Janusz GÓRSKI, Krzysztof WYRZYKOWSKI

Gdansk University of Technology

A method of supporting client-provider cooperation in software acquisition processes and its evaluation criteria

Abstract. *This article presents a method of supporting client–provider cooperation in software acquisition projects. The method, called WIKLIDO, focuses on modelling of client-provider interactions and provides its users with the means of structuring their cooperation. WIKLIDO has been already successfully applied in a number of real life case studies. The article introduces the method and explains how it is to be evaluated in a systematic way with the help of metrics designed using the common GQM (Goal Question Metrics) approach.*

Streszczenie. *Artykuł przedstawia metodę Wspomagania Interakcji KLienta i Dostawcy Oprogramowania (WIKLIDO). Metoda ta została z powodzeniem zastosowana w dwóch studiach przypadków w komercyjnych projektach pozyskiwania oprogramowania. Artykuł wyjaśnia w jaki sposób metoda wykorzystuje środki modelowania do usystematyzowania interakcji pomiędzy klientem i dostawcą oprogramowania. Przedstawia również w jaki sposób wykorzystano podejście GQM (ang. Goal Question Metrics) do systematycznego wywiedzenia metryk służących do eksperymentalnej oceny metody WIKLIDO. (**Metoda wspomagania współpracy klienta i dostawcy w procesach pozyskiwania oprogramowania i kryteria jej eksperymentalnej oceny**)*

Keywords: software acquisition risks, software process improvement, client-provider cooperation, modelling, best practices.
Słowa kluczowe: pozyskiwanie oprogramowania, modelowanie.

Introduction

Insufficient cooperation between the client and the provider of software poses major risks for software acquisition processes [12]. Misunderstood objectives, inadequate scope, erroneous or missing requirements, lack of visibility of, and control over the development and deployment processes are among these risks which have their roots in insufficient cooperation and communication between clients and providers. The main stakeholders of software acquisition represent different perspectives and often different organization cultures. The client and provider domains are typically different (for instance IT on the provider's side and medicine, banking, transportation, administration etc., on the client's side). Client and provider representatives are usually of different professions, different levels of experience, knowledge and education. It makes the client–provider cooperation difficult and requires special attention during software acquisition. The shortcomings and inadequacies of this cooperation are important risk factors which can severely threaten the success of software acquisition projects.

Experience collected in the domain of software engineering led to identification of multiple best practices which are now reflected in standards and guidelines. Some of these recommendations address the client-supplier cooperation problem and provide guidelines on mitigating the related risks. The relevant standards include: ITIL (Information Technology Infrastructure Library) [9], CMMI-ACQ (Capability Maturity Model® Integration for Acquisition) [15], COBIT (Control Objectives for Information and related Technology) [8], IBM RUP (Rational Unified Process) [4], ISO 12207:2008 (Systems and software engineering - Software life cycle processes) [10], IEEE Std 828-2005 (IEEE Standard for Software Configuration Management Plans) [6], IEEE Std 1062-1998 (IEEE Recommended practice for software acquisition) [5], IEEE Std 830-1998 (IEEE Recommended practice for software requirements specifications) [7]. However, as the scope of these standards is rather broad, the client–provider cooperation is covered in terms of general recommendations and no working solutions are proposed. The standards introduce the taxonomy of terms and provide general guidelines for the recommended behaviour of the cooperating parties. They rather restrict themselves to drawing attention to the problem than provide sufficient basis for developing a working solution to the problem. This leaves a room for more specialised solutions which would directly lead to structuring the software client-provider cooperation in organisational, data exchange and procedural terms.

