Software Quality Psychology

Radosław Hofman[1]

Abstract*— *This article analysis non-technical aspects of software quality perception and proposes further research activities for this subject.

Cognitive science, psychology, micro economics and other human-oriented sciences do analyze human behavior, cognition and decision processes. On the other hand engineering disciplines, including software engineering, propose many formal and technical approaches for product quality description. Linkage between technical and subjective quality has been subject of research in areas related to food and agricultural applications and in this article we propose analysis of professional product perception which beyond doubt is a software product. This new research is called Software quality psychology.

Index terms—.Software, Quality perception, cognitive psychology.

I. Introduction and Motivation

Software Engineering emerged in the 1960's as an answer to software quality problems occurring at that time. Software products differed from other human industry products mainly because they were intangible and because their static attributes (attributes of a product that can be measured without using it) were irrelevant while dynamic attributes (attributes of a product that measure the behavior of a product when it is used under certain conditions) were of the highest importance. Software products usage is growing constantly and it is currently being used in almost every area of human activity. High quality software is then important issue not only for software developers but primarily for customers, users and people community as a whole (for example: software is in control of traffic lights, airplane steering systems, TV receivers etc.) From the 1970's until these days there were several attempts to understand and model software quality – the latest model, the ISO/IEC SQuaRE model, is still under development.

Do current approaches for defining the software quality [26] cover all issues related to its description and evaluation? When talking about the user perspective we have to take into account not only the technical software product quality, but also a set of communication occurrences, disinformation issues, expectations, beliefs or even mental states of the users. Let us assume that software production processes are stable and at their best, the same with project management processes. Are there still issues that could increase or decrease the perceived quality level? This question addresses activities which may occur or not between processes taking place in a real project (see fig. 1).

De gustibus et coloribus non est disputandum[2]. In philosophy of the mind there is an idea of qualia [13], defined as basic properties of sensory perception. Such perceptions cannot be fully explained by the person experiencing these sensations. If we assume that the quality is related to sensory perception then we could conclude that attempts to define the quality in terms of objective measures are unable to express personal views. Considering a customer or a user as an ultimate source of the software quality measure (compare CMM [37], ISO9001:2000, TQM, [44]) there is an important question: does the user quality perception follow objective measurements of quality (as for professional products), or users are following a set of observer biases because their subjective perception is unpredictable using a psychologically contextless model.

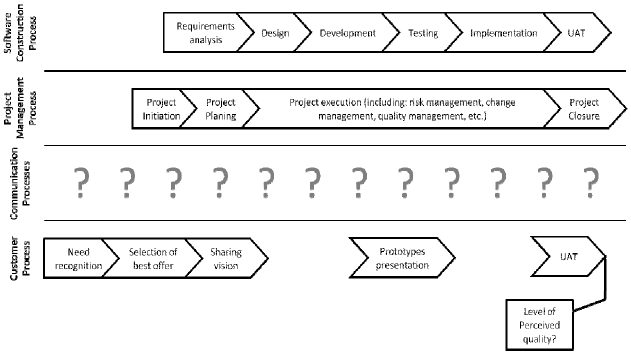

Fig. 1, Place of Software Quality Psychology

The same research question may be asked about the sources of customer or user satisfaction with the product they have. Certainly, there are attributes manifested by the product, but if satisfaction also depends on the anticipated values of the product's attributes or the way attributes are presented then it may be possible to guide the customer and the user perception process significantly changing the satisfaction level and the quality attributed to the software product.

The above questions, if the answers are yes, seem to be an important area for every software project, increasing probability that the user and the customer needs will be answered in the software attributes and the software product will be considered as having a satisfactionary level of quality.

II. Background

A. *Cognitive psychology*

Cognitive sciences, as an interdisciplinary study, concerning the human mind, intelligence, analyzing processes determining human behavior etc. Such studies were present in the philosophic works of Plato and Aristotle in ancient Greece, becoming an important stream of research in the 17th century inspired by Descartes. Despite of the rapid development of these sciences for the past 40 years, there are still more questions than answers about regarding understanding the processes of the human mind [42].

