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.

Quality models are a way of expressing quality, but despite of these definitions one has to know what does the quality of a software product mean. We do know [26] that quality means different set of characteristics for different perspectives, and for different contexts of use (the same product may be perfect in one and useless in another context of use).

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

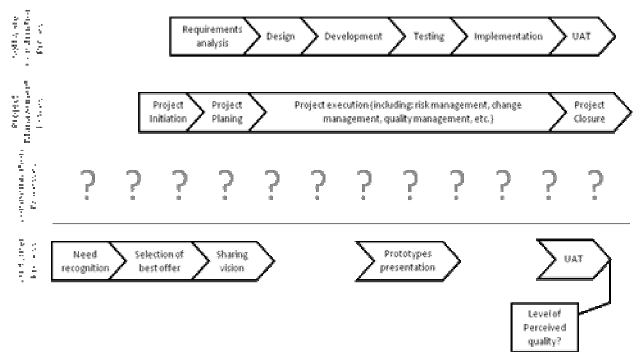

Fig. 1, Place of Software Quality Psychology

De gustibus et coloribus non est disputandum. In philosophy of the mind there is a 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 set of observer biases having their subjective perception unpredictable using a psychologically contextless model.

The same research question may be asked about sources of the customer and the user satisfaction with the product they have. Certainly there are attributes manifested by the product, but if the satisfaction depends also from anticipated attributes values 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 a software product.

Above questions, if answer is yes, seem to be important area for every software project increasing probability that user and customer needs will be answered in software attributes and the software product will be considered as having satisfactionary level of quality.

II. Background

A. Cognitive psychology

Cognitive sciences, as interdisciplinary study, concern on human mind, intelligence, analyzing processes determining human behavior etc. Such studies were present in philosophic works of Plato and Aristotle in ancient Greece, be-

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

coming important stream of research in 17th century owing to Descartes. Despite of rapid development of these sciences for past 40 years there are still more questions than answers about explanations of human mind processes [42].

In this article we concentrate on cognitive perception of software, adopting cognitive psychology, but also psychological concepts presented in 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 perception of software quality, but also we are discussing possibility of influencing this perception. In other words we propose set of experiments which could explicitly discover relation between possible attributes of product and influence on environment what in consequence would change quality perception.

B. History of valuation

Value of things is one of basic ideas in human perception related processes. Sociological, psychological, economical, ethical value models has been developed from ancient times with attempts to explain why human beings prefer or choose some things over others and how personal behavior may be guided (or fail to be guided) in valuation (compare Lawrence Kohlberg, Max Weber, von Weiser etc.). In this section we will present short summary of ideas resulting from these works which seem to be applicable in analysis of software quality perception and subjective distinction to software products made by evaluators.

In neoclassical economics or in microeconomics value of object is often described as nothing else then the price dependant from supply and demand on competitive or non-competitive market.

Putting aside classical approaches we concentrate on subjective understanding of value. One of classic examples of subjective value issue is known as diamonds and water paradox related to works of Adam Smith [43]. The question in this paradox uses observation that water is crucial for human surviving while diamonds are useless from biological point of view, but diamonds are much more expensive then water is.

Answer attempts for this paradox were given by several 19th century thinkers. Herman Gossen proposed law of diminishing marginal utility [16] arguing that if a good satisfies some need, then value of next unit of this good is diminishing. This thought was continued in Austrian school (named "philosophic") and Friderch von Wieser [50] explicitly expressed observation that satisfied needs are of less importance. In 20th century Abraham Maslow proposed pyramid of needs [31]. Basic needs (D–needs), which are not recognized unless not satisfied, and upper needs (B-needs) appear only when lower level needs are satisfied. In addition we may refer to Sigmund Freud thought that world is perceived by humans mainly on sub-conscious level [13].

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

Further on we will discuss "units" but referring to software they should not be considered as quantity of this software. We assume that for every characteristic we may increase or decrease its strength by some "unit" influencing user satisfaction. In this assumption we follow economists defining utility as quantifiable in some units.