In this article we propose a new method for supporting the software client–provider cooperation. The method is based on the models used to organise the domain of cooperation. We view our method as being complementary to the recommendations of standards and guidelines, providing a specialised implementation of these recommendations. This method is called WIKLIDO and its essence is to use modelling to structure the client–provider cooperation domain. The WIKLIDO models are used to improve the client-provider cooperation within the scope of a software acquisition process. As the scope of WIKLIDO models is restricted to direct interactions between client and provider, the models do not overlap with other models which are possibly developed to cover 'conventional' activities related to software processes, including the internal processes of the cooperating parties. Moreover, because UML is assumed as the modelling language for WILKIDO, it is relatively easy to integrate the WIKLIDO models with other processes, providing they are expressed in a similar language.

The article begins with an introduction to the WIKLIDO method, overviews present experiences with its application and then introduces the criteria to be used to evaluate the method. The evaluation criteria are given as a set of metrics. The metrics were systematically derived with the help of the GQM (Goal, Question, Metrics) method [16]. We briefly present two case studies of WIKLIDO application. These and further case studies will constitute the basis for collecting data to enable us to evaluate the added value of WILKIDO by means of the assumed evaluation criteria.

The WIKLIDO Method

Figure 1 presents WIKLIDO in its application context. It is an UML diagram [14] which shows how the WIKLIDO component is accessible to different categories of users represented by different roles.

The Process Engineer role is responsible for tailoring the WIKLIDO component to the needs of the users (represented by the Model User role). The Model User role is split into two subcategories: Software Provider and Software Client which represent the main actors of the software acquisition process. The objective of Process

Engineer is to specialize the generic WIKLIDO models for a more concrete representation which includes the peculiarities of the specific context associated with the Software Client and Software Provider. A useful metaphor is to understand the Process Engineer as a sort of catalyst which enables the cooperation between Client and Provider.

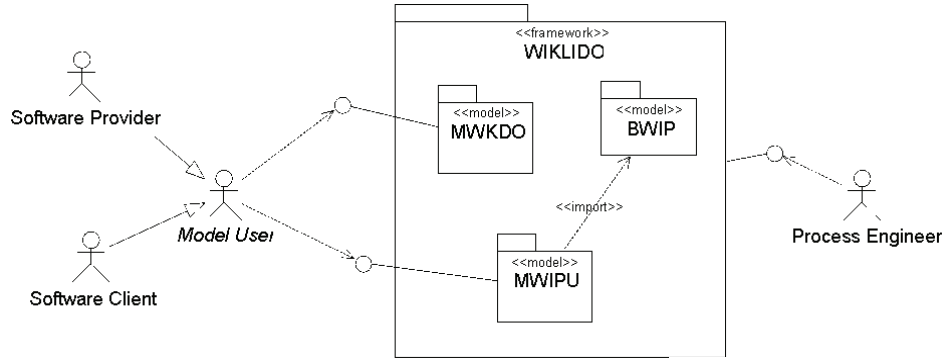

Fig.1. WIKLIDO method in its usage context.

The WIKLIDO components shown in Figure 1 are accessible to Model Users as web sites which are generated by the EPF Composer tool [3]. They include the following elements:

1. BWIP – a knowledge base of the Process Engineer. It contains the knowledge about practices of modelling the client–provider cooperation. The main elements of BWIP are:
   - definitions of fundamental concepts of modeling the client-provider cooperation - the definitions are represented by abstractions derived from the SPEM2 standard [13]. It provides the Process Engineer with the ontology of the modelling domain;
   - a model of key abstractions of the WIKLIDO approach (expressed as a UML class diagram [14]). The model includes objects engaged in the software acquisition processes as well as their relationships. It supports Process Engineer in understanding the WIKLIDO method;
   – patterns of cooperation between the Process Engineer and Model Users – these are definitions of procedures expressed in accordance with the SPEM2 standard [13]. The patterns support both Process Engineer and Model Users in systematic construction of MKWDO model;
   - templates of documents used to acquire and represent information about client-provider cooperation. The templates support Process Engineer in gathering information about client-provider cooperation practices;
   – examples of filled templates demonstrating the best practices of client-provider cooperation;
   – checklists related to client-provider cooperation. They are used to control the scope while collecting information about client-provider cooperation;

2. MWIPU – a model of cooperation of Process Engineer and the Model Users. The model employs the elements (patterns of cooperation and the terms and definitions) imported from BWIP which are made accessible to Model Users to support them while constructing the MWKDO model.