In this article we concentrate on the cognitive perception

[1] EUR ING, PhD Student Department of Information Systems at The Poznań University of Economics, Polish Standardization Committee Member, email: radekh@teycom.pl

[2] Latin: one should not discuss colors and the sens of taste

of software, adopting cognitive psychology, but also psychological concepts presented in the 18th, 19th and 20th century by Gossen, von Wieser, Kant, Hume, Freud, Maslow and other thinkers.

In our approach, it is important not only to understand the perception of software quality, but also we are discussing the possibility of influencing this perception. In other words, we propose a set of experiments, which could explicitly discover the relation between possible attributes of a product, which in consequence would allow us to affect the user's environment changing the user quality perception.

B. The history of valuation

The value of things is one of the basic ideas in human perception related processes. Sociological, psychological, economical, ethical value models have been developed from ancient times aiming to explain the reasons of subjective choices and personal preferences. Many thinkers had also tried to investigate how personal perception and behavior may be guided (or fail to be guided) in the process of valuation (compare Lawrence Kohlberg, Max Weber, von Weiser etc.). In this section we will present a short summary of ideas resulting from those works, seeming to be applicable in analysis of the evaluation processes of software quality perception and subjective distinction.

In neoclassical economics or in microeconomics the value of an object is often described as an equivalent to the object's price (dependant from supply and demand on a competitive or non-competitive market).

Putting aside such classical approaches, we concentrate on subjective understanding of the value. One of the first examples of the subjectivism of value is known as the diamonds and water paradox related to works of Adam Smith [43]. The question in this paradox uses the observation that water is crucial for human survival while diamonds are useless from the biological stand point, nevertheless diamonds are much more expensive than water is.

Attempts at finding an answer for this paradox were made by several 19th century thinkers. Herman Gossen had proposed the law of diminishing marginal utility [16], arguing that a unit of any good, which has already satisfied some need, is less desired than a unit of some other good, which has not satisfied an existing person's need. This thought was continued in the Austrian school of economics (named "philosophic") manifested by (among others) Friderch von Wieser [50] explicitly expressing observation, that satisfied needs are of less importance. In the 20th century Abraham Maslow had proposed the pyramid of needs [31]. The basic needs (D–needs), which are not recognized unless they are not satisfied, and the upper needs (B-needs) appearing only when the lower level needs are satisfied. In addition we may refer to Sigmund Freud thought that the world is perceived by humans mainly on the sub-conscious level [13].

The above ideas of the philosophical economy are important clues to understand the cognitive processes associated with the valuation of goods. Aristotle considered quality as a non quantitative factor allowing to distinguish a thing among others in the same category [25]. Thinking about valuation in terms of cognitive science, it is required to identify the mental state of the valuator, his/hers needs and the level of personal satisfaction, remembering that the satisfaction level is not linearly correlated with the saturation of needs.

Further on we will continue discussing "units" of goods but referring this idea to a software product they should not be considered as quantity of this software. We assume that for every quality characteristic we may increase or decrease its strength by some "unit", influencing the user's satisfaction. Under this assumption we follow economists defining utility as quantifiable in some units.

C. The Software Quality and Quality Models

From the 1960's the development of software products was perceived as an engineering discipline. One can also find first attempts to define goals and measures for software in that time. One of the most difficult measures to define was the software quality measure although it seemed to be a highly important attribute of the software product.

Software products brought about a new set of definition requirements in aspect of product measures and quality measures. Any measures, that had been known before (weight, size, durability, water resistance etc.) could not be applied to get significant information about a software product. The first attempts to define quality measures were made in the 1970's by McCall's [32] and Boehm's [8]. Successive attempts continue and the most current one is the SQuaRE (Software product QUality Requirements and Evaluation) model developed within the ISO/IEC25000 standards series. This new approach is perceived as the new generation of software quality models [48] and is being used for the decomposition of the end users perspective to software components requirements [1].

