An integrated life cycle quality model for general public market software products

Witold Suryn[1], Alain Abran[2], Claude Laporte[3]

[1] Département de génie électrique, École de technologie supérieure
1100, rue Notre-Dame Ouest
Montréal (Québec),
Canada H3C 1K3
wsuryn@ele.etsmtl.ca

[2] Département de génie électrique, École de technologie supérieure
1100, rue Notre-Dame Ouest
Montréal (Québec),
Canada H3C 1K3
aabran@ele.etsmtl.ca

[3] Département de génie électrique, École de technologie supérieure
1100, rue Notre-Dame Ouest
Montréal (Québec),
Canada H3C 1K3
claporte@ele.etsmtl.ca

Abstract

The business value of the software product results from its ultimate quality seen by both acquirers and end users. An integrated life cycle quality model, further called *complement model* for software product quality combines high level quality view of TL9000 Handbook and detailed view from ISO/IEC 1926 in the process of defining, measuring, evaluating and finally achieving appropriate quality of user-centered software product.
This paper presents how the use of TL9000 product operational (in-the-field) quality measures can bring benefits to setting up, measuring and evaluating the quality of the software product being developed, through its entire life cycle. The process of building quality into a software product is discussed and illustrated by TL9000-ISO complement model as well as by application process walk-through.

Proceedings of Software Quality Management XII Conference (SQM 2004)
ISBN 1-902505-56-5
British Computer Society (**BCS**) 2004

1. Complement model for quality requirements identification

Software product quality is ultimately evaluated when the product is used in its operational environment i.e. when the user validates on a daily basis the totality of the product behavior as the fulfillment of his business needs.
From the business standpoint, needs are addressed by the implementation of functional and non-functional requirements, quality of use of the product and its operational quality when being used in the field.
For the users a software product often corresponds to a black box that must effectively support their business processes. In consequence of this natural approach business needs become a driving force of quality software product development. This in turn requires that operational quality and satisfaction of using a software product set the framework for driving software product development effort: at the beginning of the development process to elicit business-related software product quality requirements, while at the end - to allow a rigorous evaluation. This *business view of quality* is illustrated in Fig.1 [17].

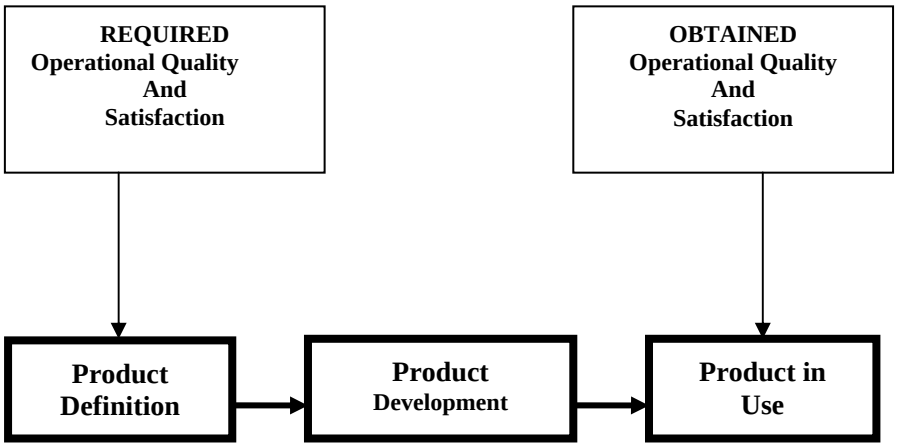

Fig. 1 Business View of software product quality

Identifying quality requirements that can be elicited, formalized and further evaluated in each phase of full software product lifecycle thus becomes a crucial task in the process of building a high quality software product.
The QUEST Forum has defined the TL9000 standards [1, 2] for the set of initial requirements for operational quality as well as for reporting on implemented quality once the software product has been developed and deployed in the field.
The TL 9000 Handbooks (TL 9000 Quality System Requirements [1] and the TL 9000 Quality System Measurements [2]) are designed specifically for the telecommunications industry to document the industry's quality system requirements and measures. The TL 9000 Quality System Requirements Handbook establishes a common set of quality system requirements for suppliers of

telecommunications products: hardware, software or services. The requirements are built upon existing industry standards, including ISO 9001. The TL 9000 Quality System Measures Handbook defines a minimum set of performance measures and cost and quality indicators to measure progress and evaluate results of quality system implementation.
TL 9000, applicability in software product lifecycle is illustrated in Fig.2.

