Software Metrics & Software Metrology

Alain Abran

Chapter 1 Introduction

This book includes 4 parts

- Part 1: Necessary concepts for a correct design of a software measurement method
    - including knowledge build over centuries in metrology
- Part 2: Examples to illustrate weaknesses in some of the most popular ‘software metrics’
- Part 3: Example of a software measurement method built to meet metrology & ISO requirements for software measurement
- Part 4: Examples of 2 key issues from metrology that are often neglected in most ‘software metrics:
    - Convertibility across distinct ‘metrics’ which claim to measure the same attribute
    - A standard-étalon for measurement

Part 1 of this book

- Part 1 of this book present the necessary concepts for a correct design of a software measurement method
  - including knowledge build over centuries in metrology

- Chapter 1 introduces some of the ambiguities in 'software metrics' and discusses their immaturity as technologies.
- Chapter 2 presents a measurement context model to help clarify the distinct concepts and steps in measurement
- Chapter 3 presents the classical concepts of measurement codified in the metrology body of knowledge that sustains measurement in science and engineering
- Chapter 4 discusses the differences between measurement in the metrological sense and quantification
- Chapter 5 discusses the design requirements for software measurement methods

Agenda

This 1^{st} chapter covers:

- Introduction to the term « metrics » & its ambiguities
- Software measurement: is it mature or not?
- Software measurement as a new technology
- The designs of software measures must be verified

Agenda

This 1st chapter covers:

- **Introduction to the term « metrics » & its ambiguities**
- Software measurement: is it mature or not?
- Software measurement as a new technology
- The designs of software measures must be verified

Introduction to the term « metrics »

- In software engineering, the term "metrics" is used in reference to multiple concepts.

For example:

- the quantity to be measured (measurand [1]),
- the measurement procedure,
- the measurement results or
- models of relationships across multiple measures, or
- measurement of the objects themselves.

- [1] A measurand is defined as a particular quantity subject to measurement [ISO International Vocabulary of Metrology (VIM)].

Introduction to the term « metrics »

In the software engineering literature, the term was, up until recently, applied to:

- Measurement of a concept: e.g. cyclomatic complexity,
- Quality models: e.g. ISO 9126 – software product quality, and
- Estimation models: e.g.
  - Halstead's effort equation,
  - COCOMO I and II – estimation models,
  - Use Case Points,
  - etc.

Introduction to the term « metrics »

Many curious problems have been identified:

- Proliferation of publications on metrics for concepts of interest, but with a very low rate of acceptance and use by either researchers or practitioners, as well as a lack of consensus on how to validate so many proposals.

- Inventory of software metrics so diversified and with so many individual proposals that it is not seen as economically feasible for either the industry or the research community to investigate each of the hundreds of alternatives proposed to date
  - Ex: to measure software complexity
  - To measure performance, etc.

Agenda

This 1st chapter covers:

- Introduction to the term « metrics »
- **Software measurement: is it mature or not?**
- Software measurement as a new technology
- The designs of software measures must be verified

Software Measurement: Is it Mature or Not?

- The IEEE Computer Society defines software engineering as:

  "(1) The application of a systematic, disciplined, ***quantifiable*** *approach to the development, operation, and maintenance of software; that is, the application of engineering to software.*

  (2) The study of approaches as in (1)" [IEEE 610.12]

  - It is obvious that measurement is fundamental to software engineering as an engineering discipline.
  - But what is the status of measurement within software engineering, and how mature is the field of knowledge on software measurement?

Software Measurement: Is it Mature or Not?

- Over recent decades, hundreds of so-called software metrics have been proposed by researchers and practitioners alike, in both theoretical and empirical studies, for measuring software products and software processes:

  - Most of these metrics were designed based either:
    - on intuition on the part of researchers or
    - on an empirical basis, or
    - both.

  - They have most often been characterized by the ease with which some entities of the development process can be counted automatically.

Software Measurement: Is it Mature or Not?

- Researchers have most often used the concepts of measurement *theory* as the foundation for their analytical investigation.

