Mapping Between ISO 9126 on Software Product Quality Metrics and ISO 12207 on Software Life Cycle Processes

Rafa E. Al-Qutaish and Hamed J. Al-Fawareh

*École de Technologie Supérieure (ÉTS), University of Québec, 1100 Notre-Dame St., Montréal, QC, H3C 1K3, Canada, rafa.al-qutaish.1@ens.etsmtl.ca
**Dept. of Computer Science, Zarqa Private University, P. O. Box 2000, Zarqa 13110, Jordan, fawareh@zpu.edu.jo

ABSTRACT

Nowadays, the International Organization for Standardization (ISO) is working on the next generation of the software product quality standards which will be referred to as Software Product Quality Requirements and Evaluation (SQuaRE – ISO 25000 series). However, this series of standards will replace the current version of ISO 9126 International Standard which consists of inventories of proposed metrics to measure the quality of the internal, external, and in-use software product. For each of these metrics there is a cross-reference on where they could be applied (measured) during the ISO 12207 Software Life Cycle Processes and activities (SLCP). This paper provides a mapping between those two standards to highlights the weaknesses of these cross-references and proposes a number of suggestions to address them.

Keywords: *Software Measurement, Software Quality Metrics, Software Life Cycle Processes (SLCP), ISO 9126, ISO 12207.*

1. INTRODUCTION

Measurements have a long tradition in natural sciences. At the end of the 19th century the physicist, Lord Kelvin, formulated the following about measurement: "*When you can measure what you are speaking about, and express it into numbers, you know some thing about it. But when you can not measure it, when you can not express it in numbers, your knowledge is of a meager and unsatisfactory kind: It may be the beginning of knowledge, but you have scarcely in your thoughts advanced to stage of science*" [20].

Moreover, Roberts [21] points out in his book about measurement theory that: "*A major difference between well-developed sciences such as physics and some of the less well-developed sciences such as psychology or sociology is the degree to which things are measured*".

In the area of software engineering, the concept of software measurement (or what is commonly called software "metrics") is not new. Since 1972, a number of so-called software "metrics[1]", or "measures", have been developed. From the wide range of software measures, four basic theories have been the source of the majority of the research conducted on software measurement. Some of these measures have been defined by Halstead [7, 8], Albrecht [3], DeMarco [5], and McCabe [19].

The definition of a "measure" is an empirical objective assignment of a number or a symbol to an entity to characterize a specific attribute [6]. Moreover, Ince *et al.* [11] have defined the software "metrics" as numerical values of quality which can be used to characterized how good or bad that the product is in terms of properties such as its proneness to error. In Addition, "metrics" defined in [10] as quantitative measures of the degree to which a system, component, or process possesses a given attribute, while within the ISO 15939, it was defined as a variable to which a value is assigned as a result of measurements [15].

In order to standardize the software product quality measurement process, in 1991, the ISO published its first international consensus on the terminology for the quality characteristics for software product evaluation; this standard was called as Software Product Evaluation - Quality Characteristics and Guidelines for Their Use (ISO 9126: 1991) [14].

From 2001 to 2004, the ISO published an expanded version, containing both the ISO quality models and inventories of proposed measures for these models (ISO 9126 parts 1, 2, 3, and 4) [12, 16-18].

Recently, the ISO has recognized a need for further enhancement of ISO 9126 International Standard, primarily as a result of advances in the fields of information technologies and changes in environment [4]. Therefore, the ISO is now working on the next generation of software product quality standards [22], which will be referred to as Software Product Quality Requirements and Evaluation (SQuaRE – ISO 25000

[1] While the term "metrics" is used in ISO 9126, the use of this term will be abandoned and replaced by "measures" in the upcoming new ISO 25000 as an initial step towards harmonizing the software engineering measurement terminology with the ISO 15939. [11]

series). This series of standards will replace the current ISO 9126 International Standard. However, many researches have focused on some weaknesses on the current ISO 9126 and even on the draft versions of the upcoming new ISO 25000 series of standards (SQuaRE) [1, 2].

The current version of the ISO 9126 consists of inventories of proposed metrics to measure the quality of the internal, external, and in-use software product. However, for each of these metrics there is a cross-reference on where they could be applied (measured) during the ISO 12207 Software Life Cycle Processes and activities (SLCP). This paper provides a mapping between these two standards to highlights the weaknesses of these cross-references and proposes a way to address them.

This paper is organized as follows: section 2 presents an overview of the related software engineering standards, that is, ISO 9126 and ISO 12207. Section 3 shows a detailed mapping of the ISO 9126 metrics to where they could be measured during the Software Life Cycle Processes (SLCP) provided by ISO 12207. Section 4 discusses the results of this mapping. Finally, Section 5 concludes the paper with some comments and suggestions.