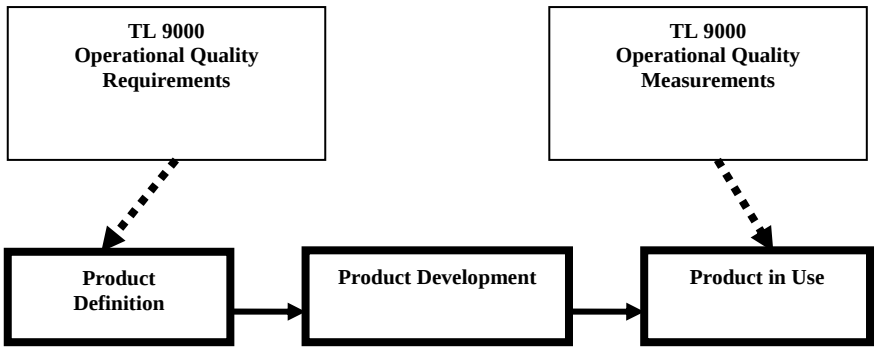

Fig.2 Applicability of TL9000 standards in software product lifecycle

In parallel the ISO/IEC Subcommittee 7 (SC7) on system and software engineering has developed set of quality standards for the full development process. These standards take the initial quality requirements into account during each of the development phases, allowing the quality planning, its design, monitoring and control.
Software product quality can be evaluated by measuring internal attributes (typically static measures of intermediate products), or by measuring external attributes (typically by measuring the behaviour of the code when executed), or by measuring quality in use attributes. The objective is for the product to have the required effect in a particular context of use. To produce these effects measurement and evaluation of the quality of software product has to be present during all its lifecycle (Fig. 3).

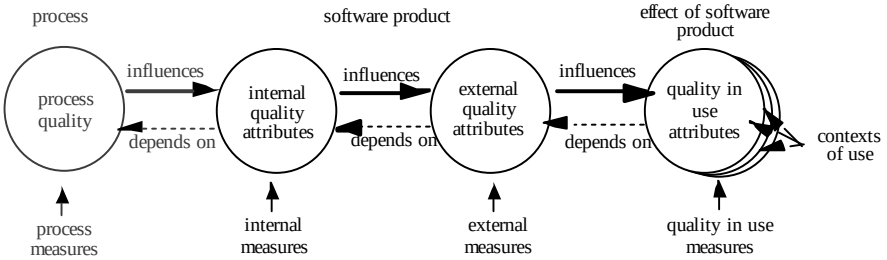

Fig.3 Quality in lifecycle

ISBN 1-902505-56-5

Moreover, the proper quality measurement and evaluation methodologies have to be present *and* applied. ISO/IEC 9126 series of standards [3, 4, 5, 6] offers both broadly recognized quality models (Fig.4) and appropriate measurements together with scales and measurement methods. ISO/IEC 14598 series of standards [7, 8, 9, 10, 11, 12] is a complementary set offering the support for software quality evaluation processes.

Figure 4 presents how these ISO/IEC standards integrate to the TL9000.
The practical use of these two combined sets of standards requires however a much more detailed view and, in order to define, plan and implement the quality, the precise identification of applicable standards and their particular documents for each phase of software development process. Moreover, for the utilizable, understandable and practically applicable mapping of the discussed standards the proper model of software development has to be chosen.
The number of existing software life cycle models [15, 16] indicates the differentiation of the approaches to software development so, in consequence, to software quality. To avoid ambiguity the authors have decided to base their Complement Model on the system life cycle model that has been recognized equally by the industry and academia. For the purposes of this paper the chosen system life cycle model is the one recommended in ISO/IEC 15288 – 2002 [13].