Quality models are the most important way to express the quality requirements in commonly understandable terms. In this article we use ISO/IEC SQuaRE vocabulary and consider the *software quality in use* as the representation of the user or customer perspectives of software quality.

D. The quality perception modeling

The need for measuring the quality of products is the natural consequence of an assumption that quality may be used to distinguish goods in the same category. The first software quality models and needs to measure the users opinion appeared in McCall [32] and Boehm [8] publications. At that time these were the only concepts.

In 1984 Grönroos described the quality as a function of expectations, dividing perception into three dimensions: functional, technical and image (perception of the vendor) [17] making the basis for the SERVQUAL model [36]. This model, and its successors are widely used quality perception models [24] not only for IT products but also for airline services, fast-food, telecommunications, banking, physiotherapy, web sites, healthcare and many others [6], [33].

Another approach to define the software quality perception is based on the belief revision theory [38]. This method adopts the AGM paradigm [5] or the Grove's system of spheres approach [18] and proposes an epistemological approach to define beliefs and their revision processes following the assumption that the observer is a rational, deductive agent using the principle of minimal change.

The above approach uses the assumption that users are rational agents using deductive reasoning and that beliefs may be represented in a formal system. The authors do not analyze the context of user (the context of purpose) nor the user's personal aspects (tiredness, attitude, treating evaluation seriously etc). It should be mentioned that the authors continue to measure technical quality factors, as defined in ISO9126, although usage of these is commonly regarded as too abstract to express the user's perspective [48]. The most important problem of the results is the problem of repetitive observations on the same group of users. In this case we may expect that the evaluation experiment was influencing users' opinions and their tendency for changing beliefs to a similar level could have been the effect of a group thinking phenomenon or could have been influenced by large amounts of external information not related to the software product being evaluated. In this article we propose a much broader view on the quality perception not limited to intrinsic software attributes.

It seems useful to analyze non software oriented attempts aiming to define the quality perception. One such attempt, in terms of cognitive processes, was made by Steenkamp in his dissertation in 1986 [46], [45], revised by Oprel in 1989 [35]. Since this work there have been several publications using the Steenkamp's model for analysis of the quality perception of food, plants etc., including research results in the area of social background influence on food quality perception [41].

Although some of the ideas used in Steenkamp's model are undoubtedly common for the perception of food products and software products, there are still some important differences. First of all, we have to distinguish the role of the person making valuation of the software quality [26], while Steenkamp considers all subjects as consumers. The second important difference is a change over time, which is not addressed in Steenkamp's model nor in successive models. Finally we may add that unlike food, software is used mainly by organizations (groups of individuals) and a group's behavior may influence the individual perception of quality.

Concluding this literature review we stress that there is no commonly accepted method for measuring the user perception of the software product's quality nor the common understanding of what "the user perception" actually is. The perception model presented in this article considers a wide view on user's cognitive processes, proposing research in areas not related to a software product directly, but probably influencing the perceived quality.

III. Software quality perception

A. Quality requirements and their limitations

The modern software engineering approaches explicitly state, that most of the software projects begin with the analysis phase which aims to establish the functional and the quality requirements (Software Engineering Body of Knowledge). Publications regarding the software quality lifecycle, accept this approach adding another type of requirements to be gathered: the evaluation process requirements [23], [21]. There are several guidelines how to identify requirements and build a comprehensive and correct set of stated requirements [12].

The most typical approaches adopt formal procedures for the acceptance of system requirements [23], [7], [28], [49] or an agile [3] or evolutionary [29] approach to the software scope definition. All of these approaches seem to be helples in the areas of discoverying implied user needs[3] or needs that are being satisfied during the analtzis phase. Elusion approach does not prevent users from forgetting obvious issues (from their prespective) while the evaluation of the prototype does not need to employ the users' complete set of needs having only an abstract reason to work with the evaluated prototype (this bias is partially evened out in the evolutionary approach [15]). If the software is to support the new internal organizational scheme (a software implementation project is often associated with some general project of a organizational change) then users have a limited ability to evaluate it before the new structure is implemented.