2. RELATED SOFTWARE ENGINEERING ISO STANDARDS

2.1 ISO 9126

The ISO 9126 series of standards now consists of one International Standard [12] and three Technical Reports [16-18]:

1. ISO 9126-1: Quality Model [12].
2. ISO TR 9126-2: External Metrics [16].
3. ISO TR 9126-3: Internal Metrics [17].
4. ISO TR 9126-4: Quality in Use Metrics [18].

The first document of the ISO 9126 series – Quality Model – contains two-parts quality model for software product quality [12]:

1. Internal and external quality model.
2. Quality in use model.

The first part of the two-parts quality model determines six characteristics in which they are subdivided into twenty-seven subcharacteristics for internal and external quality, as in Figure 1 [12]. These subcharacteristics are a result of internal software attributes and are noticeable externally when the software is used as a part of a computer system. The second part of the two-part model indicates four quality in use characteristics, as in Figure 2 [12].

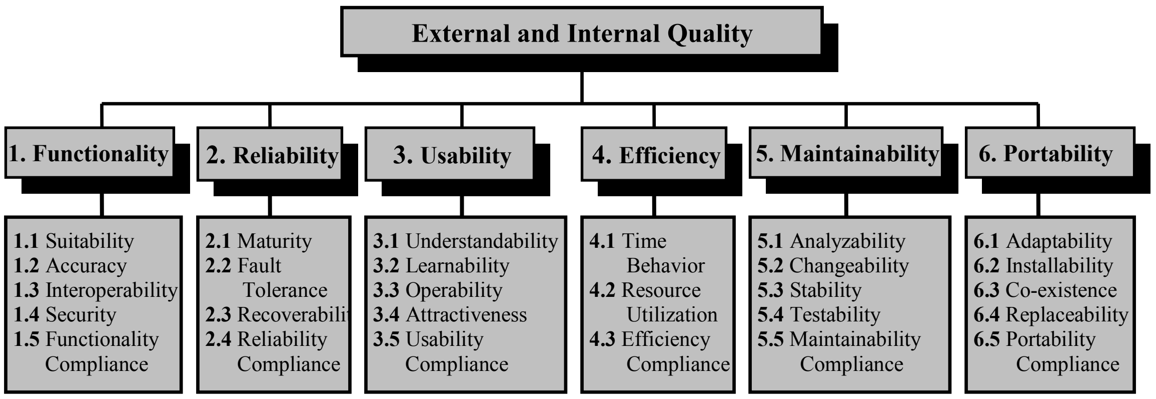


Figure 1: ISO 9126 Quality Model for External and Internal Quality (Characteristics and Subcharacteristics) [12].

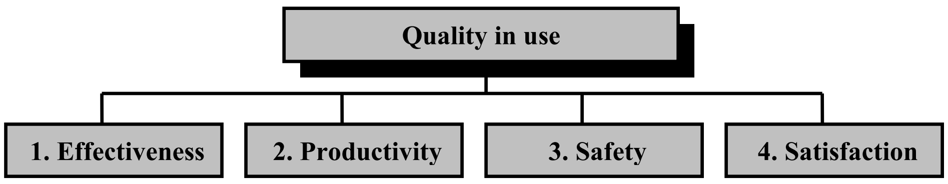


Figure 2: ISO 9126 Quality Model for Quality in Use (characteristics) [12].