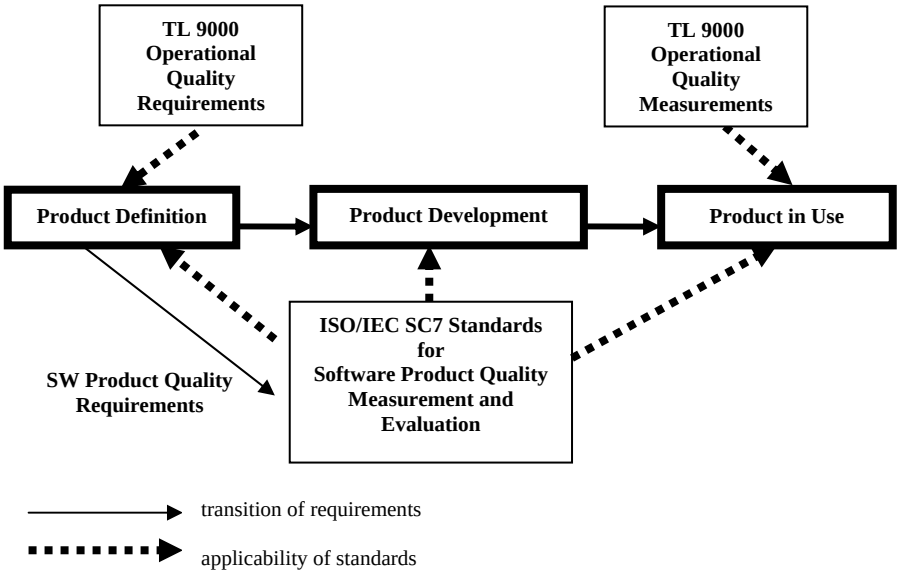

Fig.4. Integration between TL9000 and ISO/IEC SC7 standards

The ISO/IEC SC7 standards are complementary to the QuEST Forum's TL 9000 Handbooks; they address throughout the software development life cycle quality requirements, measurement and evaluation on the level of functional and non-functional requirements, creating this way the continuity between *operational*

quality requirements and *operational quality measurements* offered by TL 9000. The ISO/IEC standards being further discussed and applied are:

- ISO/IEC 9126 series - Software and System Engineering – Software Product Quality Metrics. 1999-2002 [3, 4, 5, 6]
- ISO/IEC 14598 series – Software and System Engineering – Software Product Evaluation. 1996-2002 [7, 8, 9, 10, 11, 12]

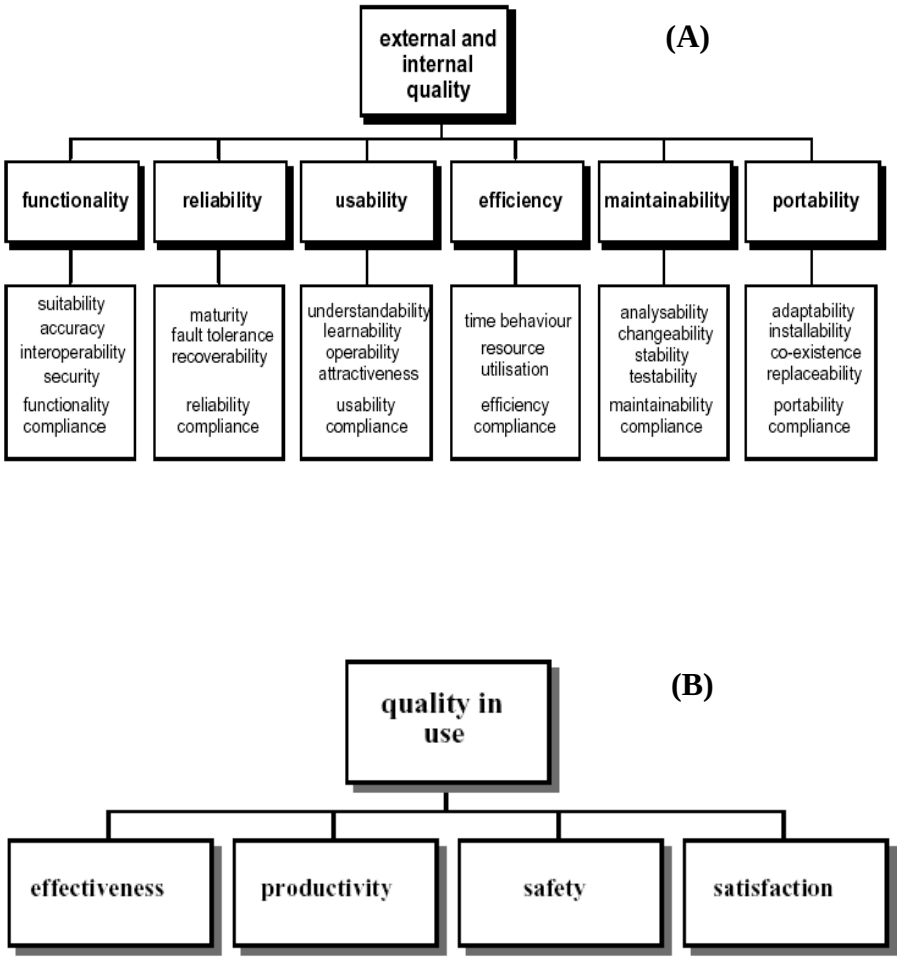

Fig.5 ISO/IEC 9126 quality model: external and internal quality (A) and quality in use (B)