C. Software Quality and Quality Models

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

Software products stated new set of definition requirements in aspect of product measures and quality measures. Any measures known for ages (weight, size, durability, water resistance etc.) could not be applied to get significant information about software. First attempts to state quality measures were done in 1970s by McCall's [32] and Boehm's [8] quality models. Such attempts are still taking place and currently the most modern is the SQuaRE (Software product QUality Requirements and Evaluation) model developed within ISO/IEC25000 standards series. This new approach is perceived as new generation of software quality models[48][48] and is being used for decomposition of end users perspective to software components [1].

Quality models are important way to express quality in commonly understandable terms. In this article we use ISO/IEC SQuaRE vocabulary and consider software quality in use as representation of user and customer view on software quality.

D. Quality perception modeling

The need of measuring quality of the products is a natural consequence of stating quality as thing that matters. First software quality models and need to measure users opinion appeared in McCall [32] and Boehm [8] publications. At that time it was only concepts.

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

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

Above approach uses assumption that users are rational agents following deductive reasoning and that beliefs may be represented in formal system. Authors does not analyze context of user (context of purpose) nor users personal aspects (tiredness, attitude treat evaluation seriously etc). It should be mentioned that authors continue to measure technical quality factors, as defined in ISO9126 although usage of them is commonly regarded as to abstract for users to express user point of view [48]. The most important problem of those results is problem of repetitive observations on the same group of users. In this case we may expect that evaluation experiment was also influencing users opinion and tendency of changing beliefs to similar level could be group thinking effect or could be influenced by dozens of external information having nothing to do with the software. In this article we propose much broader view on quality perception not narrowed to intrinsic software attributes.

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

Although some of ideas used in this model are undoubtedly common for perception of food products and software products, there are some important differences. First of all we have to distinguish role of person making valuation of software quality [26], while Steenkamp considers all people as consumers (we can distinguish between consumers and people responsible for food purchasing for some organizations). Second important difference is change over time, what is not addressed in Steenkamp's neither in successor models. In last place we may add that unlike food, software is used mainly by organizations (groups of individuals) and group behavior may influence individual perception of quality.

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

III. Software quality perception

A. *Quality requirements and limitations*

Modern software engineering approaches explicitly state, that software project begins with analysis resulting in establishment of functional and quality requirements (Software Engineering Body of Knowledge). Publications concerning software quality lifecycle accept this approach adding another type of requirements to gather – evaluation process requirements [23], [21]. There are several guidelines how to identify requirements and build comprehensive and correct set of stated requirements [12].

On the other hand it is commonly accepted to define quality as ability to satisfy stated and implied needs [22]. Implied needs remain however unaddressed in software engineering state of art. Main stream approach assumes that users explicitly express they requirements, give them priorities, analysts identify and analyze user requirements and at the end of this phase users accept complete set of requirements [23], [7], [28], [49].

Looking back on great thoughts associated with cognitive psychology we may identify significant limitation associated with such approach. Although users are certainly best source for defining goal and purpose of software usage, they are unable to identify their own basic needs or are unable to assign priority expressing real importance of the need which are being satisfied.

An alternative for formal specification agreeing can be found in agile development methodologies. Most of them stress need for being close to user and allow user to evaluate software product as often as possible [3]. Another example is evolutionary approach assuming deliveries to live environment several times in the project lifecycle [29]. These methods should be much more effective and allow higher success ratio for IT projects.

Although agile methodologies are intended to answer problems with incompleteness of user perspective they seem to be vulnerable from the same cognitive issue. Users evaluating software prototype do not need to employ they complete set of needs having only abstract reason to work with evaluated prototype (this bias is partially evened out in evolutionary approach [15]). We may think about it as having brain storm meeting in subject "are these tools enough for us to live on desolated island?". Unless people were really put in real situation they do evaluate appropriateness of product against their assumptions about imaginary reality. If software is to support new internal organization (planned organizational change is often associated with new software to be implemented) then users have limited ability to evaluate it.