3. MWKDO – a model describing the client–provider cooperation. It is presented in terms of abstractions defined in SPEM2 [13]. The main elements of MWKDO model are patterns of the client–provider cooperation.

The Figures 2 and 3 represent examples of cooperation patterns derived from a case study [2]. The patterns focus on cooperation related to requirements management. The requirements were related to the Patient Flow Module of the eMedSolution system [2]. The patterns were generated with help of the EPF Composer tool [3]. Figure 2 presents an activity diagram representing workflows integrating the cooperating roles. Figure 3 shows an activity detail diagram describing tasks to be performed by the users. The diagram also depicts input and output products of the tasks.

The application of the WIKLIDO method comprises the sequence of steps shown in Figure 4.

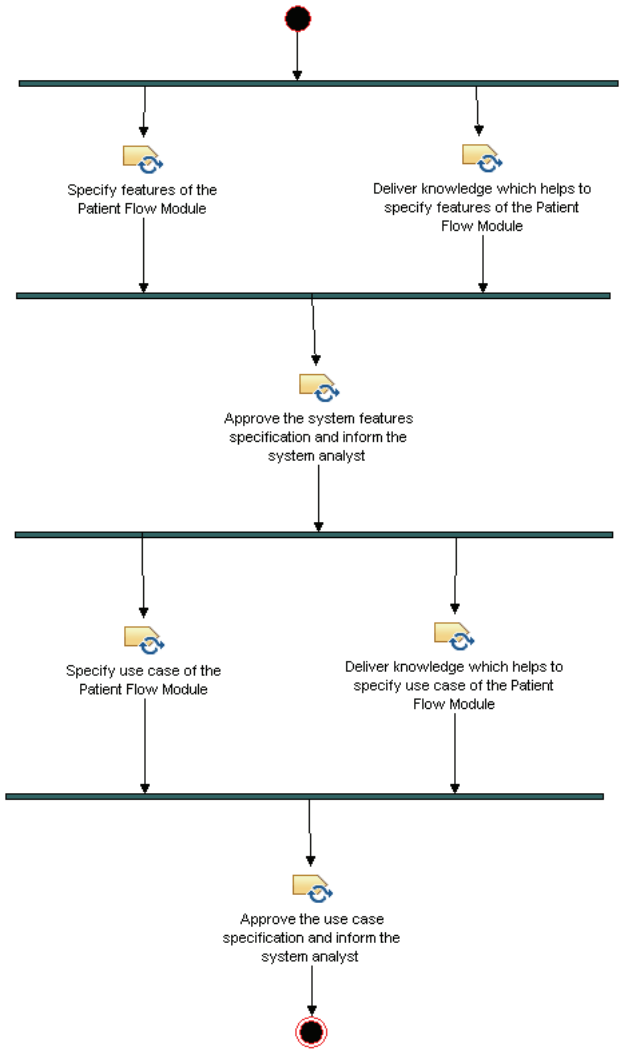

Fig.2. Activity diagram of the pattern of the client–software provider cooperation

