓ In **Science** & **Engineering**
  - ❖ Numbers without accompanying error estimates are suspect and possibly useless
  - ❖ Also true in Software Engineering
  - ❖ Some examples of uncertainty of simple functions:

Function	Function Uncertainty
$X = A \pm B$	$(\Delta X)^2 = (\Delta A)^2 + (\Delta B)^2$
$X = cA$	$\Delta X = c\,\Delta A$
$X = c(A \times B)$ or $X = c(A/B)$	$(\Delta X/X)^2 = (\Delta A/A)^2 + (\Delta B/B)^2$
$X = cA$	$\Delta X/X = \lvert n \rvert\,(\Delta A/A)$
$X = \ln(cA)$	$\Delta X = \Delta A/A$
$X = \exp(A)$	$\Delta X/X = \Delta A$

- ✓ In **Software Engineering**
  - ❖ Same concepts applicable in particular to parametric estimation models
  - ❖ + additional cost drivers, the more sources of uncertainty introduced into the estimation model → if not properly managed, a propagation of errors may result

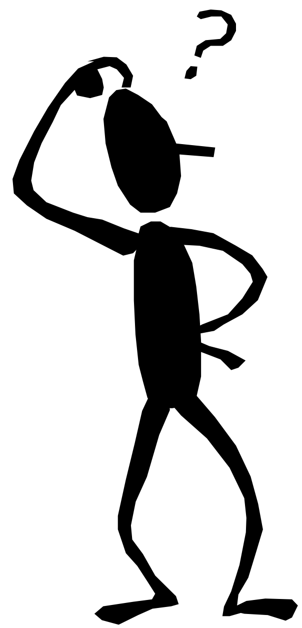
Summary

- Software project estimation is still a challenge for most software organizations and their customers:
  - Significant cost overruns and delays
  - Less functionalities delivered than promised
  - Unknown levels of quality (requested and delivered)
- Software estimation models
  - On-going research from more than 30 years ago
  - Several approaches considered, but few supported with enough historical data
- A major issue is the evaluation of the quality for such models
  - Criteria for V&V for:
    - Input parameters, derived inputs, outputs
  - Evaluation of such models by:
    - Their own builders
    - Independent reviewers
- Conclusions
  - Users of estimation models MUST control the quality of the estimation models they intend to use!

Food for thoughts

If your estimation model cannot adequatly explain past performance, how can you expect it to predict the future?

Q & A



Merci beaucoup! Thank you!

alain.abran@etsmtl.ca; luigi.buglione@eng.it