Another problem associated with agile methodologies is that this approaches forces user to exert what leads to discouragement. One has to remember that on early phases user receives non-working or having poor quality software prototypes for evaluation but it may be observed that user prepossess his mind rejecting software on final acceptance.

Concluding this section we shall underline gap between stated requirements and user needs, especially implied needs. We also have to remember that software engineering defines mainly technical (intrinsic) quality attributes, while user valuates only these attributes which are associated with his observations of software and its surrounding from perspective of his everyday use of software and his 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 human beings adopt their a'priori concepts and knowledge perceiving their thoughts not real attributes of observed reality [19]. David Hume analyzing where human concepts come from observed that people tend to reject observations outstanding from other observations or their beliefs [47]. Both of above works are widely accepted in modern psychology where thesis that human perception is only interpretation of human mind is rather common [30].

Referring to buyer decision process, defined as psychological and cognition model such as motivation and need reduction [11] one may see that before decision of buying new software was made, customer recognized his need, analyzed alternatives and made decision. This is first time when some expectations have appeared – that new product will satisfy customer recognized need (with implied needs), and will still satisfy needs satisfied with old system (this requirement is not often explicitly formulated). This attitudes play role of priming processes [27] preparing a'priori concepts as in Kant's theory.

Before software product is seen by customer and users for the first time it has already been evaluated and its quality was assessed. This observation is obvious for brand managers promoting physical goods but seems to be not applied in software industry.

C. *Software quality perception model*

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

Proposed model is omitting elements associated with preference and choice making which may be seen in other quality perception models, as an result on focusing on software produce evaluation purposes. Software in most cases, especially tailored software is fully evaluated after purchase and implementation, so there are no decisions about purchase to be made.

Software quality perception model is presented on fig. 2.

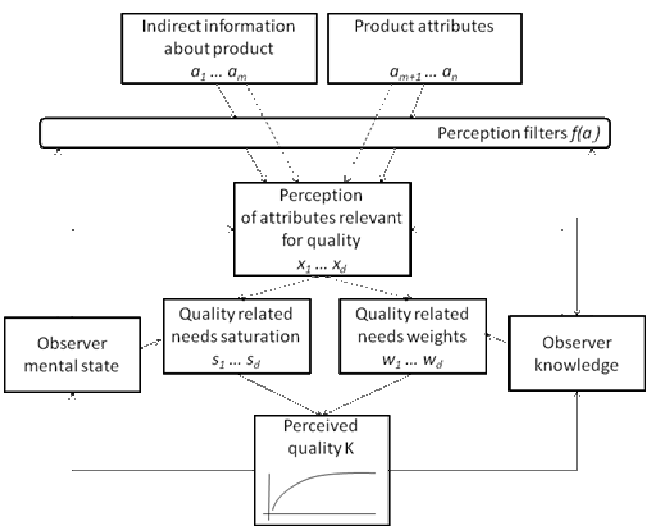

Fig. 2, Software quality perception model

Product attributes (intrinsic attributes) and information associated with product (extrinsic attributes) are at first filtered by observer attention filter. From the mathematical point of view filtering is application of certain function ($f_x(a_x)$) for original attribute. This function is a consequence of observer mental state and knowledge (including past experience with product or similar kinds of products) etc.

Evaluation product quality employs establishment of perceived attributes relevant for quality – observer chooses set of attributes expressing quality seen through perception filters. This process is also based on observer mental state and his knowledge. We assume that observation (information about perceived attribute value) augments user knowledge.

In next stage observer relates perceived quality related attributes values through perspective of needs taking general importance of need and subjective saturation of need as the base. Overall quality gives feedback to observer knowledge and his mental state.

x_i variables (perceived value of attributes relevant for quality) may be interpreted as direct measures assigned in observation process (where observation is evaluation of product or processing information associated with product) or derived measures based on simple measures (for example number of software errors and number of lines of code are simple measures; number of software errors per number of lines of codes is derived measure relevant for quality). This concept is similar to Quality Measure Elements idea in ISO/IEC 25021.