  - However, measurement theory deals with only a subset of the classical set of measurement concepts, and by focusing solely on measurement theory, researchers have investigated mainly:
    - the representation conditions,
    - the mathematical properties of the manipulation of numbers, and
    - the proper conditions for such manipulations [Fenton 1997, Zuse 1997].

  - Measurement in scientific fields, including engineering, is based on a large body of knowledge built up over centuries, even millennia, which is commonly referred to as "metrology".

Software Measurement: Is it Mature or Not?

- In the literature on software metrics, there is almost no reference to the classical concepts of metrology in investigations into the quality of the *metrics* proposed to the software engineering community.

  - Only recently have some of the metrology-related concepts been introduced in the software engineering community, including the selection of the ISO Vocabulary on Metrology (VIM) as the basis for measurement terminology for all future ISO standards on software measurement.

Software Measurement: Is it Mature or Not?

- One of the peculiarities of software engineering relative to other engineering and scientific disciplines is its lack of general use of quantitative data for decision making. Symptoms:
  - a very limited number of accepted software measures available to practitioners and recognized as mature enough to be recognized as international standards, and
  - a very small number of rigorous experimental studies (which constitute general practice in the engineering and medical fields, for example).

- In mature disciplines, there is:
  - a large international consensus on measures, in addition to established measurement methods and an etalon for each, and
  - a significant number of measuring instruments to be used in different contexts and under various constraints, many of them certified and calibrated against internationally recognized, and traceable, etalons.

Software Measurement: Is it Mature or Not?

- In the software measurement domain, adoption of ISO standards for measurement is very recent – and very limited:

ISO standards for measuring the functional size of software

- ISO 14143-1 on the mandatory set of characteristics of software functional size (i.e. a meta-standard) – Parts 2 to 6 of ISO 14143 are only technical reports
- 5 ISO recognized specific measurement methods to implement the quantification of these functional size characteristics: ISO 19761- COSMIC, ISO 20926-IFPUG, ISO 20968-MKII, ISO 24570-NESMA and ISO 29881-FISMA.
  - Note: The software functional size measurement process is not yet mature enough for there to be a single universal way of measuring software functional size.

ISO standards for the measurement of software quality

- The set of models of software product quality in ISO 9126-1.
- The 3 catalogs of + 250 “metrics” in Parts 2 to 4 of ISO 9126
  - They are **still only ISO technical reports**: much work remains to be done to bring them up to ISO standard status.

Software Measurement: Is it Mature or Not

What does this lack of maturity mean for software measurement?

- Many, if not most, of the software measures proposed to the industry have not been seriously analyzed, nor sufficiently mature.
  - In contrast to other fields of science and engineering, these software measures lack the credibility to be used as a basis for decision making

    .
- Verification criteria for software measures should be comprehensive, carefully defined, and agreed upon.
  - Designers of software measures should document how well their proposed measures meet these verification criteria.

Software Measurement: Is it Mature or Not

Impact of lack of credibility of software measures:

- It is not until it can be demonstrated unambiguously that a proposed measure achieves a high level of measurement quality that it can be expected to reach a level of credibility in the practitioner and manager communities, and then be used in practice on a large scale.

- Impact of the absence of software measure credibility:
  - when objective and quantitative data are required for decision making in software engineering, researchers and practitioners must often design and develop their own individual software measures and measurement methods
    - whereas these already exist in other fields of knowledge.

Agenda

This 1st chapter covers:

- Introduction to the term « metrics »
- Software measurement: is it mature or not?
- **Software measurement as a new technology**
- The designs of software measures must be verified

Software Measurement as a New Technology

Technology is defined as the set of methods and materials used to achieve industrial or commercial objectives:

- It also includes processes and the knowledge related to them, referred to as “know-how”.

From that perspective: software measurement = a technology.

- Mature technologies:
  - have been fined tuned over many years and have been adapted with a number of features and tools to fit various contexts and to facilitate their use by non experts.
- Innovative and immature technologies:
  - require significantly more expertise for using them in their ‘immature’ status.