Another problem with the agile methodologies is related with the exertion of the user what leads to a discouragement. One has to remember that during the early phases of the project the user receives a non-working or having poor quality prototype for evaluation. It may be observed that the user prepossess his mind and finds the software non reliable on the final acceptance task.

Concluding this section we shall underline the gap between the stated requirements and the user needs, especially the implied needs. We also have to remember that the software engineering defines mainly the technical (intrinsic) quality attributes, while the user valuates only these attributes which are associated with his own observation of the software from the perspective of his everyday tasks regarding both: the software and his temporal state of mind.

B. Expectations

In this section we shall recall important works of two thinkers. Immanuel Kant in his theory of perception argued that the human beings adopt their a'priori concepts and knowledge, perceiving their own thoughts (not the real attributes of the observed reality) [19]. David Hume analyzing the sources of human concepts observed that the people tend to reject observations outstanding from their beliefs [47]. Both of the above works are widely accepted in modern psychology what may be put in the statement that the human perception is only an interpretation of the human mind [30].

Referring to the buyer decision process, defined as the psychological and cognitive model (e.g. the motivation and the need reduction [11]) one may see that before the decision of buying a new software was made, the customer had recognized some need, had analyzed alternatives and had made the decision. This is the first time when some expectations have appeared. The decision was made in order to satisfy the customer's recognized need (with implied needs). What may not be obvious is the part of the decision regarding the current customer solution (if the new one is to satisfy needs satisfied which are currently satisfied). This attitudes is a priming process [27] preparing the a'priori perspective as in the Kant's theory.

Before the software product is presented for the first time to the customer and users it had already been evaluated by

[3] The most of the quality definitions state that the quality is an ability to satisfy stated and implied user needs [22]

this group and its quality had been assessed. This observation is a basic knowledge for the brand managers promoting physical goods but seem to be not applied in the software industry.

C. The software quality perception model

Models are intended to be a reliable prediction of future observations, repositories of rules collated and integrated from past research [39]. The software quality perception model proposed in [20] is designed according to the above idea taking into account cognitive psychology, the quality perception models for the other types of products and the commonly accepted software quality model SQuaRE.

The proposed model is omitting elements associated with the preference and choice making what may be seen in other quality perception models, as the result of focusing on the software produce evaluation purposes. Software, in most of the cases (especially the tailored software) is fully evaluated after the purchase and implementation, so there are no buyer decisions to be made.

The software quality perception model is presented on fig. 2.

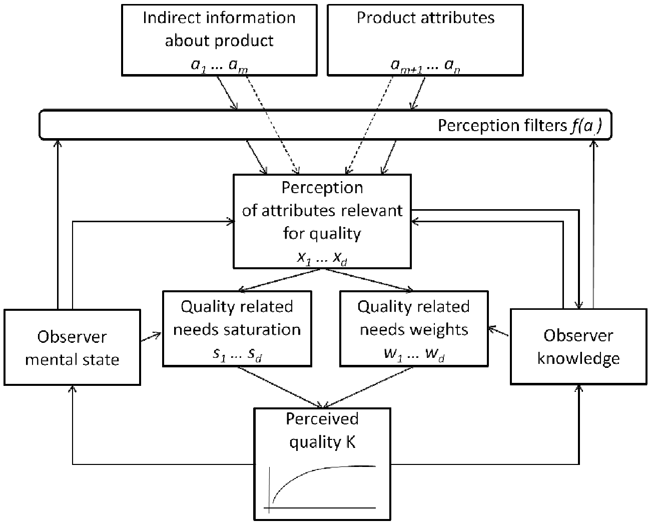
Indirect information about product
$a_1 \dots a_m$
Product attributes
$a_{m+1} \dots a_n$
Perception filters $f(a)$
Perception of attributes relevant for quality
$x_1 \dots x_d$
Observer mental state
Quality related needs saturation
$s_1 \dots s_d$
Quality related needs weights
$w_1 \dots w_d$
Observer knowledge
Perceived quality K