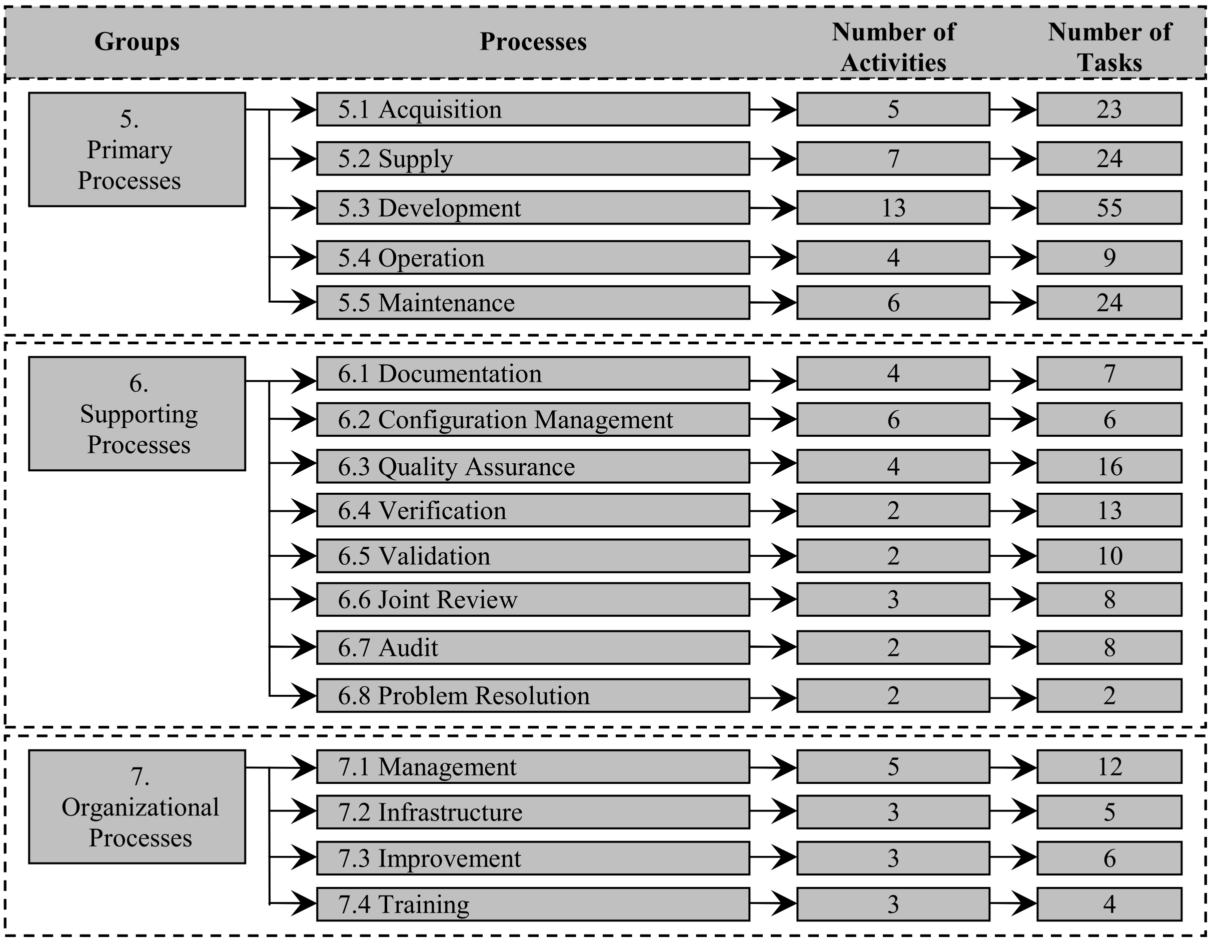
The second, third, and fourth documents of the ISO 9126 series provide the following information [16]:

1. Sets of metrics for each external quality sub-characteristic, internal quality sub-characteristic, and quality in use characteristic.
2. Explanations of how to apply and use these sets of metrics.
3. Examples of how to apply these metrics during the software product lifecycle.

2.2 ISO 12207

It consists of processes, activities for each process, and tasks for each activity [9, 13]. Figure 3 shows the software life cycle processes, the number of activities in each process, and the number of tasks in each process. The full list of the process, activities, and tasks can be seen in ISO 12207 and IEEE/EIA 12207 (the IEEE/EIA 12207 is the IEEE version of the ISO 12207).

The ISO 12207 software life cycle processes are grouped into three broad classes: primary; supporting; and organizational. Primary processes are the prime movers in the life cycle; they are acquisition, supply, development, operation, and maintenance. Supporting processes are documentation, configuration management, quality assurance, joint review, audit, verification, validation, and problem resolution. A supporting process supports another process in performing a specialized function. Organizational processes are management, infrastructure, improvement, and training. An organization may employ an organizational process to establish, control, and improve a life cycle process.



Groups	Processes	Number of Activities	Number of Tasks
5. Primary Processes	5.1 Acquisition	5	23
	5.2 Supply	7	24
	5.3 Development	13	55
	5.4 Operation	4	9
	5.5 Maintenance	6	24
6. Supporting Processes	6.1 Documentation	4	7
	6.2 Configuration Management	6	6
	6.3 Quality Assurance	4	16
	6.4 Verification	2	13
	6.5 Validation	2	10
	6.6 Joint Review	3	8
	6.7 Audit	2	8
	6.8 Problem Resolution	2	2
7. Organizational Processes	7.1 Management	5	12
	7.2 Infrastructure	3	5
	7.3 Improvement	3	6
	7.4 Training	3	4

Figure 3: ISO 12207 Software Life Cycle Processes, Activities, and Tasks.

3. MAPPING BETWEEN ISO 9126 AND ISO 12207

For each metric of the internal, external, and in-use metrics, the ISO 9126 parts 2, 3, and 4 provides the following information:

- Metric name.
- Purpose of the metric.
- Method of application.
- Measurement formula.
- Interpretation of Measured value.
- Metric scale type.
- Measure type.
- Input to measurement.
- ISO 12207 SLCP Reference.
- Target audience.

Within the following subsections, detailed mappings between the ISO 9126 quality metrics of the Internal, external, and in-use software product and the ISO 12207 software life cycle processes and activities will be provided. In more details, this mapping will focus on an investigation of the “ISO 12207 Software Life Cycle Processes (SLCP) References” provided by ISO 9126 for each of its metrics.