Fig.6 presents the TL9000-ISO detailed **complement model** which aims to help search and identify appropriate quality requirements starting with the phase of business-related quality attributes of non-existent product and finishing by quality measurement of the product used in the field

The following part of this paper will discuss the complete process of defining quality requirements and quality attributes, through the choice of quality measures up to the final product quality evaluation.

2. Process

The proposed process will be analyzed with the assumption that the software product does not exist yet. Additionally, it will be taken as an auxiliary hypothesis that the market demand for this particular product is not mature as well.
For the simplicity of the application of the presented process the steps to follow correspond directly to software life cycle phases used as the backbone in complement model in Fig.6.

Discovery Phase. This phase places the whole process on Business System Environment level, where three sets of requirements have to be identified and defined:

- Functional and non-functional requirements of the product (out of the scope of this paper)
- Operational quality requirements, and
- Quality in Use requirements

It is important to note here, that according to the model of quality in software life cycle defined in ISO/IEC 9126-1 [3] the requirements of Quality in Use contribute to specifying External Quality requirements, which in turn contribute to specifying Internal Quality requirements. This sub-process clearly indicates that the attributes of Quality in Use have the direct impact on technical and technological decisions that (will) have to be taken when the development process starts. Continuing this line of the analysis the person responsible for defining new software product quality attributes will have to analyze Quality in Use characteristics [6], identify applicable measures and assign target values for each of them The ISO standard to be applied to complete this task is ISO/IEC 9126 – Part 4: Quality in Use Metrics [6]. The characteristics to be analyzed are:

- effectiveness
- productivity
- safety, and
- satisfaction

It is also strongly recommended to refer in this phase to ISO/IEC 9126 – Part1: Quality Model [3], to reach a comfortable level of familiarization with quality models, their characteristics and subcharacteristics.
Quality in Use requirements help define success criteria of the new software product however alone they will not assure the product's long term success in the market. Such a success is achieved when quality in use comes together with, among the others, fulfilled operational quality requirements.
Again, continuing this line of the analysis the person responsible for defining new software product quality attributes will have to analyze operational quality requirements, identify applicable measures and assign target values for each of them.

TL 9000 – Quality Management System Measurement Handbook [2] identifies four (4) categories of requirements and/or measurements applicable to software products:

- common measurements – referring to number of problems reported, response time, overdue problem responsiveness and on-time delivery
- hardware and software measurements – referring to system outage
- software measurements – referring to software installation and maintenance
- service measurement – referring to service quality

The final set of quality requirements and their targeted values, comprising of both operational quality and Quality in Use requirements will then become the major milestone and contributor in the definition of functional and non functional requirements of the future software product. From this point on the process of setting up, measuring and evaluating software product quality may appear fairly classic but due to all the activities executed in this phase the *user perception of the software product quality* is already "sewn" into the overall (functional and non-functional) definition of the new software product.

Requirements Analysis Phase. As this phase produces the translation of requirements (both quality and functional) from stakeholders' perspective into technical and technological terms the level of abstraction changes from "business" to "IT system" and the environment changes to Information System Environment. In this environment the applicable quality requirements define external and internal quality attributes of software product to be developed.
The ISO standards applied in this phase are:

- ISO/IEC 9126 – Part 2: External Quality Metrics [4], and
- ISO/IEC 9126 – Part 3: Internal Quality Metrics [5]

It has to be stressed here, that the attributes of both external and internal quality being defined in this phase make direct descendants of quality requirements previously set up in the Discovery phase, so the critic rule of traceability in software engineering is being conserved.

Implementation Phase. Software coding, unit, integration and alpha testing make the usual contents of this phase. In other words, this phase as the first in the whole life cycle creates *a product* that can be measured and evaluated (prototypes created in previous phases are not being recognized as *a product* in this paper). It is true, that the created product is intermediate and changes many times before becoming a ready-to-use solution, but exactly due to this fact it is critical to measure and evaluate its quality. The product is now in Software Development Environment and every iteration with measured and evaluated quality produces indications yielding further improvements.