Fig. 2, The software quality perception model

The product attributes (intrinsic attributes) and information associated with the product (extrinsic attributes) are in the first stage filtered by the observer's attention filter. From the mathematical point of view, filtering is an application of a certain function ($f_x(a_x)$) for the original attribute. This function is a consequence of the observer's temporal mental state and knowledge (including the past experience with the product or a similar kind of products) etc. The evaluation product quality employs the establishment of perceived attributes relevant for the quality – the observer chooses a set of attributes expressing the quality seen through perception filters. This process is also based on the observer's mental state and his knowledge. We assume that the observation (information about perceived attribute value) also augments the user's knowledge for the future.

In the next stage the observer combines perceived quality related attributes values through the perspective of needs, taking into account the general importance of a need and the subjective saturation of this need as the base. The overall quality value returns feedback to the observer's knowledge and his mental state.

x_i variables (the perceived value of attributes relevant for the quality) may be interpreted as the direct measures assigned in the observation process (where the observation is a evaluation of the product or a processing information associated with product) or the derived measures based on the direct measures. This concept is similar to the Quality Measure Elements idea in ISO/IEC 25021.

Another important issue associated often with the evaluation is a usage of somebody's else perspective. In this case the observer relates needs saturation to the imaginary mental state, attributed to future users of the product. This attribution is prone to be mistaken, because the observer is using his knowledge base while the users will use a different knowledge base (compare the theory of the mind [42]).

The calculation of the subjective quality value, or in the other words conversion to a uni-dimensional quality measure, in the most of the quality models is a linear combination of simple measures and weights:
(compare [46]). This approach adopts the assumption that each "unit" of the attribute value influences the overall quality index with the same strength.

The quality perceived by human should be considered as a latent variable and in consequence one could employ the Rash polytomus model [4] designed to measure personal attitudes. Irrespective to the mathematical model involved, we assume that, the quality perception of a single attribute value follows the diminishing marginal value concept. We also assume that attribute values are additive so the overall quality value may be calculated as:

Where $F_i(s_i,w_i,x_i)$ is a monotonic function.

Similar to [46], the perceived quality in the above model differs from the other approaches in the regard of the quality as neither absolute nor as objective.

This model may be further extended with the quality perception change patterns (the quality lifecycle patterns) – at this moment, we only assume, that due to the observer mental state and knowledge change, the quality attributes perception changes too over time.

D. Affecting the quality perception

If the user perception of quality is dependant not only from the intrinsic product attributes, then there are several methods to influence the overall quality perception. Such methods are currently being used for commercial of the shelf products (COTS), although there should be a possibility to use them in the most of software projects.

We could affect the quality value during the evaluation and the long term operation but it is likely that various methods will have a differing impact on quality values during those two stages.

An example of a positive impact on the quality value during the evaluation is a use of the primacy effect (customer is being presented the high quality product at first).

Perception affecting methods are known mainly in marketing and psychology research. In this article we substantiate the need for further research in the area of affecting software quality perception methods. According to the authors best knowledge, there are no publicly available publications in this subject.

IV. Evaluation of the perception model

A. The Software Quality Psychology

The modern experimental psychology, understood as discipline of psychological science has its origins in the late XIXth century when Wilhelm Wundt founded the first psychological laboratory near Leipzig [9].

The goal of experiments in Software quality psychology is to trace the effect on the quality perception from user's and customer's perspective caused by events occurring in the project what follows the general goal of experiments (tracing the cause-effect relations).

The evaluation of human oriented judgments requires a set of operational definitions, aiming to be valid and reliable, for ensuring that a term is interpreted equally by the researcher and persons being examined. The most important definition in this area is the definition of software quality: Software product quality is something which distinguishes software products in the same category (based on the Aristotle definition of quality). The software quality splits into characteristics according to the SQuaRE model [23].