3.1 INTERNAL QUALITY METRICS

Within the ISO 9126-3 on software product internal quality metrics, there is 70 metrics. These metrics can be applied during the software life cycle. Internal quality defined in ISO 9126-1 as the totality of characteristics of the software product from an internal view. Internal quality is measured and evaluated against the internal quality requirements. Details of software product quality can be improved during code implementation, reviewing and testing, but the fundamental nature of the software product quality represented by internal quality remains unchanged unless redesigned [12].

Figure 4 shows the number of internal quality metrics which can be applied (measured) during each of the ISO 12207 software life cycle processes. For example, within the "verification process" (of the "supporting processes") 59 metrics can be applied (measured). As an example, Appendix A shows a detailed structure of the software product internal quality metrics' names and where they can be measured during the software life cycle processes or activities along with the corresponding characteristic and subcharacteristic for each of those metrics. In this appendix, only the software life processes/activities which have internal quality metrics are mentioned.

However, from Figure 4 we can note that there is no metrics which could be measured during 4 out of 5 primary life cycle processes. This means that there is no any metric from ISO 9126 external quality metrics could be useful during the acquisition, supply, operation, and maintenance primary life cycle processes. Moreover, there is no metrics which could be measured during 3 out of 8 of the supporting life cycle processes; that is, documentation, configuration management, and audit processes.

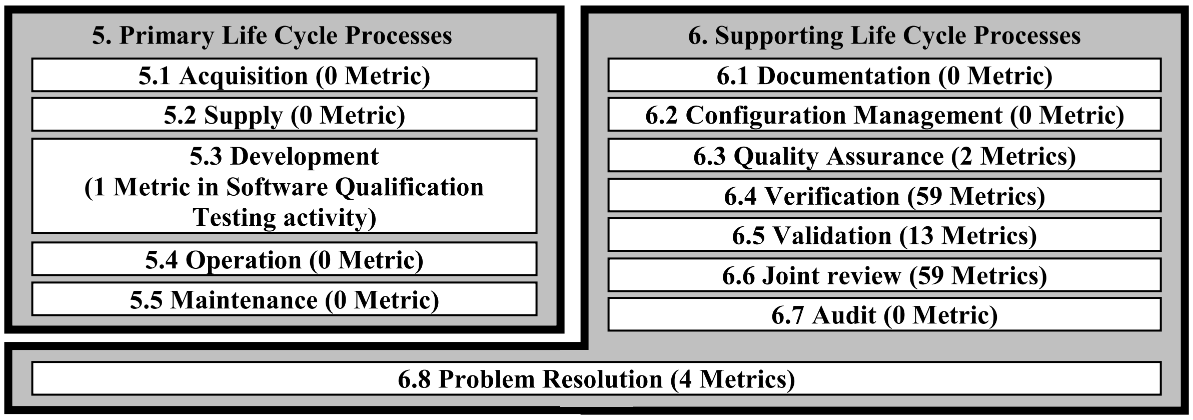

5. Primary Life Cycle Processes
5.1 Acquisition (0 Metric)
5.2 Supply (0 Metric)
5.3 Development (1 Metric in Software Qualification Testing activity)
5.4 Operation (0 Metric)
5.5 Maintenance (0 Metric)
6. Supporting Life Cycle Processes
6.1 Documentation (0 Metric)
6.2 Configuration Management (0 Metric)
6.3 Quality Assurance (2 Metrics)
6.4 Verification (59 Metrics)
6.5 Validation (13 Metrics)
6.6 Joint review (59 Metrics)
6.7 Audit (0 Metric)
6.8 Problem Resolution (4 Metrics)

Figure 4: The ISO 9126-3 Internal Quality Metrics and where they could be measured in the SLCP.

3.2 EXTERNAL QUALITY METRICS

Within the ISO 9126-2 on software product external quality metrics, there is 110 metrics. These metrics can be applied during the software life cycle. External quality defined in ISO 9126-1 as the totality of characteristics of the software product from an external view. It is the quality when the software is executed, which is typically measured and evaluated while testing in a simulated environment with simulated data using external metrics. During testing, most faults should be discovered and eliminated. However, some faults may still remain after testing. As it is difficult to correct the software architecture or other fundamental design aspects of the software, the fundamental design usually remains unchanged throughout testing [12].

Figure 5 shows the number of external quality metrics which can be applied (measured) during each of the ISO 12207 software life cycle processes. For example, within the "operation process" of the "primary processes", 93 metrics can be applied (measured).

