ÉTS GÉLOG Software Engineering Research Lab.

SPQ^{MM}:

A Software Product Quality Maturity Model using ISO/IEEE Standards, Metrology and Sigma Concepts

PhD Thesis Defense

By:

Rafa Al Qutaish

Supervised by:

Dr. Alain Abran

Presented in Partial Fulfillment of the Requirements for the Degree of PhD

June 12th, 2007

Agenda

- Problem Statement
- Research Objectives
- Literature Review: A Brief Summary
- Research Methodology
- The SPQ^{MM}: Its Structure & Usage
- Research Contributions
- Conclusions and Future Work
- Publications

Problem Statement

- Each software measure/metric represent a quality attribute.
- In ISO 9126, there is a set of software measures/metrics for each quality characteristic.
  - ✓ Thus, we will have many numbers which could be used to represent a specific characteristic.
  - ✓ This will not be interpreted easily by the decision makers.
  - ✓ Instead, using a single (ranking) number within a quality model would be easier to interpret by decision makers.

Problem Statement

- Such technique is already used in the evaluation of the software processes using capability and/or maturity models.
- There are many capability and maturity models to assess and evaluate a specific set of processes to produce an equivalent maturity level (a single value to reflect the maturity of that set of processes).
  - ✓ Such as CMM, S^{3M}, TMM, etc.
  - ✓ These assessment models are about ‘process’ while we are interested in developing a ‘product quality’ assessment model, named SPQ^{MM}.

Problem Statement

- This SPQ^{MM} will not be based on our own or any individual view of software quality, but on the emerging consensus in the software engineering standards community.
- To build a such quality maturity model, we will use ISO standards, including the ISO 9126 measures.
    - ✓ These measures need to be verified against the metrology concepts.

Research Objectives

- The first objective lies in the building of an understanding of the designs and definitions of the current proposed measures for software product quality to determine their strengths and weaknesses:
  - ✓ From the metrology concepts and ISO 15939 perspectives.
- In particular:
  - ✓ We will verify the ISO 9126 measures against the metrology concepts.
  - ✓ Build an ISO-based information model to address the harmonization issues in the ISO 25020 and ISO 25021 new standards.