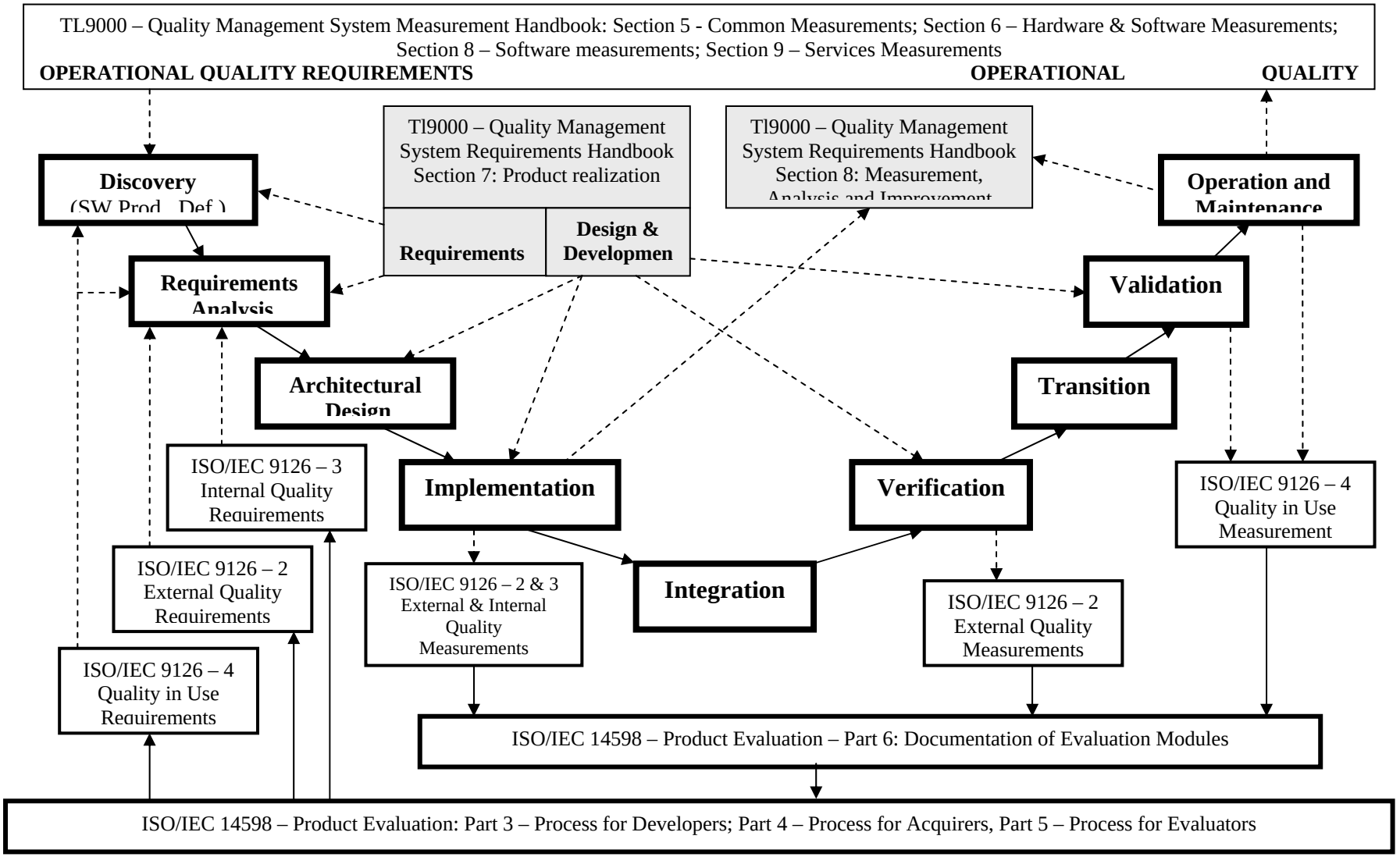

Fig.6. Complement model for quality definition and evaluation processes

Proceedings of Software Quality Management XII Conference (SQM 2004)
ISBN 1-902505-56-5
British Computer Society (**BCS**) 2004

This process is very well supported by appropriate standardization instruments that allow measurement, documentation and evaluation of Internal Quality (and, if needed, External Quality) attributes defined in Requirements Analysis phase. The recommended procedure consists of:

- Measurements of Internal and External Quality attributes. Documents to be used: ISO/IEC 9126 – Part 2 and 3 [4, 5]
- Documentation of measurements. Document to be used: ISO/IEC 14598 – Part 6 [12]
- Evaluation of the quality of the intermediate products. Documents to be used, depending on the position of the evaluating entity: ISO/IEC 14598 – Part 3: Process for Developers [9], Part 4: Process for Acquirers [10] or Part 5: Process for Evaluators

The results of measurements of Internal and External Quality attributes are compared with target values assigned to them in previous phases and the conclusions are feed backed to development teams as the corrective measures of improvement.

Verification Phase makes a perfect opportunity for evaluation of ready-to-use product quality in its Information System Environment. In other words, the product is integrated (supposedly *complete*) and should correspond to stakeholder's functional and non-functional requirements. This explicitly means that External Quality requirements have to be satisfied in this phase. The process of the evaluation of External Quality requires a similar procedure as Internal Quality evaluation and is being similarly well supported by standardization instruments. The recommended procedure consists of:

- Measurements of External Quality attributes. Document to be used: ISO/IEC 9126 – Part 2 [4]
- Documentation of measurements. Document to be used: ISO/IEC 14598 – Part 6 [12]
- Evaluation of the quality of the product. Documents to be used, depending on the position of the evaluating entity: ISO/IEC 14598 – Part 3: Process for Developers [9], Part 4: Process for Acquirers [10] or Part 5: Process for Evaluators

The results of measurements of External Quality attributes are compared with target values assigned to them in previous phases. The resulting conclusions may be feed backed as the corrective measures of improvement. The feedback may be directed to different phases of the process depending on the level of the severity of discrepancies between required and obtained External Quality.

Validation Phase moves the software product back to the business level i.e. to Business System Environment where satisfying *business requirements* is the most important and ultimate task of the product. The system returns to its "black box" status (as it started in Discovery phase) where the user validates its usefulness for conducting his business, usually with no regard to technicalities.
This again explicitly means that Quality in Use requirements have to be satisfied "here and now". The process of the evaluation of Quality in Use requires the same

procedure as External Quality evaluation and is being equally well supported by standardization instruments. The recommended procedure consists of:

- Measurements of Quality in Use attributes. Document to be used: ISO/IEC 9126 – Part 4 [4]
- Documentation of measurements. Document to be used: ISO/IEC 14598 – Part 6 [12]
- Evaluation of the quality of the product. Documents to be used, depending on the position of the evaluating entity: ISO/IEC 14598 – Part 3: Process for Developers [9], Part 4: Process for Acquirers [10] or Part 5: Process for Evaluators

The results of measurements of Quality in Use attributes are compared with target values assigned to them in previous phases. The resulting conclusions may be feed backed as the corrective measures of improvement. The feedback may be directed to different phases of the process depending on the level of the severity of discrepancies between required and obtained Quality in Use.

Operation and Maintenance Phase is recognized theoretically as the consecutive phase in the development process, while in fact this phase lives by its own rules. The most important aspects distinguishing Operation and Maintenance phase from all the previous phases are *time* and *control level*. The duration of Operation and Maintenance cannot be planned (even if there are attempts to forecast this period) and the phase itself is to a great extent events-driven. Last but not least is the environment, Business System Environment that practically excludes any long term *active* experiments or measurements. But *passive* measurements are exactly what is needed in this phase.
Operational quality measurements require data, which to be representative have to be collected over relatively long period of time. In this case the procedure uses TL 9000 Quality Management System Measurements Handbook [2] in order to perform needed calculations and evaluate obtained operational quality. Depending on the area of measurement and evaluation the results can be used immediately, f.e. for improvements of the service quality, or in next round of product development, if the evaluation indicates weaknesses of the product being in the field.

Applying measurements and evaluation of Quality in Use in Operation and Maintenance phase proves it very sense especially in cases of large and complicated software products. Validation phase, where Quality in Use is being measured and evaluated for the first time makes a relatively short period with limited exploration opportunities (as f.e. limited number of users) while Operation and Maintenance phase offers natural circumstances with unlimited time and exhaustive conditions of exploitation. Thus the important question in this case would be “how long?”
The structure “product-user” usually reaches its level of stability after few months of exploitation so it makes sense to conduct Quality in Use measurements and evaluation through the similar period. Further measurement efforts would not most probably deliver substantial data due to the “routinization” of interaction between the user and the product. The measurement and evaluation procedure for Quality in Use in Operation and Maintenance phase would be the same as proposed for

Validation phase. The evaluation results can be useful both immediately (*evolutional* role of maintenance process) and in long term perspective, when new product or its release will be considered.