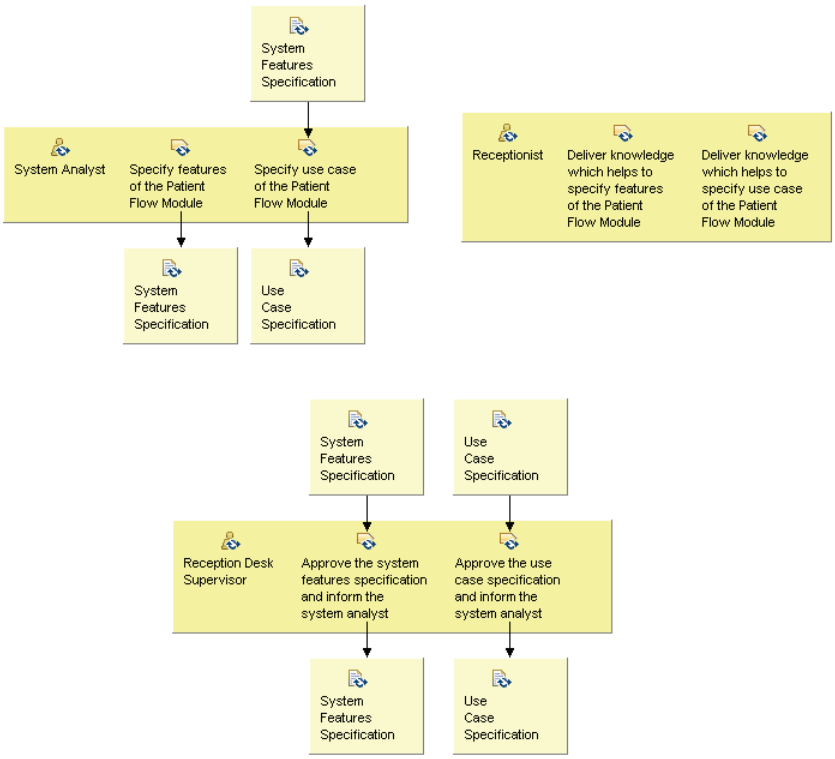

Fig.3. Activity detail diagram of the pattern of the client–software provider cooperation.

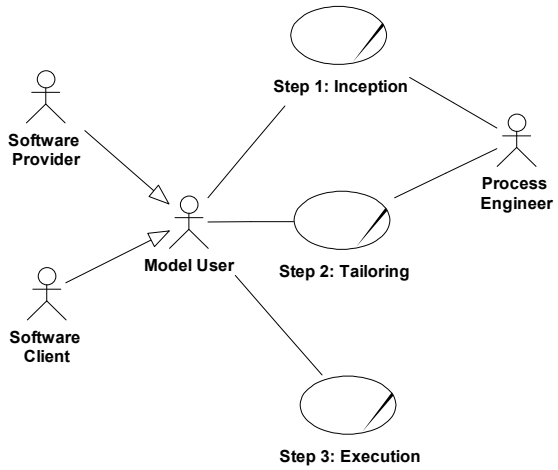

Fig.4. The steps of the WIKLIDO method.

Step 1: Inception - The objective of this step is to define the scope of the WIKLIDO approach in a specific software acquisition project. The execution of this step is coordinated by Process Engineer who is an expert in using WIKLDO. In this step, the Process Engineer is responsible for identification of the roles of cooperating partners, identification of the cooperation areas requiring the WIKLIDO support and selection of the best practices to be applied by client and provider in the software acquisition processes.

Step 2: Tailoring – The objective is to develop the MWKDO model. The execution of the step is coordinated by Process Engineer. With his/her help Model Users specialize the generic models included in BWIP to the sufficient level of detail (choosing this level is one of the important decisions to be taken in this step). The cooperation between Process Engineer and Model Users is driven by the processes represented in MWIPU.

Step 3: Execution – during this step the MWKDO model is used to support the client-provider cooperation in the software acquisition processes.

Case studies

Two case studies were realized with the cooperation of industrial partners in the process of researching the WIKLIDO approach.

In the first case study the MWKDO model was produced for the project aiming at acquisition of a hospital management system. The role of software provider was played by a hospital management software producer (ISH Polska), and the role of client - by the Swissmed Centrum Zdrowia hospital. The subject of acquisition was the eMedSolution system [2]. The produced model was used to support requirements elicitation and their subsequent mapping on the eMedSolution system.