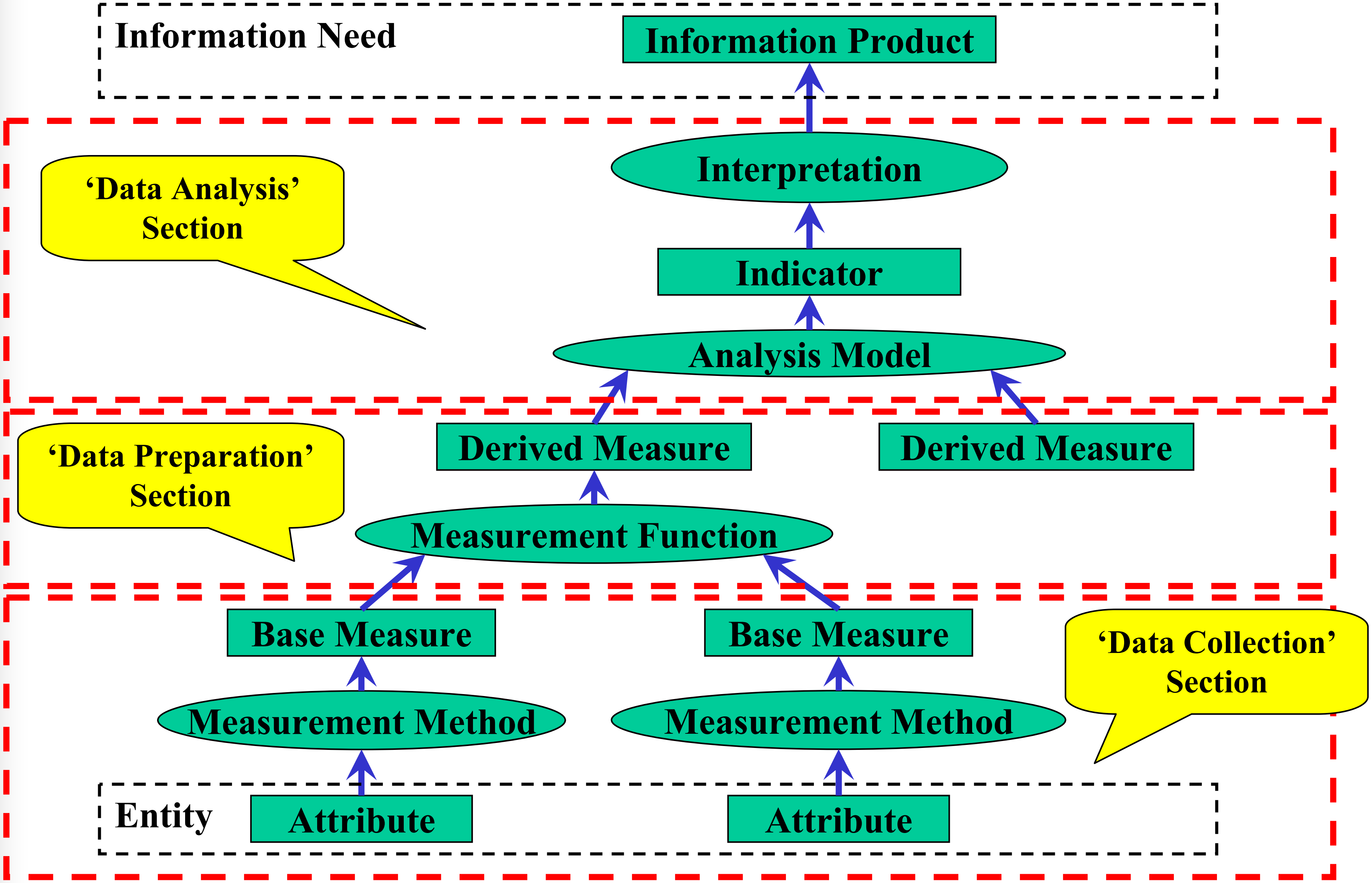
Research Objectives

- While the second objective aims at building a quality maturity model based on software engineering standards. In particular, it will be based on:
  - ✓ Measures based on sound metrological foundations.
  - ✓ Industry consensus on base measures for software product quality.
  - ✓ Industry consensus on software product quality models.
  - ✓ Industry consensus on software integrity levels.

Literature Review: A Brief Summary

- ISO 9126.
- ISO 15939 *(2002)*:
  - ✓ It contains a set of definitions for the commonly used terms in the field of software measurement.
  - ✓ It contains an information model to help in determining what have to be specified during measurement planning, performance and evaluation.
- Next is this information model.

Literature Review: A Brief Summary



Information Need

Information Product

Interpretation

Indicator

Analysis Model

‘Data Analysis’ Section

Derived Measure

Derived Measure

Measurement Function

‘Data Preparation’ Section

Base Measure

Base Measure

Measurement Method

Measurement Method

‘Data Collection’ Section

Entity

Attribute

Attribute

Literature Review: A Brief Summary

- ISO 25000 – SQuaRE:
  - ✓ The main objective of this new series is the coordination and harmonization of its content with ISO 15939.
  - ✓ It will consist of 5 divisions.
  - ✓ Quality measurement division is one of these divisions.
  - ✓ This division will consist of the following documents:
    - **ISO 25020 on measurement quality model and guide *(ISO, 2007)*.**
    - **ISO 25021 on quality measure elements *(ISO, 2007)*.**
    - ISO 25022 on measurement of internal quality.
    - ISO 25023 on measurement of external quality, and
    - ISO 25024 on measurement of quality in-use.

Literature Review: A Brief Summary

- Maturity Models:
  - ✓ Process Maturity Models:
    - CMMi-SW *(SEI, 2002)*.
    - TMM *(Burnstein et al., 1996)*.
    - ISO 15504 *(ISO, 2004)*.
  - ✓ Product Maturity Models:
    - OSMM *(Golden, 2004)*.
    - Software Product Maturity Model *(Nastro, 1997)*.

Literature Review: A Brief Summary

- Current Product Maturity Models Limitations:
  - ✓ OSMM Limitations:
    1. To be used with the open source software products when they are completed, i.e. they are ready for the release.
    2. Mostly useful when an organization or an individual needs to choose between a variety of open source software products.
    3. Not based on any quality model.

Literature Review: A Brief Summary

✓ Software Product Maturity Model Limitations:

1. For an executable software product.
2. Not based on any comprehensive quality model, but only on a small number of product quality characteristics (there are five of them).
3. Designed for the software product itself, rather than the quality of the software product.
4. For each element, there is only one measure.
5. Built to track and report the software development effort during an incremental life-cycle

Literature Review: A Brief Summary

- ISO 15026 *(1998)*:
  - ✓ This standard is used to determine the software integrity level based on:
    - the software product failure consequence (catastrophic, major, severe, or minor).
    - and their frequency of occurrence (frequent, probable, occasional, remote, improbable, or incredible).
  - ✓ The determined integrity levels could be:
    - trivial, low, intermediate, or high.
  - ✓ Example: Failure Consequence is major & its frequency is improbable → the integrity level is intermediate.

Literature Review: A Brief Summary

- IEEE 1012 *(1998)*:
  - ✓ This IEEE standard defines four software integrity levels which vary from high integrity to low integrity.