This definition cannot be used strictly for engineering purposes, but since the Software quality psychology is aiming to measure the quality perceived by a user, we assume that the user is the ultimate source of the software product quality valuation.

One has to remember that perception issues, that the Software quality psychology is about to trace, may be prone from the social context of users. For example, we may expect the group thinking effect or the regression to common valuation effect in structured organizations [20]. Tracing the perception of software quality among such structures requires the modeling of real world's relations between participants, perceived objects and informative relations between them.

B. The experiment methodology

Designing an experiment, we want its results to be valid and sound. Although it is difficult to a'priori estimate the soundness or the application possibility of the experiment results, it is possible to estimate the validity of an experiment. Valid experiment results may then be applied in practice, what will result in the soundness estimation.

The typical formal procedure for psychological experiments is an independent groups plan for testing the one or more independent variables impact.

The design of experiment has to address issues related with the need of longitudinal research (research during the observed project), the appropriate sampling method and also methods to exclude the confounding (alternative explanations of the observed effect [10]) including the accidental share of information between independent groups or the possibility of Hawthorne effect occurrence [2] etc.

From our perspective, the one of the effective experiment methods, is to design a special instance of the environment where the experimenter could control the occurrence of certain events. For example: in [20] there is a description of an application in which the experimenter is able to apply the expected quality level (understood as the probability of a fault).

According to Mook [34], if one is testing a theory based on the psychological studies then the external validity is not of the key importance [40]. If the experimenter is aiming to analyze psychological achievements and phenomenon in the aspect of the software quality perception, then Mook's remarks about the external quality should draw the attention to the internal validity and the effect size as the most important part of the experiment characteristics.

In summery we conclude that the software delivery environment is a complex and difficult to model for the purpose of a psychological experiment. It is of the key importance to design a experiment, including subjects, their relations, the application, project delivery issues, evaluation tasks, the quality expression method etc. as in the real world in order to obtain results applicable to certain project reality.

V. Summary and further research

A. The conclusion

In this article we have presented our research in the area of the psychological perception of the software quality assuming that the quality perception is a highly subjective manner. The presented software quality perception model requires more psychological research to confirm its validity, strength and its usefulness in practical applications.

The software engineering defines several software development lifecycles, methodologies, approaches etc. Although the most of them concentrates on the software production processes resulting in the delivery of a deliberate set of intrinsic product attributes we may observe that the same products delivered with a various history track record may be perceived differently by users. This is the area of the interest of Software quality psychology research.

The Software quality psychology should not be interpreted as the replacement of good practices and the engineering craft of the software development and delivery. It should be understood as an additional set of knowledge, helpful in projects lifecycles for the purpose of the achievement of a higher level of the user satisfaction from the product.

The consequence of a higher quality overall mark evaluated by the user is significant – the probability of the acceptance of the software increases, the customer satisfaction increases and, in consequence, the business cooperation between the vendor and the customer grows.

B. Current research plan

Current research plans concentrate on the evaluation of the effect size for particular independent variables. According to the model itself, the perception is affected by a temporal mental state. At the lower level the temporal mental state drags the observer's attention to certain characteristics, on the middle level it impacts the observer's motivation with the vision of the product usage and on the upper level this state affects the subjective feeling about the importance of observed characteristic. The second affection source is the observer knowledge drawing the attention to the characteristics set which are often important, giving an input for associations between the observations and the typical consequences of a certain product behavior. The knowledge also influences the aware assignment of the weights to certain characteristics. Our current research aims to measure the effect for all of the above interrelations.

These interrelations are resulting from the theoretical basis of psychology and cognitive sciences investigated in non IT areas. In this situation, we do expect the existence of such relations also for the IT industry. But we still require more research to estimate the importance of the impact on the real situation from the psychological interactions during the project. After the first measures of the effect size were done, we are planning to design and measure the reaction strength for a specific strategy chosen in the IT project. These research activities should result in setting up a new knowledge base of the best practice in the managing of the customer perception of the software product.