In the second case study the MWKDO model was developed for the project aiming at acquisition of a financial monitoring and control system. The role of software provider was played by COMARCH SA software house and the role of client by the IT department of Polish Ministry of Finance. The subject of acquisition was the SIMIK system. The MWKDO model was built to support cooperation of the partners while performing system analysis and design. The partners agreed that the model integrates recommendations derived from RUP and the best practices of the IT Department of the Ministry.

The case studies were important milestones in the course of developing WIKLIDO and contributed to the method structure and contents. They also provided for the first evaluation of the method in its target environment. The feedback received from the WIKLIDO users involved in the case studies was very positive. The informal interviews with the participants from both, client and provider sides strongly suggested that the method brings a significant added value to the software acquisition processes. This motivated the authors to plan for a more systematic evaluation of the method, based on a carefully designed data collection and metrics.

The overall evaluation plan

The following goals for evaluation of WIKLIDO were selected:

1. Assessment of usability of the BWIP knowledge base.
2. Assessment of usability of the MWIPU model.
3. Assessment of efficiency of the process of developing MWKDO.
4. Assessment of usability of the MWKDO model.
5. Assessment of effectiveness of the process of applying MWKDO.

The metrics for evaluating to which extent the above goals are achieved were systematically derived from the goals using the GQM (Goal Question Metrics) method [16]. In the following sections we present details of the application of GQM and the resulting metrics. Each goal is presented by means of the GQM table, followed by the

questions helping in assessing the goal and the metrics proposed for devising the answer to the question.

The goals 1, 2 and 3 focus on assessing WIKLIDO from the Process Engineer viewpoint. They assess usability of the components supporting Process Engineer and the process of tailoring WIKLIDO to a particular application (by developing MWKDO).

The goals 4, 5 focus on assessing WIKLIDO from the Model Users perspective. They consider the usability of MWKDO and the process of applying MWKDO by the Model Users.

For efficiency assessment the quantitative metrics were used. For usability and effectiveness assessment the qualitative metrics (with the scale: *low, middle, high*) were used. Using the qualitative scale helps in identification of WIKLIDO components which: require improvement (*low*), are acceptable (*middle*) or contribute the surplus value (*high*).

The data are to be collected with the help of questionnaires [1], [17]. For each qualitative metric the Shapiro–Wilk test is to be applied, to deal with the problem of sample sizes [11]. The test helps to verify the hypothesis that samples follow a normally distributed population.

For the details of GQM the interested reader is referred to e.g. [16].

Metrics for assessing BWIP

The goal of assessment is presented in Table 1.

Table 1. The goal of assessing BWIP.

Assess	*Process Engineer Knowledge Base*
in order to	*Evaluate*
What	*Usability*
from the perspective of	*Process Engineer*
in the context of	*an experiment in industrial environment.*

The following questions and metrics were formulated to support this assessment:

Q1: What is the level of difficulty of acquiring BWIP knowledge?

M1: The level of difficulty of acquiring the knowledge of BWIP.

Q2: What level of support is provided by BWIP?

M1: The level of implementation of the client–software provider cooperation templates.

M2: The level of applicability of the process engineer–model user cooperation templates.

M3: The level of detail of the process engineer–model user cooperation templates.

M4: The level of applicability of the checklists.

Metrics for assessing MWIPU

The goal of assessment is presented in Table 2.

Table 2. The goal of assessing MWIPU.

Assess	*Process Engineer–User Model*
in order to	*Evaluate*
what	*Usability*
from the perspective of	*the model user*
in the context of	*an experiment in industrial environment.*

The following questions and metrics were formulated to support this assessment:

Q1: What is the level of difficulty of acquiring MWIPU knowledge?

M1: The level of difficulty of acquiring the knowledge of MWIPU.

Q2: What level of support is provided by MWIPU?

M1: The level of applicability of the process engineer–model user cooperation templates.

M2: The level of completeness of the process engineer–model user cooperation templates.

Metrics for assessing development of MWKDO

The goal of assessment is presented in Table 3.