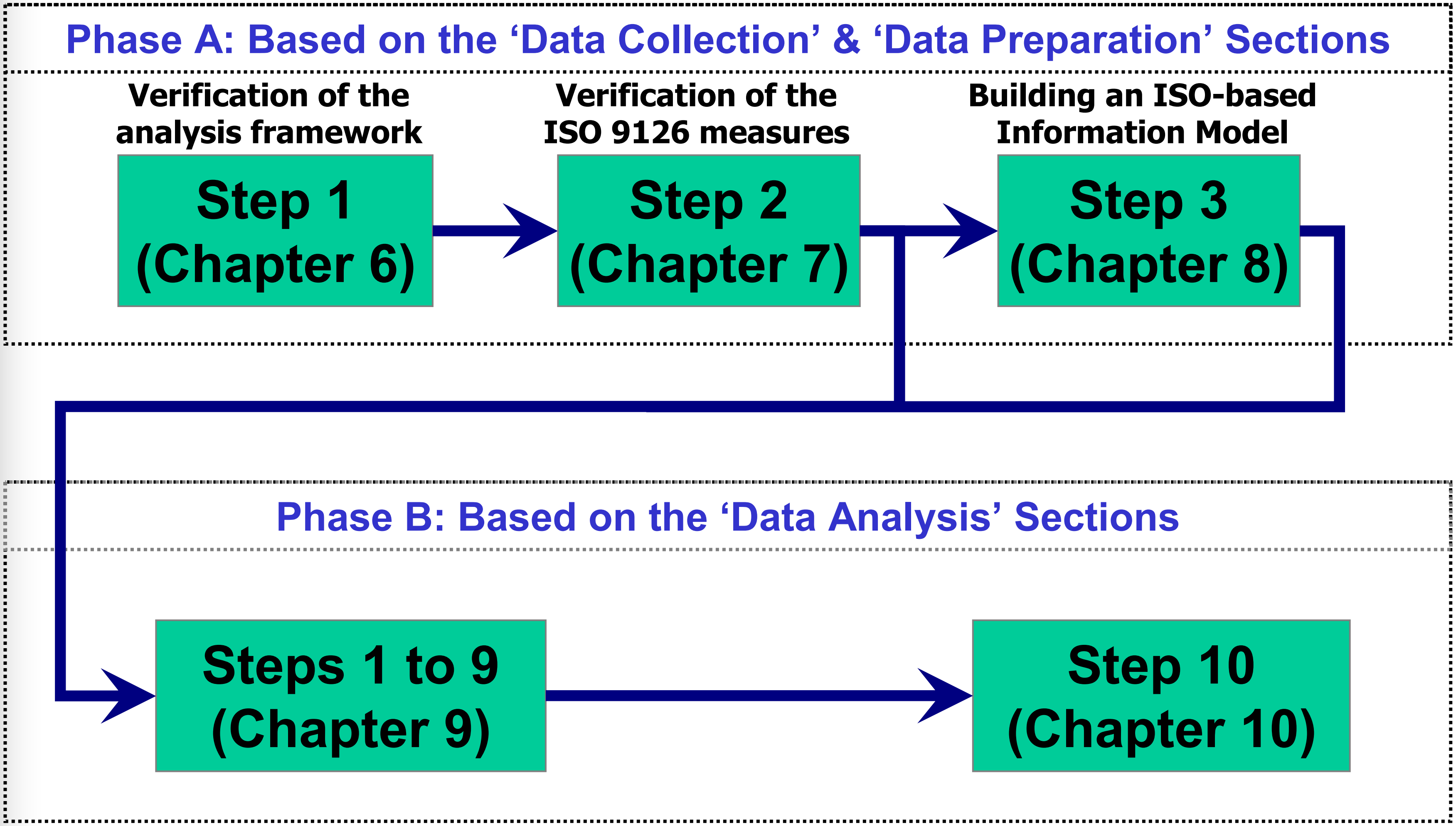
“Assignment of Software Integrity Levels”

consequences	Occurrences			
	Reasonable	Probable	Occasional	Infrequent
Catastrophic	4	4	4 or 3	3
Critical	4	4 or 3	3	2 or 1
Marginal	3	3 or 2	2 or 1	1
Negligible	2	2 or 1	1	1

Literature Review: A Brief Summary

- Six-Sigma Concepts:
  - ✓ It has been designed by Motorola Company in 1986,
  - ✓ It defined as a measure of the defects to improve the quality *(Motorola, 1986)*.
  - ✓ Six-sigma means 99.99966% quality level, this means that we could have only 3.4 DPMO.
  - ✓ In six-sigma, the sigma value has been shifted by 1.5 sigma, (to be 6 instead of 4.5).

