CHAPITRE 3

PROPRIÉTES DES MESURES EN GÉNIE LOGICIEL

Ce chapitre présente 70 propriétés des mesures de logiciels proposées par des chercheurs du domaine ainsi que six autres propriétés recommandées par la norme ISO 14143. Le chapitre présente une vue sur ces propriétés et leur utilité dans la conception des étalons.

3.1 Introduction

Dans son livre *A Framework for Software Measurement*, Zuse (1998) présente les raisons de formuler les propriétés des mesures des logiciels : *The reason for formulating proprieties of software measures is, among others, to provide a standard of software measures*. En effet, pour assurer la vérification des mesures, plusieurs chercheurs ont proposé des ensembles de propriétés, parfois appelés *axiomes*, que la mesure à vérifier devrait satisfaire. Mais, dans la littérature en génie logiciel, il n'y a pas de consensus, ce qui se traduit par un ensemble incohérent de propriétés, du fait que les attributs à mesurer peuvent avoir différentes interprétations. Par exemple, certains axiomes de Weyuker (1988) exigent que l'attribut de la complexité du logiciel soit défini comme étant sa compréhensibilité; pour d'autres chercheurs, tel que McCabe (1976), la complexité est liée à la testabilité du code source. Cette diversité d'opinions et cette absence de consensus rendent les utilisateurs potentiels de ces mesures sceptiques et les travaux pour normaliser les mesures elles-mêmes difficiles ainsi que le design et le choix des étalons de mesure une tâche ardue.

3.2 Inventaire des propriétés proposées pour les mesures des logiciels

Zuse présente dans son livre *A Framework of Software Measurement* un inventaire de 70 propriétés des mesures des logiciels désirables proposées par plusieurs chercheurs, qui

sont rapportées comme telles en langue anglaise dans les tableaux suivants pour respecter leur définition d'origine.

Les propriétés des mesures de logiciels proposées par Bache (1990) sont présentées au tableau 2.

Tableau II

Les propriétés des mesures de logiciels proposées par Bache

No	Nom de la propriété	Définition de la propriété
1	Axiom of Positivity	Positivtity means adding something increases the complexity.
2	Sequence is more Complex Than the Maximum of the Components	It says that the whole, consisting of a sequence, is more complex than the maximum of the components.
3	Weak Commutativity	Is defined as u(P1 o P2) = u(P2 o P1) for all P1, P2 e P, where P is, for example, a set of flowgraphs and u is a measure.
4	Strongest Independence Substitution Condition	
5	Invariance of Nesting	
6	Dominance of the Relation H > G in Nested Flowgraph I	
7	Dominance of the Relation H > G in Nested Flowgraph I	
8	Nested Structures are more Complex than Sequences	
9	Dominance of a more Complex Component	
10	VINAP Measures	Measures control flow complexity.

Les propriétés des mesures de logiciels identifiées par Conte et al. (1986) sont présentées au tableau 3.

Tableau III

Les propriétés des mesures de logiciels identifiées par Conte et al.

No	Nom de la propriété	Définition de la propriété
1	Simplicity	Does the measure lead to a simple result? The measure should produce a single value.
2	Robustness	Is the measure sensitive to the artificial manipulation of some factors that do not affect the performance of the software?
3	Predescriptiveness	Can the measure be used to guide the management of software development or maintenance?
4	Analyzeability :	Can the value of the measure be analyzed using standard statistical tools?

IEEE (1993) a publié son standard 1061 *Quality Metrics Methodology* qui contient des directives pour la validation des mesures de logiciel. Les propriétés des mesures de logiciels exigées par IEEE sont présentées au tableau 4.

Tableau IV

Les propriétés des mesures de logiciels exigées par IEEE

No	Nom de la propriété	Définition de la propriété
1	Tracking	If a metric M is directly related to a quality factor F, for a given product or process, then a change in a quality factor value from F_{T1} to F_{T2} at times T1 and T2, shall be accomplished by a change in metric value from M_{T1} to M_{T2}, which is the same direction.
2	Consistency	If factor values F_1, F_2,...,F_n, corresponding to products or processes 1,2, ..., n, which have the relationship $F_1 > F_2 > F_n$, then the corresponding metric values shall have the relationship $M_1 > M_2 > M_n$.
3	Discriminative power	A metric shall be able to discriminate between high quality software components and low quality software components. For instance, the set of metric values associated with the former should be significantly higher than those with latter.