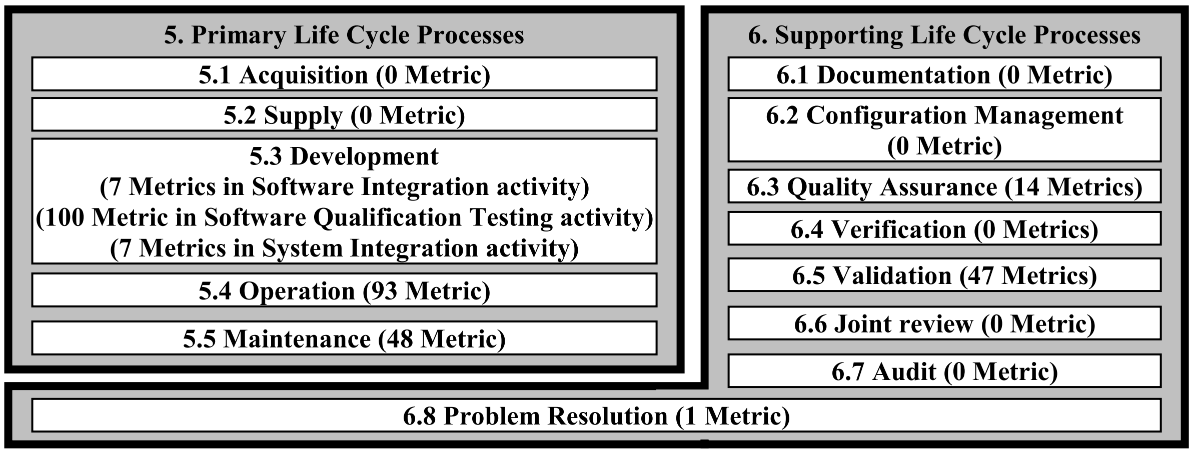

5. Primary Life Cycle Processes
5.1 Acquisition (0 Metric)
5.2 Supply (0 Metric)
5.3 Development (7 Metrics in Software Integration activity) (100 Metric in Software Qualification Testing activity) (7 Metrics in System Integration activity)
5.4 Operation (93 Metric)
5.5 Maintenance (48 Metric)
6. Supporting Life Cycle Processes
6.1 Documentation (0 Metric)
6.2 Configuration Management (0 Metric)
6.3 Quality Assurance (14 Metrics)
6.4 Verification (0 Metrics)
6.5 Validation (47 Metrics)
6.6 Joint review (0 Metric)
6.7 Audit (0 Metric)
6.8 Problem Resolution (1 Metric)

Figure 5: The ISO 9126-2 External Quality Metrics and where they could be measured in the SLCP.

3.3 QUALITY IN USE METRICS

Within the ISO 9126-2 on software product quality in use metrics, there is 15 metrics. These the 15 metrics can be applied during the software life cycle. Quality in Use defined in ISO 9126-1 as the user's view of the quality of the software product when it is used in a specific environment and a specific context of use. It measures the extent to which users can achieve their goals in a particular environment, rather than measuring the properties of the software itself. The term 'user' refers to any type of intended users, including both operators and maintainers, and their requirements can be different. [12].

Figure 6 shows the number of quality in use metrics which can be applied (measured) during each of the ISO 12207 software life cycle processes. For example, during the "software qualification testing" activity of the "development process" of the "primary processes", 12 metrics can be applied (measured).

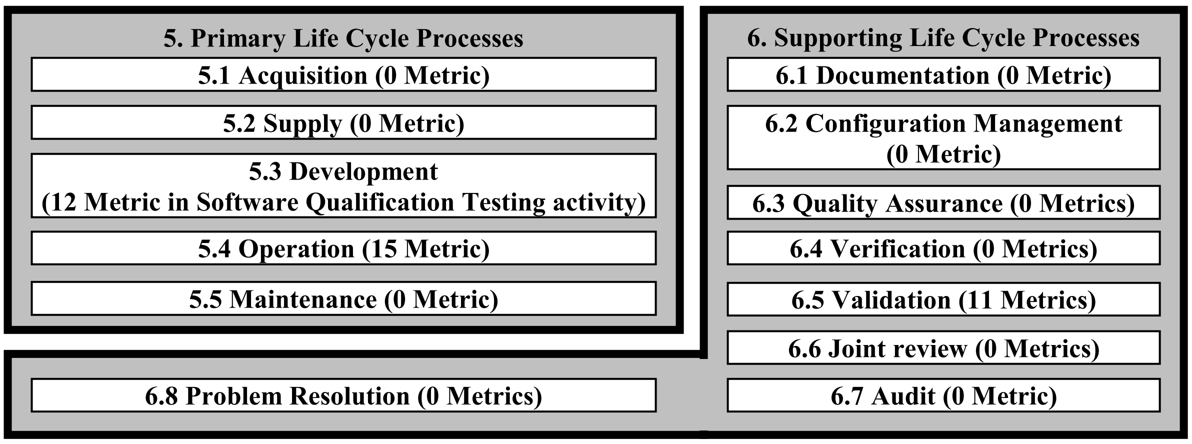

5. Primary Life Cycle Processes
5.1 Acquisition (0 Metric)
5.2 Supply (0 Metric)
5.3 Development (12 Metric in Software Qualification Testing activity)
5.4 Operation (15 Metric)
5.5 Maintenance (0 Metric)
6.8 Problem Resolution (0 Metrics)
6. Supporting Life Cycle Processes
6.1 Documentation (0 Metric)
6.2 Configuration Management (0 Metric)
6.3 Quality Assurance (0 Metrics)
6.4 Verification (0 Metrics)
6.5 Validation (11 Metrics)
6.6 Joint review (0 Metrics)
6.7 Audit (0 Metric)

Figure 6: The ISO 9126-4 Quality in Use Metrics and where they could be measured in the SLCP.