Research Methodology



Phase A: Based on the ‘Data Collection’ & ‘Data Preparation’ Sections

Verification of the analysis framework

Step 1 (Chapter 6)

Verification of the ISO 9126 measures

Step 2 (Chapter 7)

Building an ISO-based Information Model

Step 3 (Chapter 8)

Phase B: Based on the ‘Data Analysis’ Sections

Steps 1 to 9 (Chapter 9)

Step 10 (Chapter 10)

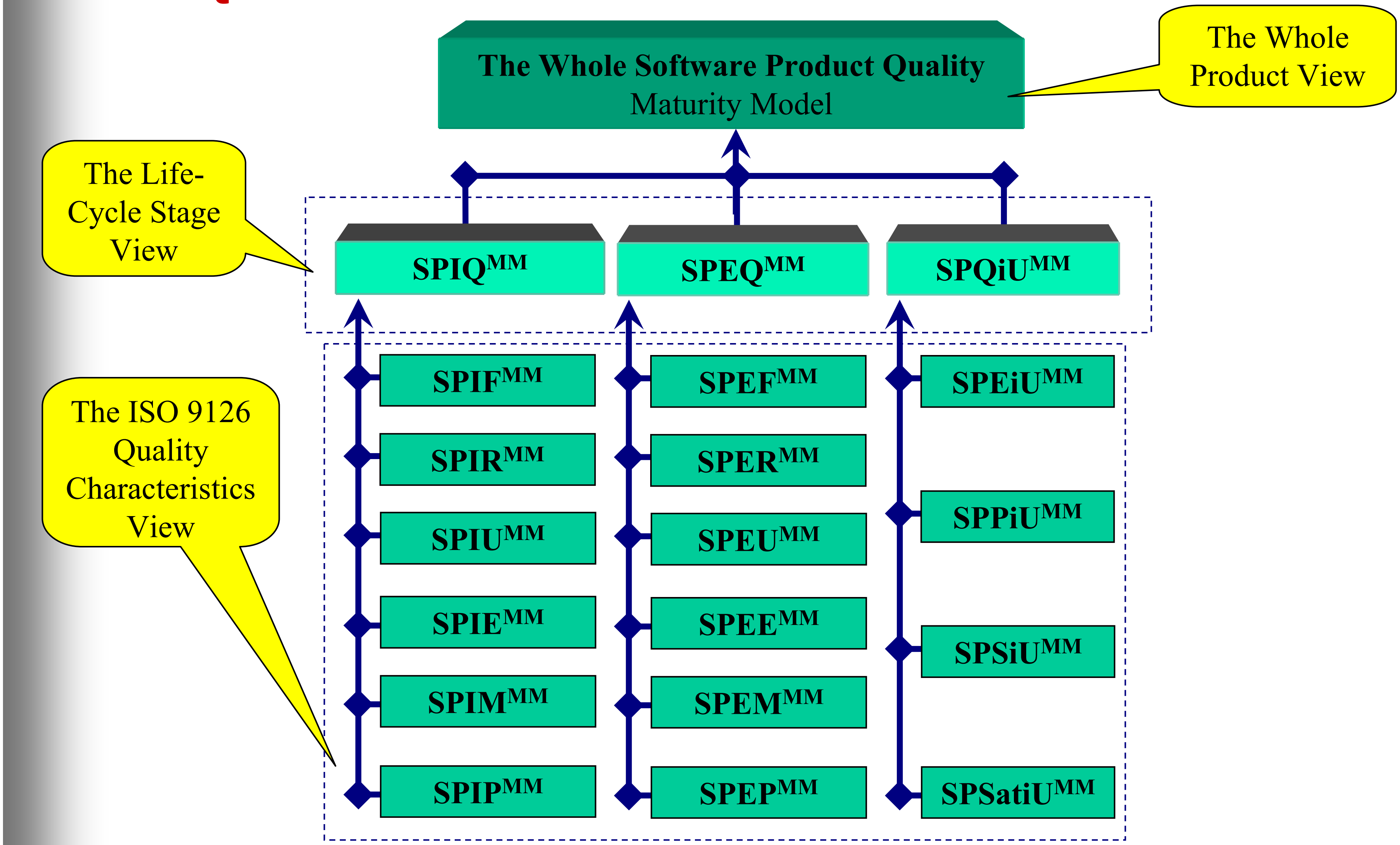
Literature Review: A Brief Summary

- Phase-B:
  1) Reviewing the literature.
  2) Identifying the quality model to be used.
  3) Identifying the contents of the QMM.
  4) Constructing a set of five maturity levels.
  5) Customizing the software product failure consequences.
  6) Customizing the software integrity levels to six levels.
  7) Review the sigma and sigma shift concepts.
  8) Mapping the sigma and sigma shift concepts to the quality levels.
  9) Mapping the sigma ranges to the maturity levels.
  10) Draw up a detailed formulas and procedures to be followed to get a quality maturity level.