VI. References

[1] Abramowicz, W., Hofman, R., Suryn, W., Zyskowski, D. *SQuaRE based quality model*. International Conference on Internet Computing and Web Services, Hong Kong: International Association of Engineers, 2008

[2] Adair, G. The Hawthorne effect: A reconsideration of the methodological artifact. Journal of Applied Psychology vol 69, 1984

[3] *Agile Manifesto*. Retrieved 2008, from http://www.agilemanifesto.org

[4] Alagumalai, S., Curtis, D.,Hungi, N. *Applied Rash Measurement: A book of exemplars*. Springer-Kluwer, 2005

[5] Alchourron, C., Gardenfors, P., Makinson, D. *On the Logic of Theory Change: Partial Meet Functions for Contraction and Revision*. Journal of Symbolic Logic , vol 50, 1985.

[6] Babulak, E., Carrasco, R. *The IT Quality of Service Provision Analysis in Light of User's Perception and Expectations*. Int. Symposium on CSNDSP. Staffordshire University, 2002.

[7] Beck, K. *Extreme programming eXplained: embrace change*. Addison-Wesley, 2000.

[8] Boehm, B., Brown, J., Lipow, M., & MacCleod, G. *Characteristics of software quality*. New York: American Elsevier, 1978.

[9] Boring, E. *History of Experimental Psychology*. Second Edition. Prentice-Hall, 1950.

[10] Campbell, D., Stanley, J. *Experimental and quasi-experimental designs for research*. Chicago, Rand McNally, 1966.

[11] Cheng, M., Luckett, P., Schulz, A. *The Effects of Cognitive Style Diversity on Decision-Making Dyads: An Empirical Analysis in the Context of a Complex Task*. Behavioral Research in Accounting, vol 15, 2003.

[12] Davis, A. *Software Requirements: Analysis and Specification*. Prentice Hall, 1993.

[13] Dennet, D. *Quining Qualia*. In Marcel, A., Bisiach, E., *Consciousness in Modern Science*. Oxford University Press, 1988.

[14] Freud, S. *Das Ich und das Es (The Ego and the Id)*. 1923

[15] Glib, T., *Principles of Software Engineering*. Wokingham, England: Addison-Wesley, 1988.

[16] Gossen, H. *Die Entwicklung der Gesetze des menschlichen Verkehrs und der daraus fließenden Regeln für menschliches Handel (The Development of the Laws of Human Intercourse and the Consequent Rules of Human Action)*, 1854.

[17] Grönroos, C. *A service quality model and its marketing implications*. European Journal of Marketing, vol 18 (4), 1984.

[18] Grove, A. *Two modellings for theory change*. Journal of Philosophical Logic, vol 17, 1988.

[19] Haden, J., Körner, S. *Kants Leben und Lehre (Kant's Life and Thought)*. (E. Cassirer, Trans.) Yale University Press, 1981.

[20] Hofman, R. *Software Quality Perception*, in Elleithy, K. (ed). *Innovations and Advanced Techniques in Systems, Computing Sciences and Software Engineering*, University of Bridgeport, Springer, in preparation – publication in 2009

[21] ISO/IEC14598. *Information technology - Software product evaluation*. Geneve: International Standardization Organization, 1999.

[22] ISO/IEC25000. *Software Engineering - Software product Quality Requirements and Evaluation (SQuaRE)*. Geneve: International Standardization Organization, 2005.

[23] ISO/IEC25010. *Software Engineering - Software product Quality Requirements and Evaluation (SQuaRE)*. Internal ISO/IEC JTC1/SC7 document – Commision Draft, 2007.

[24] Kang, G., James, J., Alexandris, K. *Measurement of internal service quality: application of the Servqual battery to internal service quality*. Managing Service Quality, vol 12 (5), 2002.

[25] Kiliński, A. *Jakość*. Warszawa: Wydawnictwo Naukowo Techniczne, 1979.