Software Measurement as a New Technology

Software measurement is definitively a new technology, and, as such:

- it shares many of the characteristics of new technologies, and
- the constraints that must be tackled to facilitate its adoption:
  - By industry at large and
  - By individual practitioners.

Software Measurement as a New Technology

What does it take for a new technology to be adopted by an Organization?

- The new, initially unknown technology must promise enough benefits to overcome the pain of changing from a known one.
- The organization must have (or gain) the technological know-how to make it work.
- The organization must be clever enough, and enthusiastic enough, to harvest its benefits, which takes time.

Software Measurement as a New Technology

What does it take for a new technology to be adopted by industry?

- The new technology must become integrated into the industry's technological environment.
- It must also become integrated into the business context (which includes its legal and regulatory aspects).
- It must have been proven to work well in a large variety of application contexts (that is, the technology must be mature, or maturing rapidly).

Software Measurement as a New Technology

What does it take for an industry to promote a new technology?

- The industry must recognize that there is a direction that has been proven to work in similar contexts.
- It must recognize that current practices are not good enough.
- It must also recognize that the players will not, on their own, submit to the pain of change (unless the environmental-regulatory context forces such a change).
- It must want to speed up the transition to the new technology.

Software Measurement as a New Technology

What does it take for software measurement to be adopted as a new technology by a <u>software organization</u>?

- Software measurement must promise enough benefits to overcome the pain of changing to an initially unknown technology.
- The organization must have the technological know-how in software measurement to make it work.
- The organization must be clever enough, and enthusiastic enough, to harvest the benefits, which takes time.

Software Measurement as a New Technology

What does it take for software measurement to be adopted as a new technology by the software industry?

- Software measurement must become integrated into the technological environment of the software industry.
- It must become integrated into the business context (which includes its legal and regulatory aspects).
- Software measurement must already have been proven to work well in a large variety of contexts (that is, it must be mature as a technology, or maturing rapidly).

Software Measurement as a New Technology

What does it take for an industry to promote software measurement as a new technology?

- Software measurement must have been proven to work in similar contexts.
- Current software measurement practices must be good enough.
- The industry must recognize that the players will not, by themselves, submit to the pain of change (unless the environmental-regulatory context forces such a change).
- It should want to speed up the transition to quantitative support for decision making.

Agenda

This 1st chapter covers:

- Introduction to the term « metrics »
- Software measurement: is it mature or not?
- Software measurement as a new technology
- **The designs of software measures must be verified**

The Designs of Software Measures Must be Verified

- Software measurement must play an increasingly important role in software engineering if it is to truly become an engineering discipline.

  - Over the past 20 years, a significant number of software metrics have been proposed for better control and greater understanding of software development practices and products.
  - Unfortunately, very few of these metrics have been looked at closely from a measurement method perspective, and it is currently difficult, because of a lack of agreed-upon frameworks of verification and validation procedures, to analyze their quality.

The Designs of Software Measures Must be Verified

What constitutes a valid metrics, or even a valid measurement method? How can a measurement method be validated?

- Various authors have attempted to address these questions in recent years, and from different points of view (mathematical, practical, etc.). *(Refer to chapter 2)*

Examples of verification questions that need to be investigated

- Does the measurement method really measure the concept intended to be measured?
- Has the measurement method been internally verified,
  - i.e. in the sense that it can be shown that it gives a proper numerical characterization of the attribute to be measured?
- Is the measurement method usable?
  - A measurement method which is as perfect as possible from a mathematical viewpoint would not be of any interest if it could not be applied (if it were far too time-consuming, for example).

Summary

This 1st chapter has covered:

- Introduction to the term « metrics » and related issues
- The current immaturity of software measurement
- Software measurement as a new technology needs to be improved significantly to be truly useful
- The necessity to verify the quality of the designs of software measures:
  - because of the current immaturity of software measurement, the practitioners cannot take for granted that the available ‘software metrics’ have strong designs as measurement methods
  - The software measurement technology must be improved and such improvements are much needed by the software engineering community

- Additional material:
  - Advanced Readings: Measurement within the Software Engineering Body of Knowledge - SWEBOK