The SPQ^{MM}: Its Structure & Usage

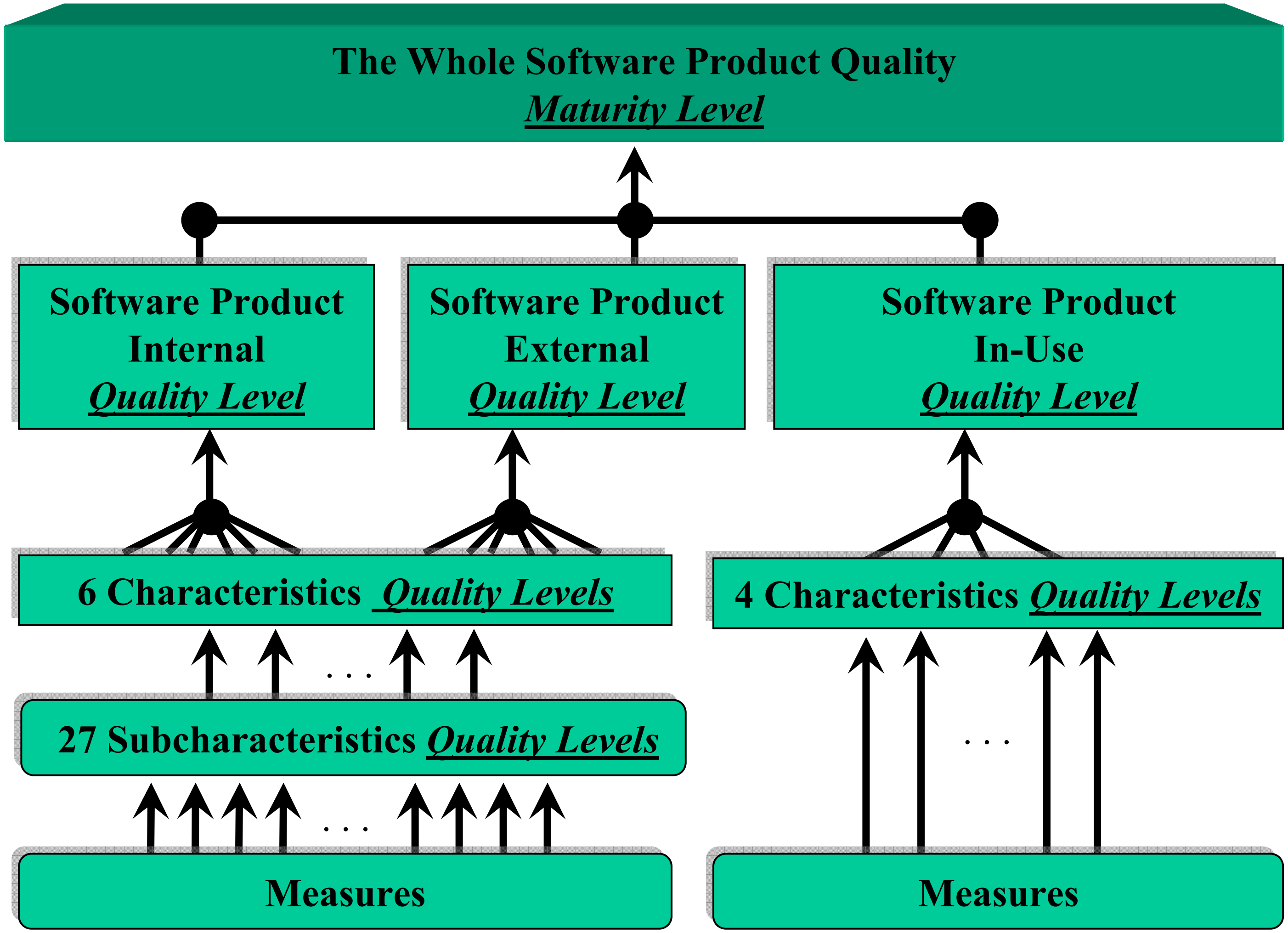
- The SPQ^{MM} can be used from the following viewpoints:
  - ✓ The Whole Software Product.
  - ✓ The Software Product Life-Cycle Stage.
  - ✓ The Software Product Characteristic.
- From the software product life-cycle stage, we can use the proposed model to assess the maturity level for:
  - ✓ Internal Software Product Quality.
  - ✓ External Software Product Quality.
  - ✓ Software Product Quality in-Use.