Tableau IV (Suite)

Les propriétés des mesures de logiciels exigées par IEEE

No	Nom de la propriété	Définition de la propriété
4	Reliability	A metric shall demonstrate the above correlation, tracking, consistency, predictability, and discriminative power properties for P percent of the application of the metric.

Les propriétés des mesures de logiciels exigées par Basili et Reiter (1979) sont présentées au tableau 5.

Tableau V

Les propriétés des mesures de logiciels exigées par Basili et Reiter

No	Nom de la propriété	Définition de la propriété
1	Sensitivity	The measure should be sensitive to externally observable differences in the development environment.
2	Intuitiveness	The relative values of the measure for specific environments should correspond to some intuitive notion about the characteristic differences for those environments.

Les propriétés des mesures de logiciels exigées par Ejiogu (1991) sont présentées au tableau 6.

Tableau VI

Les propriétés des mesures de logiciels exigées par Ejiogu

No	Nom de la propriété	Définition de la propriété
1	Empirically and Intuitive Persuasive	The empirical behaviour of the target being measured should dictate its manner of measurement.

Tableau VI (Suite)

Les propriétés des mesures de logiciels exigées par Ejiogu

No	Nom de la propriété	Définition de la propriété
2	Simple and Computable	The measure should be convenient to teach and use by the general practitioner or student; should require only simple and well-formed formulas.
3	Consistent and Objective	An independent observer or practitioner should be able to confirm the same measure using the same formula or guidelines.
4	Measure Rationalism	Must belong to the class of mathematical functions called measure functions; only these satisfy the well-known postulates of measure functions.
5	Consistency of Units and Dimensions	Must satisfy the mathematical validity of units and dimensions; combinations of target measures must only be made in admissible mathematical fashions.
6	Programming language independence	Each measure should be expressive of the software structure and not dependent on the "internals" of any programming language; in other words, the formulas should be invariant with respect to every programming language.
7	Feedback Effect	Essentially, the measure should be used to reinforce good programming habits and expedite debugging.

Les propriétés des mesures de logiciels exigées par Fenton sont présentées au tableau 7.

Tableau VII

Les propriétés des mesures de logiciels exigées par Fenton

No	Nom de la propriété	Définition de la propriété
1	Sequential Determinism	The complexity of a sequential flowgraph should be uniquely determined by the complexities of the components.
2	Non-Sequential Determinism	The complexity of a non-sequential flowgraph should be determined by both, the complexity of the components and the complexity of the non-sequential structure.
3	Linear Combination	If a measure satisfies a proper axiom scheme then any positively weighted linear combination also satisfies that axiom scheme.

Les propriétés des mesures de logiciels exigées par Jones (1994) sont présentées au tableau 8.

Tableau VIII

Les propriétés des mesures de logiciels exigées par Jones

No	Nom de la propriété	Définition de la propriété
1	Standard Definition	The metric should have a standard definition and be unambiguous.
2	Large scale statistics	The metric should not be biased and unsuited for large scale statistical studies.
3	Formal User Group	The metric should have a formal user group and adequate published data standard definition and be ambiguous.
4	Measures and Tools	The metric should be supported by tools and automation.
5	Conversion Rules Between Measures	It is helpful to have conversion rules between the metric and other metrics.
6	Not Code but Software Deliverables	The metric should deal with all software deliverables, and not just code.
7	Support all Types of Software Projects	The metric should support all types of software projects.
8	Support all Types of Programming Languages	The metric should support all types of Programming Languages.

Les propriétés des mesures de logiciels exigées par Kearny et al. (1986) sont présentées au tableau 9.

Tableau IX

Les propriétés des mesures de logiciels exigées par Kearny et al.

No	Nom de la propriété	Définition de la propriété
1	Robustness	If software complexity measures are to be used to evaluate programs, then it is important to consider the measures responsiveness to program modifications.
2	Normativeness	The interpretation of complexity measurements is facilitated if the metrics provide a norm against which measurements can be compared.
3	Specifity	Software complexity analysis may provide an assessment tool that can be used during program development and testing.
4	Predescriptive-ness	If software complexity measures are to prove useful in the containment of program complexity, then they must not only index the level of program's complexity, but also should suggest methods to reduce the amount of complexity.
5	Property Definition	The properties identified above are not rigorously defined, and it is sometimes difficult to tell whether or not a measure possesses one or another of these properties. Although a list of properties

Dans son ouvrage, Zuse (1998) confirmait que les propriétés des mesures de logiciel suggérées par Weyuker (1988) sont désirées, largement connues et utilisées pour l'évaluation des mesures de logiciel. Il réaffirmait que l'avantage est leur présentation claire qui leur permet d'être discutées facilement. Ces propriétés sont présentées au tableau 10.