4. DISCUSSION OF THE MAPPING

In ISO 9126-3, there are some external quality metrics – as in Table 1 - which have been referred to be applied during the "integration" activity of the "development process" of the "primary processes". But, within the "development" process, there are two activities related to the "integration", that is, "system integration" and "software integration". However, this document (ISO 9126-3) did not specify during which "integration" activity those metrics can be applied (measured).

Table 1: Some External Quality Metrics.

1- Estimated latent fault density	2- Incorrect operation avoidance
3- Failure density against test cases	4- Availability
5- Failure resolution	6- Mean down time
7- Fault density	8- Mean recovery time
9- Fault removal	10- Restartability
11- Mean time between failures (MTBF)	12- User support functional consistency
13- Breakdown	14- Restore effectiveness
15- Failure avoidance	16- Restorability

It is clearly noted that through the ISO 12207 "organizational processes" none of the 195 quality metrics - found in ISO 9126 series of international standards - can be applied (measured).

As mentioned in ISO 9126-1, the quality in use metrics should be measured during the execution of the software product in an actual working environment. However, in Figure 6 we can see that there is 12 metrics which could be measured through the "software qualification testing" activity. But since ISO 12207 mentioned that the "software qualification testing" activity is a part of the "development process" Thus, it's strange and make no sense to measure the 12 metrics

The "joint review process" of the "supporting processes" consists of three activities; one of these activities is the "technical reviews" activity. The "technical reviews" activity contains one task that is, "*Technical reviews shall be held to evaluate the software products or services under consideration and provide evidence that: a) they are complete . . .* " [9]. Now, if we go back to Figure 4, we will find that there is 59 internal quality metrics that could be measured during the "joint review" process. Whereas, from Figures 5 and 6, it is seen that there is no any external quality or quality in use metrics that can be applied during "joint review" process.

5. CONCLUSION

The current edition of the ISO 9126 consists of inventories of proposed metrics to measure the quality of the internal, external, and in-use software product. However, for each of these metrics there is a cross-reference on where it could be applied (measured) during the ISO 12207 software life cycle processes and activities. This paper provided a mapping between those two standards to investigate the cross-references between them. Based on this mapping, the following comments and suggestions for the upcoming new ISO 25000 series of standards (SQuaRE) can be concluded:

- There is no any metric can be measured during the "organizational processes".
- A number of external quality metrics where mentioned in ISO 9126-2 to be measured during the "integration" activity. However, within the ISO 12207 there are two activities labeled "system integration" and "software integration".

This will make the user of the ISO 9126 confused.

- Many of the ISO 9126 quality metrics referred to processes. However, as known, each process in ISO 12207 contains a number of different activities. Thus, it is more usable for the ISO 9126 users to refer to the activities of the ISO 12207. This can be done using cross-reference numbers from ISO 12207. For example, the cross-reference number 5.3.9 is referring to “primary processes”, “development process”, and “software qualification testing” activity, respectively.

In addition to the mapping in this paper, it is a good idea to investigate where to collect the data for each of the ISO 9126 quality metrics in the ISO 12207 software life cycle processes and activities. This will save time and assure that the data have been completely collected before the measurement of the metrics is performed.

REFERENCES

[1] Abran, A., Al-Qutaish, R. E., and Desharnais, J. M., "Harmonization Issues in the Updating of the ISO Standards on Software Product Quality," *Metrics News Journal*, Vol. 10, No. 2, pp. 35-44, 2005.

[2] Abran, A., Al-Qutaish, R. E., Desharnais, J. M., and Habra, N., "An Information Model for Software Quality Measurement with ISO Standards," *in Proceedings of the International Conference on Software Development (SWDC-REK'05)*, Reykjavik, Iceland, pp. 104-116, 2005.

[3] Albrecht, A. J., "Measuring Application Development Productivity," *in Proceedings of the IBM Application Development Joint SHARE/GUIDE Symposium*, Monetary, California, pp. 83-92, 1979.

[4] Azuma, M., "SQuaRE: The next Generation of ISO/IEC 9126 and 14598 International Standards Series on Software Product Quality," *in Proceedings of the European Software Control and Metrics Conference (ESCOM)*, London, UK, pp. 337-346, 2001.

[5] DeMarco, T., *Controlling System Projects*, McGraw-Hill, New York, USA, 1982.

[6] Fenton, N. E. and Pfleeger, S. L., *Software Metrics: A Rigorous and Practical Approach*, 2nd ed., PWS Publishing Company, Boston, USA, 1997.

[7] Halstead, M. H., *Elements of Software Science*, Elsevier North-Holland, New York, 1977.