The SPQ^{MM} Structure



The different viewpoints of the SPQ^{MM}

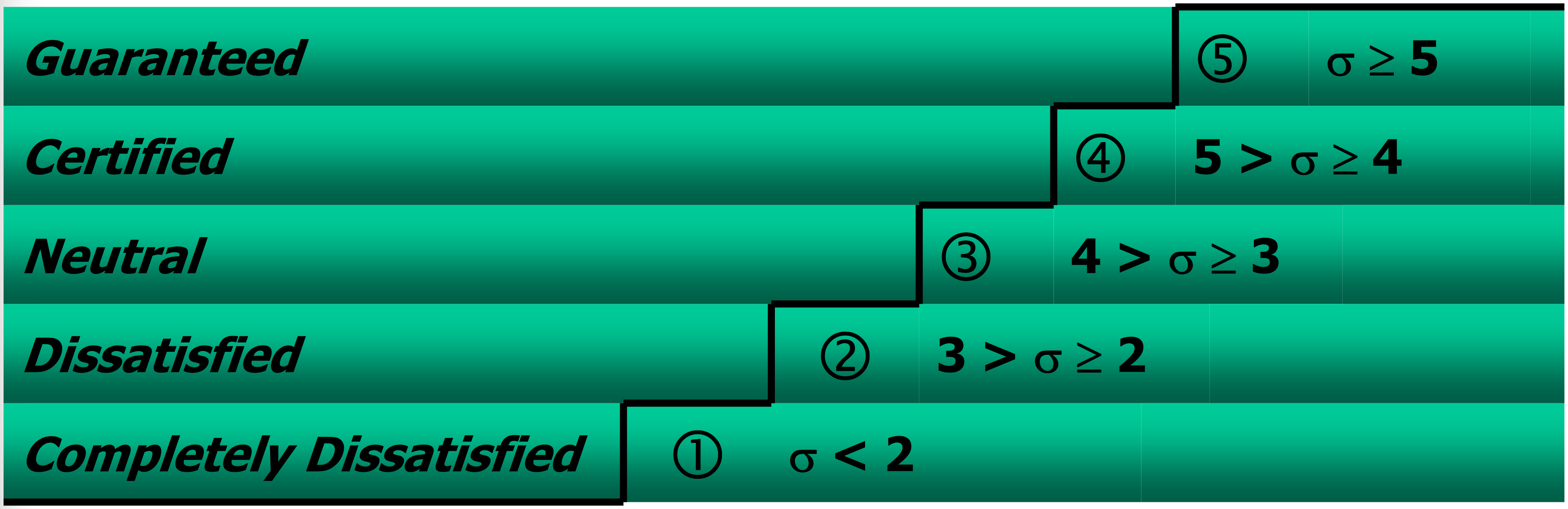
The SPQ^{MM} Structure


The Whole Software Product Quality
Maturity Level
Software Product Internal Quality Level
Software Product External Quality Level
Software Product In-Use Quality Level
6 Characteristics Quality Levels
4 Characteristics Quality Levels
27 Subcharacteristics Quality Levels
. . .
Measures
Measures

The components of the quality maturity level for the whole software product

The SPQ^{MM} Structure

- The following five quality maturity levels have been identified from the observation of general industry practices outside the software engineering domain.
- This maturity scale can be applied in turn to the three different viewpoints.



Guaranteed	⑤	$\sigma \geq 5$
Certified	④	$5 > \sigma \geq 4$
Neutral	③	$4 > \sigma \geq 3$
Dissatisfied	②	$3 > \sigma \geq 2$
Completely Dissatisfied	①	$\sigma < 2$

The quality maturity levels

The SPQ^{MM} Structure

Software Integrity Levels and Risk Classes					
5	4	3	2	1	0
Very High	High	Intermediate	Low	Trivial	None

Quality Levels (QL) for each Sigma Shift						
Zero Sigma Shift	1.5 Sigma Shift	2.0 Sigma Shift	2.5 Sigma Shift	3.0 Sigma Shift	3.5 Sigma Shift	Assigned Sigma Ranges
QL≥99.99997%	QL≥99.976%	QL≥99.865%	QL≥99.379%	QL≥97.724%	QL≥93.319%	σ ≥ 5
QL<99.99997% and QL≥99.996%	QL<99.976% and QL≥99.379%	QL<99.865 and QL≥97.724%	QL<99.379% and QL≥93.319%	QL<97.724% and QL≥84.134%	QL<93.319% and QL≥69.146%	5 > σ ≥ 4
QL<99.996% and QL≥99.865%	QL<99.379% and QL≥93.319%	QL<97.724% and QL≥84.134%	QL<93.319% and QL≥69.146%	QL<84.134% and QL≥50%	QL<69.146% and QL≥30.853%	4 > σ ≥ 3
QL<99.865% and QL≥97.724%	QL< 93.319% and QL≥69.146%	QL<84.134% and QL≥50%	QL<69.146% and QL≥30.853%	QL<50% and QL≥15.865%	QL<30.853% and QL≥6.680%	3 > σ ≥ 2
QL<97.724%	QL<69.146%	QL<50%	QL<30.853%	QL<15.865%	QL<6.680%	σ < 2

The sigma ranges based on the Quality Level and the Software Integrity Level

The SPQ^{MM} Structure

An Example

Quality Level	Original Sigma Value (OSV)	Integrity Level	Sigma Shift Value (SSV)	Shifted Sigma Value (OSV+SSV)	Corresponding Maturity Level
99%	2.32σ	5	0.0	(2.32+0.0) = 2.32σ	2
		4	1.5	(2.32+1.5) = 3.82σ	3
		3	2.0	(2.32+2.0) = 4.32σ	4
		2	2.5	(2.32+2.5) = 4.82σ	4
		1	3.0	(2.32+3.0) = 5.32σ	5
		0	3.5	(2.32+3.5) = 5.82σ	5

Determining the quality maturity levels using the SPQ^{MM}

- The following steps should be followed to get the quality maturity level of a software product:

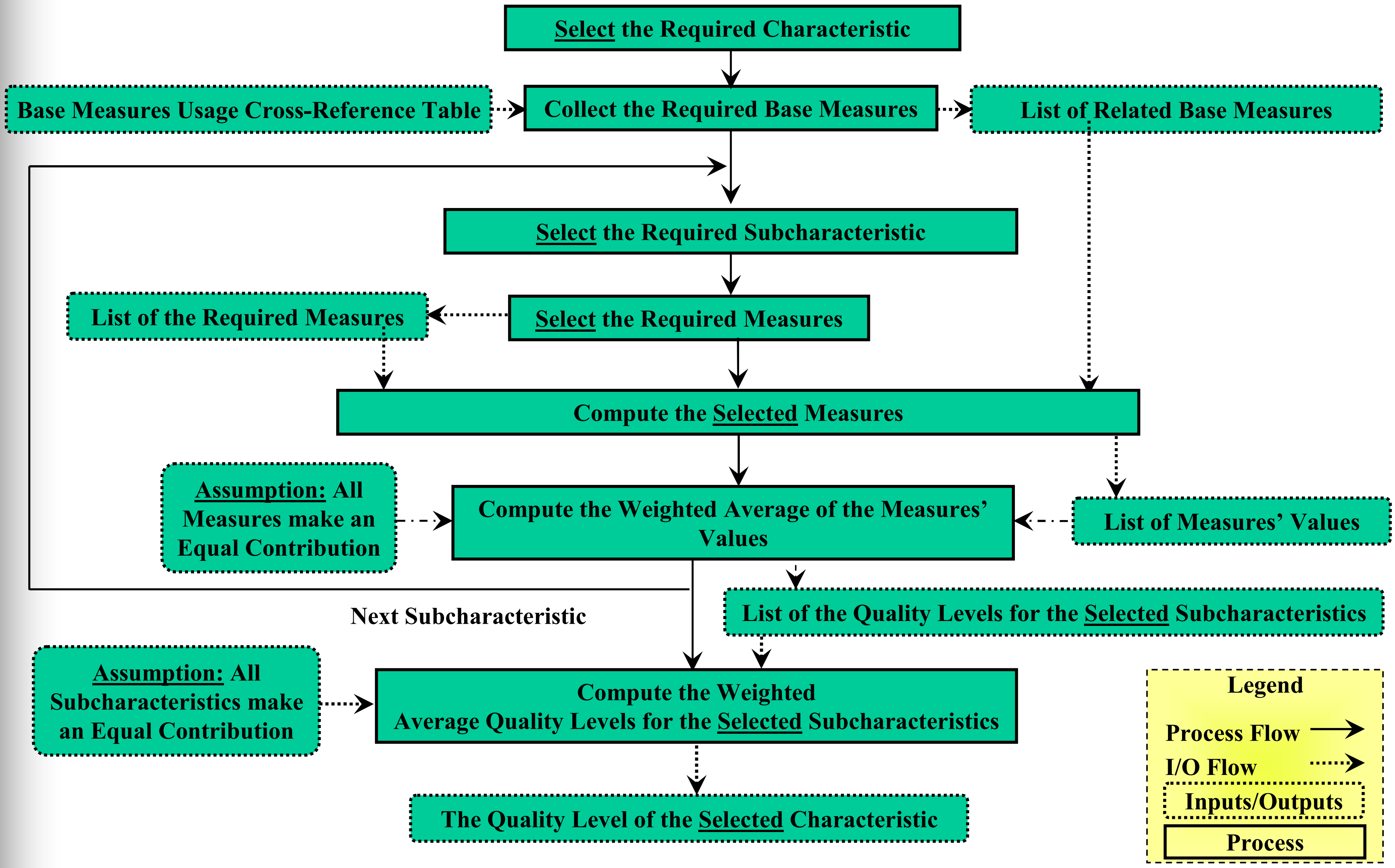
1. Determine the software integrity level.

Consequence	Occurrence					
	Frequent	Probable	Occasional	Remote	Improbable	Incredible
Catastrophic	5	5	5	4	3	2
Critical	5	5	4	3	2	1
Severe	4	4	3	2	1	1
Marginal	3	3	2	2	0	0
Minor	2	2	1	1	0	0
None	0	0	0	0	0	0

Determining the quality maturity levels using the SPQ^{MM}

2. Select the required characteristics, subcharacteristics, and derived measures.
3. Identify the required base measures for each of the selected characteristics / subcharacteristics.
4. Compute the Quality Levels of the selected software product quality characteristics.
5. Identify the sigma range.
6. Identify the maturity level.

Determining the quality maturity levels using the SPQ^{MM}



Select the Required Characteristic

Base Measures Usage Cross-Reference Table

Collect the Required Base Measures

List of Related Base Measures

Select the Required Subcharacteristic

List of the Required Measures

Select the Required Measures

Compute the Selected Measures

Assumption: All Measures make an Equal Contribution

Compute the Weighted Average of the Measures' Values

List of Measures' Values

Next Subcharacteristic

List of the Quality Levels for the Selected Subcharacteristics

Assumption: All Subcharacteristics make an Equal Contribution

Compute the Weighted Average Quality Levels for the Selected Subcharacteristics

The Quality Level of the Selected Characteristic

Legend

Process Flow →

I/O Flow ⋯→

Inputs/Outputs

Process

Research Contributions

1. Verification of the applicability of using metrology concepts to software measures to investigate their design and definition.
2. Identification of some of the harmonization issues arising with the addition of new documents like the ISO 25020 and ISO 25021.
3. Identification of a list of base measures needed to evaluate the ISO 9126 derived measures. In addition, a cross-reference list between the base measure and the related characteristics / subcharacteristics has been built.