[26] Kitchenham, B., Pfleeger, S. *Software Quality: The Elisive Target*. IEEE Software, vol 13 (1), 1996.

[27] Kolb, B., Whishaw, I. *Fundamentals of Human Neuropsychology*. Freeman, Worth Publishers, 2003.

[28] Kruchten, P. *The Rational Unified Process: An Introduction* (3rd edition). Addison-Wesley Professional, 2003.

[29] Krzanik, L. *Enactable models for quantitative evolutionary software processes*. In Tully, C. *Proceedings of the Forth International Software Process Workshop (ISPW '88)*. Moretonhampstead, Devon, UK: IEEE Computer Society, 1988.

[30] Libet, B., Sutherland, K., Freeman, A. *The Volitional Brain: Towards a Neuroscience of Free Will*. Thorverton, UK: Imprint Academic, 2000.

[31] Maslow, A. *A Theory of Human Motivation*. Psychological Review vol 50, 1943.

[32] McCall, J., Richards, P., Walters, G. *Factors In software quality*. Griffiths Air Force Base, NY, Rome Air Development Center Air Force Systems Command, 1977.

[33] Miguel, P., daSilva, M., Chiosini, E., Schützer, K. *Assessment of service quality dimensions: a study in a vehicle repair service chain*. POMS College of Service Operations and EurOMA Conference New Challenges in Service Operations. London, 2007.

[34] Mook, D. *In defense of external invalidity*. American Psychologist vol. 38, 1983

[35] Oprel, L. *Kwaliteit in breder perspectief, Proefstation voor de Bloemisterij Aalsmeer,* 1989

[36] Parasuraman, A., Zeithaml, V., Berry, L. *A conceptual model of services quality and its implication for future research*. Journal of Marketing, vol 49 (4), 1985.

[37] Paulk, M., Weber, C., Curtis, B., Chrissis, M. *The Capability Maturity Model: Guidelines for Improvement of the Software Process*. Addison-Wesley, 1995.

[38] Peppas, P., Williams, M. *Constructive modellings for theory change*. Notre Dame Journal of Formal Logic, vol 36 (1), 1995.

[39] Rickert, K. *Beef cattle production*. In Tijskens, L., Hertog, M., Nicolai, B., Food process modelling. Cambridge: Woodhead Publishing, 2001.

[40] Shaughnessy, J., Zechmeister, E., Zechmeister, J. *Research Methods in Psychology*. Seventh edition. McGraw-Hill, 2005.

[41] Sijtsema, S. *Your health!? Transforming health perception into food product characteristics in consumer-oriented product design* (WAU no. 3359). Wageningen University Dissertation, 2003.

[42] Smart, J. *The Identity Theory of Mind*. In Stanford Encyclopedia of Philosophy, 2007.

[43] Smith, A. *An Inquiry into the Nature and Causes of the Wealth of Nations*, 1776.

[44] Stavrinoudis, D., Xenos, M., Peppas, P., Christodoulakis, D. *Early Estimation of Users' Perception of Software Quality*. Software Quality Journal, vol 13, 2005.

[45] Steenkamp, J. *Product quality: an investigation into the concept and how it is perceived by consumers* (WAU no. 1253). Wageningen University Dissertation, 1989.

[46] Steenkamp, J., Wierenga, B., Meulenberg, M. *Kwali-teits-perceptie van voedingsmiddelen deel 1. Swoka*. Den Haag, 1986.

[47] Stroud, B. *Hume*. London & New York: Routledge, 1977.

[48] Suryn, W., Abran, A. *ISO/IEC SQuaRE. The seconod generation of standards for software product quality*. IASTED2003, 2003.

[49] Wiegers, K. *Software Requirements 2: Practical techniques for gathering and managing requirements throughout the product development cycle* (second edition). Redmont: Microsoft Press, 2003.

[50] Wieser, F. *Der natürliche Werth (Natural Value)*. 1889.