[8] Halstead, M. H., "Natural Laws Controlling Algorithm Structure," *ACM SIGPLAN Notices*, Vol. 7, No. 2, pp. 19-26, 1972.

[9] IEEE, *IIEEE/EIA-12207: Information Technology – Software Life Cycle Processes*, the Institute of Electrical and Electronics Engineers, New York, USA, 1996.

[10] IEEE, *Std. 610.12-1990: Standard Glossary of Software Engineering Terminology*, the Institute of Electrical and Electronics Engineers, New York, USA, 1990.

[11] Ince, D. S., Sharp, H., and Woodman, M., *Introduction to Software Project Management and Quality Assurance*, McGraw-Hill, New York, USA, 1993.

[12] ISO/IEC, *ISO/IEC 9126-1: Software Engineering - Product Quality - Part 1: Quality Model*, International Organization for Standardization, Geneva, Switzerland, 2001.

[13] ISO/IEC, *ISO/IEC 12207: Information Technology - Software life cycle processes*, International Organization for Standardization, Geneva, Switzerland, 1995.

[14] ISO/IEC, *ISO/IEC IS 9126, Software Product Evaluation - Quality Characteristics and Guidelines for Their Use*, International Organization for Standardization, Geneva, Switzerland, 1991.

[15] ISO/IEC, *ISO/IEC IS 15939: Software Engineering - Software Measurement Process*, International Organization for Standardization, Geneva, Switzerland, 2002.

[16] ISO/IEC, *ISO/IEC TR 9126-2: Software Engineering - Product Quality - Part 2: External Metrics*, International Organization for Standardization, Geneva, Switzerland, 2003.

[17] ISO/IEC, *ISO/IEC TR 9126-3: Software Engineering - Product Quality - Part 3: Internal Metrics*, International Organization for Standardization, Geneva, Switzerland, 2003.

[18] ISO/IEC, *ISO/IEC TR 9126-4: Software Engineering - Product Quality - Part 4: Quality in Use Metrics*, International Organization for Standardization, Geneva, Switzerland, 2004.

[19] McCabe, T. J., "A Complexity Measure," *IEEE Transaction on Software Engineering*, vol. 2, No. 4, pp. 308-320, 1976.

[20] Pressman, R. S., *Software Engineering: A Practitioner's Approach*, 3rd ed., McGraw-Hill, New York, USA, 1992.

[21] Roberts, F. S., *Measurement Theory, with Applications to Decision Making, Utility, and the Social Sciences*, Addison Wesley, New York, USA, 1979.

[22] Suryn, W., Abran, A., and April, A., "ISO/IEC SQuaRE: The Second Generation of Standards for Software Product Quality," *in Proceedings of the 7th IASTED International Conference on Software Engineering and Applications*, California, USA, 2003.

APPENDIX A
ISO 9126-3 INTERNAL QUALITY METRICS AND WHERE THEY COULD BE APPLIED (MEASURED) IN ISO 12207 PROCESSES AND ACTIVITIES

5-Primary Processes:

5.3 Development:

9) Software qualification testing:

1. *Functional specification stability (1.1[2])*

5.4 Operation:

1. *Functional specification stability (1.1)*

6-Supporting Processes:

6.3 Quality Assurance:

1. *Functional specification stability (1.1)*
2. *Test adequacy (2.1)*

6.4 Verification:

1. *Computational accuracy (1.2)*
2. *Precision (1.2)*
3. *Data exchangeability (data format based) (1.3)*
4. *Interface consistency (protocol) (1.3)*
5. *Functional Compliance (1.5)*
6. *Intersystem standard compliance (1.4)*
7. *Fault detection (2.1)*
8. *Fault removal (2.1)*
9. *Test adequacy (2.1)*
10. *Failure avoidance (2.2)*
11. *Incorrect operation avoidance (2.2)*
12. *Restorability (2.3)*
13. *Restoration Effectiveness (2.3)*
14. *Reliability Compliance (2.4)*
15. *Completeness of description (3.1)*
16. *Demonstration capability (3.1)*
17. *Evident functions (3.1)*
18. *Function understandability (4.1)*
19. *Completeness of user documentation and/or help facility (3.2)*
20. *Input validity checking (3.3)*
21. *User operation cancellability (3.3)*
22. *User operation Undoability (3.3)*
23. *Customizability (3.3)*
24. *Physical accessibility (3.3)*
25. *Operation status monitoring capability (3.3)*
26. *Operational consistency (3.3)*
27. *Message Clarity (3.3)*
28. *Interface element clarity (3.3)*
29. *Operational error recoverability (3.3)*
30. *Attractive interaction (3.4)*
31. *User Interface appearance customizability (3.4)*
32. *Usability Compliance (3.5)*
33. *Response time (4.1)*
34. *Throughput time (4.1)*
35. *Turnaround time (4.1)*
36. *I/O Utilization (4.2)*
37. *I/O Utilization Message Density (4.2)*
38. *Memory utilization (4.2)*
39. *Memory utilization message density (4.2)*
40. *Transmission Utilization (4.2)*
41. *Efficiency Compliance (4.3)*
42. *Activity recording (5.1)*
43. *Readiness of diagnostic function (5.1)*
44. *Change recordability (5.2)*
45. *Change impact (5.3)*
46. *Modification impact localization (5.3)*
47. *Completeness of built-in test (5.4)*
48. *Autonomy of testability (5.4)*
49. *Test progress observability (5.4)*
50. *Maintainability Compliance (5.5)*
51. *Adaptability of data structures (6.1)*
52. *Organizational Environment adaptability (6.1)*
53. *Hardware Environmental Adaptability (H/W, network) (6.1)*
54. *System software Environmental adaptability (OS, concurrent application) (6.1)*
55. *Porting User Friendliness (6.1)*
56. *Continued use of Data (6.3)*
57. *Functional inclusiveness (6.3)*
58. *Available co-existence (6.4)*
59. *Portability Compliance (6.5)*