Research Contributions

4. Applying the sigma concepts to the measured quality levels of the software product, by mapping the quality level to the corresponding sigma value.
5. Building of a maturity model to assess the maturity level of the software product quality from different views (the characteristic view, the life cycle stage view, and the whole software product view) based on a set of ISO and IEEE standards.

Conclusion and Future Work

- The design of a software product maturity model to assess the quality of a software product, therefore, represented a new challenge in software engineering. In this thesis, we presented a product quality assessment model based on some ISO/IEEE standards and sigma concepts. Specifically, we discussed the structure of the quality maturity model from the following three distinct points of view:
  - ✓ The whole software product.
  - ✓ The software product life cycle stage.
  - ✓ The software product characteristic

Conclusion and Future Work

- The proposed SPQ^{MM} can be used to determine the maturity of the quality of a software product. Specifically, it can be used to:
  - ✓ Certify a quality maturity level for a new software product.
  - ✓ Benchmark two existing software products.
  - ✓ Assess the quality of the software product during the development life cycle stages.
  - ✓ Assess the maturity of the internal quality of a software product.
  - ✓ Compare the maturity levels of the life cycle stages quality.

Conclusion and Future Work

- Limitations:
  - ✓ Limited to the ISO 9126 and SQuaRE quality models and their set of measures which have been analysed.
  - ✓ The results yielded by the SPQ^{MM} are initially based on the assumption of the equal weights of all measures, all characteristics and all subcharacteristics.
    - To avoid making this assumption, an organization can apply, for instance, the PCA or AHP to a large set of historical data to find a corresponding weight for each measure, characteristic or subcharacteristic.

Conclusion and Future Work

- Future Work:
  - ✓ The quality of the software product also depends on other elements rather than the software product itself. Such elements are: <u>documentation</u>, <u>training</u>, <u>supporting</u> and <u>integration</u>, Therefore, based on these elements, the following quality maturity models need to be developed:
    - SPDQ^{MM}, SPIntegQ^{MM}, SPSQ^{MM} and SPTQ^{MM}.
  - ✓ These quality maturity models could be integrated to the proposed SPQ^{MM}.
  - ✓ They could be automated, at least partially, by means of web-based tools to make the assessment procedure of the quality maturity levels faster and easier.

Publications

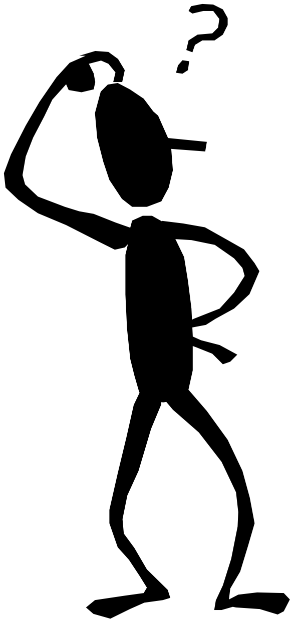
- The following are the outcomes of our research:

  1. "An Analysis of the Design and Definitions of Halstead's Metrics", in the 15th International workshop on Software Measurement - IWSM'2005, Montreal, Canada, 12-14 Sept. 2005, p. 337-352.
  2. "Harmonization Issues in the Updating of the ISO Standards on Software Product Quality", Metrics News Journal, vol. 10, no 2, 2005, p. 35-44.
  3. "An Information Model for Software Quality Measurement with ISO Standards", International Conference on Software Development – SWDC-REK, Iceland, 27 May-1 Jun. 2005, p. 104-116.
  4. "Analysis of the ISO 9126 on Software Product Quality Evaluation from the Metrology and ISO 15939 Perspectives", WSEAS Transactions on Computers, vol. 5, no 11, 2006, p. 2778-2786.

Publications

5. "Investigation of the Metrology Concepts within ISO 9126 on Software Product Quality Evaluation", in the 10th WSEAS International Conference on Computers - ICComp'2006, Greece, 13-15 Jul. 2006, p. 864-872.
6. "ISO-Based Models to Measure Software Product Quality", *Accepted* as a Chapter in "Software Quality Measurement" – to be Edited by G. Vijay, FCAI - the Institute of Chartered Financial Analysts of India, Hyderabad, India, 2007.
7. "Assessment of Software Product Quality: Determining the Maturity Levels", *Submitted* to the ASQ Software Quality Professional Journal, American Society of Quality, USA, 2007.

Merci – Thanks – Grazie – شكراً



Q & A