3. Applicability considerations

- The process discussed in part 2 of this paper omits three phases of the development process present in the complement model from Fig.3. These phases are: Architectural Design phase, Integration phase and Transition phase. The reasons for not considering these phases come from the fact that ISO/IEC standards address them poorly or do not address them at all. So to discuss these phases the authors would have to use as the support the material from specific application areas (f.e. industrial) therefore risking arriving to non-broadly verified conclusions.

- The discussion of the cycle of identification, definition, measurement and evaluation of software product quality presented in part 2 of this paper takes as *the* hypothesis reader's familiarity with basic concepts present in ISO/IEC 14598 and ISO/IEC 9126 series. As the presentation and explanation of these concepts is out of the scope of this paper it is strongly recommended that for better comprehension of the discussed subject the readers refer to ISO/IEC 9126 – Part 1: Product Quality - Quality model [3] and ISO/IEC 14598 – Software Product Evaluation – Part 1: General overview [7]
- Both TL9000 and ISO/IEC standards offer the *process support* for identification, definition, measurement and evaluation of software product quality. In case of TL9000 Quality Management System Requirements Handbook [1] the support processes are located on the corporate level (refer to two modules, gray boxes in the complement model from Fig.3). In case of ISO/IEC standards the support is placed on measurement process management level and is being offered through ISO/IEC 14598 – Software Product Evaluation –Part 2: Planning and management [8]
- The process illustrated by complement model from Fig.6 does not need to be executed literally as presented, i.e. starting from Discovery phase and ending by Operation and Maintenance phase. It is the reader's decision in which point to enter and in which to exit the process, thus which actions to undertake and execute and which to neglect. However, such a decision must take into consideration the following issues:
    - When entering the process in point different than Discovery phase the user takes the risk of omitting (or neglecting) the operational quality requirements and Quality in Use requirements in software product quality definition. This may severely reduce the final quality of a software product
    - Entering the process in any point different than Discovery phase may reduce flexibility of iterations within the model

The quick reference summary of the process discussed in paragraph 3 is presented in Fig.7.

4. Bibliography

1. *TL9000 Quality Management System Requirements Handbook*, Release 3.0, QuEST Forum 2001
2. *TL9000 Quality Management System Measurements Handbook*, Release 3.0, QuEST Forum 2001
3. *ISO/IEC 9126 – Software and System Engineering – Product quality – Part 1: Quality model*. 1999-2002
4. *ISO/IEC 9126 – Software and System Engineering – Product quality – Part 2: External Quality Metrics*. 1999-2002
5. *ISO/IEC 9126 – Software and System Engineering – Product quality – Part 3: Internal Quality Metrics*. 1999-2002
6. *ISO/IEC 9126 – Software and System Engineering – Product quality – Part 4: Quality in Use Metrics*. To be published in 2004
7. *ISO/IEC 14598 – Software and System Engineering – Software Product Evaluation – Part 1: General overview*. 1999-2002
8. *ISO/IEC 14598 – Software and System Engineering – Software Product Evaluation – Part 2: Planning and management*. 1999-2002
9. *ISO/IEC 14598 – Software and System Engineering – Software Product Evaluation – Part 3: Process for developers*. 1999-2002
10. *ISO/IEC 14598 – Software and System Engineering – Software Product Evaluation – Part 4: Process for acquirers*. 1999-2002
11. *ISO/IEC 14598 – Software and System Engineering – Software Product Evaluation – Part 5: Process for evaluators*. 1999-2002
12. *ISO/IEC 14598 – Software and System Engineering – Software Product Evaluation – Part 6: Documentation of evaluation modules*. 1999-2002
13. *ISO/IEC 15288 - Software and System Engineering – Life Cycle Management – System Life Cycle Processes*. 2002
14. *ISO/IEC 15939 - Software and System Engineering – Software Measure*, 2002
15. Pfleeger S.L., *Software Engineering, Theory and Practice*, Second Edition. Prentice Hall 2001
16. Pressman R.S., *Software Engineering: A Practitioner's Approach*, Fifth Edition. McGraw-Hill 2001
17. Suryn W., Abran A., Bourque P., Laporte C., "Software Product Quality Practices. Quality Measurement and Evaluation using TL9000 and ISO/IEC 9126". Proceedings of STEP2002, Computer Society Press, 2003