6.5 Validation:

1. *Functional adequacy (1.1)*
2. *Functional implementation completeness (1.1)*
3. *Functional implementation coverage (1.1)*
4. *Functional specification stability (1.1)*
5. *Access auditability (1.4)*
6. *Access controllability (1.4)*
7. *Data corruption prevention (1.4)*
8. *Data encryption (1.4)*
9. *Failure avoidance (2.2)*
10. *Incorrect operation avoidance (2.2)*
11. *Ease of setup retry (6.2)*
12. *Installation effort (6.2)*
13. *Installation flexibility (6.2)*

[2] This number refers to the <u>characteristic-number.subcharacteristic-number</u> which can be taken from Figure 1. For example, the number 1.1 refers to the "Functionality" characteristic and "Suitability" subcharacteristic which means that this metric (*Functional specification stability*) is used to measure the "Suitability" subcharacteristic in which it is a part of the "Functionality" of any software product, this metrics could be measured during the "Software qualification testing" activity of the "development" process. Throughout these Appendices, the Software Quality Metrics names have been written in *italic* fonts.

6.6 Joint Review:

1. *Functional adequacy (1.1)*
2. *Functional implementation completeness (1.1)*
3. *Functional implementation coverage (1.1)*
4. *Computational accuracy (1.2)*
5. *Precision (1.2)*
6. *Data exchangeability (data format based) (1.3)*
7. *Interface consistency (protocol) (1.3)*
8. *Access auditability (1.4)*
9. *Access controllability (1.4)*
10. *Data corruption prevention (1.4)*
11. *Functional Compliance (1.4)*
12. *Intersystem standard compliance (1.4)*
13. *Fault detection (2.1)*
14. *Fault removal (2.1)*
15. *Failure avoidance (2.2)*
16. *Incorrect operation avoidance (2.2)*
17. *Restorability (2.3)*
18. *Restoration Effectiveness (2.3)*
19. *Reliability Compliance (2.4)*
20. *Completeness of description (3.1)*
21. *Demonstration capability (3.1)*
22. *Evident functions (3.1)*
23. *Function understandability (3.1)*
24. *Completeness of user documentation and/or help facility (3.2)*
25. *Input validity checking (3.3)*
26. *User operation cancellability (3.3)*
27. *User operation Undoability (3.3)*
28. *Customizability (3.3)*
29. *Physical accessibility (3.3)*
30. *Operation status monitoring capability (3.3)*
31. *Operational consistency (3.3)*
32. *Message Clarity (3.3)*
33. *Interface element clarity (3.3)*
34. *Operational error recoverability (3.3)*
35. *Attractive interaction (3.4)*
36. *User Interface appearance customizability (3.4)*
37. *Usability Compliance (3.5)*
38. *Response time (4.1)*
39. *Throughput time (4.1)*
40. *Turnaround time (4.1)*
41. *Efficiency Compliance (4.3)*
42. *Activity recording (5.1)*
43. *Readiness of diagnostic function (5.1)*
44. *Change recordability (5.2)*
45. *Change impact (5.3)*
46. *Modification impact localization (5.3)*
47. *Completeness of built-in test (5.4)*
48. *Autonomy of testability (5.4)*
49. *Test progress observability (5.4)*
50. *Maintainability Compliance (5.5)*
51. *Adaptability of data structures (6.1)*
52. *Organizational Environment adaptability (6.1)*
53. *Hardware Environmental Adaptability (H/W, network) (6.1)*
54. *System software Environmental adaptability (OS, concurrent application) (6.1)*
55. *Porting User Friendliness (6.1)*
56. *Continued use of Data (6.3)*
57. *Functional inclusiveness (6.3)*
58. *Available co-existence (6.4)*
59. *Portability Compliance (6.5)*

6.8 Problem Resolution:

1. *Functional specification stability (1.1)*
2. *Test adequacy (2.1)*
3. *Failure avoidance (2.2)*
4. *Incorrect operation avoidance (2.2)*