Tableau X

Les propriétés des mesures de logiciel suggérées par Weyuker

No	Nom de la propriété	Définition de la propriété
1	Basic Assumption of a Measure	A measure should distinguish between at least two programs P and Q in at least one case.

Tableau X (Suite)

Les propriétés des mesures de logiciel suggérées par Weyuker

No	Nom de la propriété	Définition de la propriété
2	Finitely many Identifiers	Let c be a non-negative number, then there are only finite many programs of complexity c.
3	Equivalence Classe	There are programs P and Q such that u(P) = u(Q), where u is a measure.
4	Same Functionality, but different Complexity	Even though two programs the same function, it is the details of the implementation that determine the program's complexity.
5	Weak Positivity	The measure should be sensitive to adding something.
6	Rejection of the Weakest Independence Condition C1	Programs are objects composed from simpler programs (or more properly program bodies). Thus it is important to consider the relative complexities of program bodies. The Condition C1 means if a program P1 is equal to program P2 it implies that P1 concatenated with P are equal to P2 concatenated with P. The implication is irreversible.
7	Weak Commutativity	Program complexity should be responsive to the order of statements and hence the potential interaction among statements.
8	Renaming	If P is a renaming of Q, then u(P) = u(Q). This property states that uniformly changing variable names should not affect a program's Complexity.
9	Wholeness	The whole must be at least as big as the sum of the parts.

Les propriétés des mesures de logiciels exigées par Lakshmanan et al. (1991) sont présentées au tableau 11.

Tableau XI

Les propriétés des mesures exigées par Lakshmanan et al.

No	Nom de la propriété	Définition de la propriété
1	Non-negativity	If the program only contains sequential code (referred to as a basic block B) then Complexity(SB) = 0. If the program X is not a basic block, then Complexity(SX) > 0
2	Functional independence under sequencing	Complexity(SX;B) = Complexity(SX) Complexity(SX;B) = Complexity(SX) + Complexity(SB) = Complexity(SX)
3	Symmetry under sequencing	Complexity(SX;Y) = Complexity(SY;X) Complexity(SX;Y) = Complexity(SX) + Complexity(SY) = Complexity(SY;X)
4	Monotonicity under sequencing	Complexity(SX;Y) < Complexity(SX;Z) if Complexity(SY) < Complexity(SZ) Complexity(SX;Y) = Complexity(SX;Z) if Complexity(SY) = Complexity(SZ)
5	Additivity under sequencing	Complexity(SX;Y) = Complexity(SY) + Complexity(SX)
6	Functional independence under nesting	Adding a basic block B to a system X through nesting does not increase its complexity Complexity(SB@X) = Complexity(SX)
7	Monotonicity under nesting	Complexity(SY@Xi) < Complexity(SZ@Xi) if Complexity(SY) < Complexity(SZ)
8	Monotonicity under nesting	Complexity(SY) < Complexity(SY@X)
9	Sensitivity to nesting	Complexity(SX;Y) < Complexity(SY@X) if Complexity(SY) > 0

Les propriétés des mesures de logiciels exigées par Watts (1987) sont présentées au tableau 12.

Tableau XII

Les propriétés des mesures de logiciels exigées par Watts

No	Nom de la propriété	Définition de la propriété
1	Comparability	The measure must be comparable with other measures of the same criteria
2	Economy	The simpler, and therefore, the cheaper the measure is to use, the better
3	Objectivity	The results should be free from subjective influences. It must not matter who the measurer is
4	Reliability	The result should be precise and repeatable
5	Standardisation	The measure should be unambiguous and allow for comparison
6	Usefulness	The measure must address a need, not simply measure for its own sake
7	Validity	The metric should measure the right attribute

Navlaka (1986) exige qu'une bonne mesure ait les propriétés suivantes (voir tableau 13 suivant).

Tableau XIII

Les propriétés des mesures de logiciels exigées par Navlaka

No	Nom de la propriété	Définition de la propriété
1	Correctness	From the same data and rules, the same results must always be obtained.
2	Repetitiveness	It doesn't matter the person who makes the measurement, results must always be exactly the same at different time.

Les propriétés des mesures de logiciels définies par ISO TR 14143-3 (2002) sont présentées au tableau 14.