Another important issue is evaluation for somebody's else perspective. In this case observer relates needs saturation to imaginary mental stated attributed to future users of the product, but probably using only observer state of knowledge (compare theory of the mind [42]). Such observation method tends to produce quality measures differing from real users evaluation, because evaluators are prone to mistakes in setting up imaginary state of mind of future users (e.g. cognitive bias called conjunction fallacy – tendency to assume that specific conditions are more probable then general ones), and using different knowledge base (user having wider/narrower knowledge about product usage will assign different set of weights).

Calculation of subjective quality value, or in other words conversion to uni-dimensional quality measure, in most of quality models is an addition operation of single measures multiplied by attribute weight: $K(x,w) = \sum_i w_i *$ (compare [46]). This approach adopts assumption that each "unit" of attribute value influences overall quality index with same strength.

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

$$K(x,w,s) = \sum_i F_i(s_i, w_i, x_i)$$

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

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

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

D. *Affecting quality perception*

If user quality is dependant not only from intrinsic product attributes, then there are several methods to influence overall quality perception. Such method are being used mainly for repeatable products (COTS), although there is theoretical possibility to use them in most of software projects.

We may affect quality value during evaluation and long term operation but it is likely that different methods will have different impact on those two values.

Example of positive impact on quality value during evaluation is use of primacy effect presenting high quality product first.

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

IV. Evaluation of Perception model

A. *Software Quality Psychology*

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

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

Evaluation of human oriented judgments require operational definitions, aiming to be valid and reliable, for making sure that a term is interpreted equally by the researcher and examined subjects. 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 Aristotle definition of quality). Software quality splits to characteristics, sub-characteristics according to SQuaRE model [23].

This definition cannot be used strictly for engineering purposes, but since Software quality psychology is aiming to measure user perceived quality, we assume that people have ability to order object according to quality measure they assign to objects.

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

B. *Experiment methodology*

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

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

Examining events, information packages etc. during delivery project impacting software quality perception creates natural need for longitudinal research what makes random sampling very difficult. Also tested situations require much attention to exclude confounding - alternative explanations of observed effect [10] (for example if the project manager of first project is a nice woman and in the second a rude man then their attitude may impact perception of product quality regardless of other differences). Design should also address accidental exchange of information between independent groups and possibility of Hawthorne effect [2].

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

According to Mook [34] if one is testing theory based on psychological studies then external validity is not of key importance [40]. If experimenter is then applying psychological discoveries and phenomenon to software quality perception investigation, assuming that perception of software quality relies on cognitive processes and follows cognitive biases, then Mooks remarks about external quality should draw attention to internal validity and effect size as the most important part of experiment characteristics.

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

V. Summary and further research

A. *Conclusion*

In this article we have presented our research in area of psychological perception of software quality assuming perception as highly subjective manner. Presented software quality perception model requires more psychological research to confirm its validity and strength and also usefulness in practical applications.

There exists several software development lifecycles, methodologies, approaches and so on. Although most of them concentrates on software production processes resulting in delivery of deliberate set of intrinsic product attributes we may observe that technically exactly the same products delivered with application of the same methodology may be perceived differently by users. This is the area of interest of Software quality psychology research.

Software quality psychology should not be interpreted as replacement of good practices and engineering craft on software development and delivery. It should be understood as additional set of knowledge helping in projects to achieve higher level of user satisfaction for the product.

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

B. Current research plan

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

These interrelations are resulting from theoretical basics of psychology and cognitive sciences, so it is almost obvious that they exist. But what is important is measurement of their potential impact on real situation. After first measures of effect size were done we are planning to design and measure reaction strength for specific strategy in IT project. These research activities should result in setting up of a knowledge base for best practice in managing customer perception of software.

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.