ISBN 1-902505-56-5
British Computer Society (**BCS**) 2004

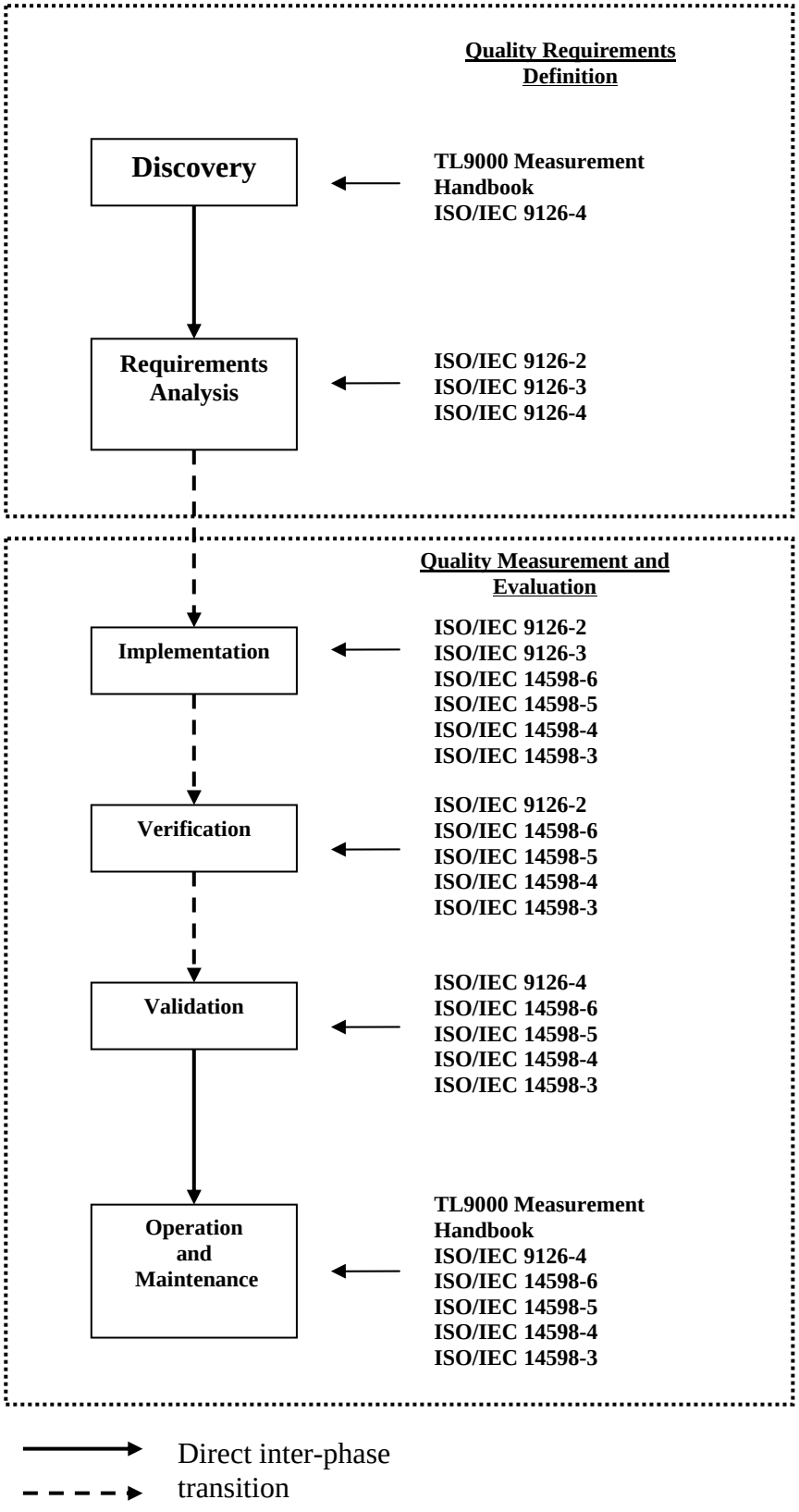

Fig.7 Quick reference summary for quality definition and evaluation process