Tableau XIV

Les propriétés des mesures de logiciels définies par ISO 14143-3

No	Nom de la propriété	Définition de la propriété
1	Répétitivité	L'étroitesse de l'accord entre les résultats de mesurages successifs du même mesurande, mesurages effectués dans la totalité des mêmes conditions de mesure (AFNOR, 1994).
2	Reproductibilité	L'étroitesse de l'accord entre les résultats des mesurages du même mesurande, mesurages effectués en faisant varier les conditions de mesure (AFNOR, 1994).
3	Exactitude	L'étroitesse de l'accord entre les résultats d'un mesurage et une valeur vraie d'un mesurande.
4	Convertibilité	La mesure est convertible en d'autres mesures
5	Seuil de Tolérance	
6	Applicabilité aux domaines fonctionnels	

3.3 Constatation sur les propriétés des mesures des logiciels

D'après l'inventaire ci-dessus des propriétés exigées par les chercheurs en mesure des logiciels, neuf propriétés seulement ont été recommandées par plus d'un groupe de chercheurs. Ce sont la standardisation, la simplicité, l'uniformité, l'objectivité, l'indépendance du langage de programmation, la fiabilité, la robustesse, la pre-description et l'intuition. Il est important de noter que la propriété « standardisation » est la plus exigée, ce qui rejoint l'avis de l'équipe de cette recherche et renforce la motivation de développer un référentiel afin d'améliorer la standardisation des mesures des logiciels.

La majorité des propriétés des mesures de logiciels exigées par Ejiogu sont citées par d'autres chercheurs. Les propriétés données par Watts et Navlaka attirent plus l'attention

de l'équipe de cette recherche car elles sont plus près de sa compréhension de la mesure des logiciels.

Pour la construction du référentiel, cet inventaire est à considérer et en particulier les propriétés de mesure proposées par la norme ISO 14143-3 qui sont la répétabilité, la reproductibilité, l'exactitude, la convertibilité, le seuil de discrimination et l'applicabilité aux domaines fonctionnels, pour donner plus de crédibilité à ce travail.

3.4 Les étalons de mesure

Parmi toutes les propriétés citées, une en particulier a été choisie pour la problématique de recherche choisie pour cette thèse, soit la standardisation des mesures. Une définition récente d'un étalon de mesure est présentée par (Antoine *et al.*, 2003)) : « Un étalon est un type de grandeur qui sert à déterminer ou à représenter matériellement l'unité de mesure, éventuellement l'un de ses multiples. Un étalon doit être précis, exact, reproductible et universel ». Cette recherche, toutefois adopte la définition reconnue par la communauté internationale de la normalisation en mesure, soit celle définie dans le *Vocabulaire international des termes fondamentaux et généraux de métrologie* : « Un étalon est défini comme mesure matérialisée, appareil de mesure, matériau de référence destiné à définir, réaliser, conserver ou reproduire une unité ou une ou plusieurs valeurs d'une grandeur pour servir de référence. » (ISO, 1993).

Selon le dictionnaire de sciences économiques, la standardisation est « la fixation de normes rigoureuses pour la fabrication de produits ou de leurs composants, qui deviennent ainsi tous identiques » (2003); la standardisation débouche sur le fait que les produits, parce qu'ils sont identiques, sont interchangeables. Lorsque l'on étudie certaines caractéristiques des mesures des logiciels sur deux types de logiciels distincts, les différences observées peuvent être affectées par des facteurs qui différencient les deux logiciels (par exemple, l'environnement de développement et l'expérience des développeurs peuvent être différents d'un logiciel à l'autre). La standardisation est un

moyen de réduire les effets de ces facteurs et, par conséquent, de rendre les mesures effectuées comparables.

3.5 Conclusion

En génie logiciel, il est loin d'y avoir un consensus sur les propriétés requises pour les mesures du logiciel et l'inventaire dressé plus haut en fait le constat. La recherche d'un consensus sur les propriétés des mesures des logiciels doit préoccuper aussi bien les chercheurs que les industriels. Ce chapitre a présenté un inventaire dressé à partir des propriétés désirables des mesures des logiciels proposées par plusieurs chercheurs du domaine et exposé par Zuse dans son livre *A Framework for Software Measurement*, et des propriétés des mesures des logiciels recommandées par la norme ISO 14143-3. Cet inventaire peut aider dans la sélection des propriétés des mesures de logiciel, comme les mesures de la taille fonctionnelle des logiciels. De cet inventaire, une seule propriété est retenue comme focus de recherche soit la **standardisation** des mesures.

Afin d'approfondir les connaissances en matière de standardisation des mesures, le chapitre suivant présente les concepts de standardisation et de métrologie.