Table 3. The goal of assessing development of MWKDO.

Assess	*Process of developing MWKDO*
in order to	*Evaluate*
what	*Efficiency*
from the perspective of	*Process engineer and the MWIPU model user*
in the context of	*an experiment in industrial environment.*

The following questions and metrics were formulated to support this assessment:

Q1: What is the effort of modelling the client–software provider cooperation?

M1: The effort of the process engineer involved in modelling the client–software provider cooperation.

M2: The effort of the MWKDO users involved in specifying the client–software provider cooperation.

M3: The effort of the verification of the modelling results by the MWKDO users.

M4: The effort of domain training of the process engineer needed to implement the WIKLIDO method.

M5: The number of the MWKDO users involved in the process of modelling the client–software provider cooperation.

Metrics for assessing MWKDO

The goal of assessment is presented in Table 4.

Table 4. The goal of assessing the client–software provider cooperation model.

Assess	*Client–Software Provider Cooperation Model*
in order to	*Evaluate*
what	*Usability*
from the perspective of	*the MWKDO model user*
in the context of	*an experiment in industrial environment.*

The following questions and metrics were formulated to support this assessment:

Q1: What is the level of difficulty of acquiring MWKDO knowledge?

M1: The level of difficulty of acquiring the knowledge of MWKDO by its users.

Q2: What level of support is provided by MWKDO for the software client?

M1: The level of applicability of the client–software provider cooperation templates.

M2: The level of increasing support for analysing the potential of software acquisition.

M3: The level of understanding of its role in the client–software provider cooperation processes.

M4: The level of increasing support for harmonizing the client cooperation processes with its other processes, which are normally required.

M5: The level of increasing the level of detail of the key cooperation processes.

M6: The level of increasing support for the elaboration of the concept of cooperation with the software provider.

M7: The level of increasing support for the continuous improvement of the cooperation with the software provider.

M8: The level of probability of reusing MWKDO in another software acquisition projects.

Q3: What level of support is provided by MWKDO for the software provider?

M1: The level of applicability of the client–software provider cooperation patterns.

M2: The level of increasing support for analysing the potential of software delivery.
M3: The level of understanding of its role in the client–software provider cooperation processes.
M4: The level of increasing support for harmonizing the provider cooperation processes with its other processes, which are normally required.
M5: The level of increasing the level of detail of the key cooperation processes.
M6: The level of increasing support for the elaboration of the concept of cooperation with the client.
M7: The level of increasing support for the continuous improvement of the cooperation with the client.
M8: The level of probability of reusing MWKDO in another software acquisition projects.

Q4: Does MWKDO support the identification of the risks of cooperation?

M1: The number of risks of the cooperation, identified by the MWKDO model, for the client.
M2: The number of risks of the cooperation, identified by the MWKDO model, for the software provider.

Metrics for assessing the process of applying MWKDO

The goal of assessment is presented in Table 5.

Table 5. The goal of assessing the process of applying MWKDO.

Assess	*The process of applying MWKDO.*
in order to	*Evaluate*
what	*Effectiveness*
from the perspective of	*the MWKDO model user*
in the context of	*an experiment in industrial environment.*

The following questions and metrics were formulated to support this assessment:

Q1: What is the efficiency of the cooperation based on MWKDO from the perspective of the client?

M1: The level of increasing the client's confidence in the software provider.
M2: The level of formalizing the client–software provider cooperation.
M3: The level of increasing support for documenting the cooperation process.
M4: The level of increasing support for the communication with the software provider.
M5: The level of increasing support for solving conflicts with the software provider.
M6: The level of increasing support for the acquisition of a software provider having the greatest potential of effective cooperation.

Q2: What is the efficiency of the cooperation based on MWKDO from the perspective of the software provider?

M1: The level of increasing the software provider's confidence in the client.
M2: The level of formalizing the client–software provider cooperation.
M3: The level of increasing support for documenting the cooperation process.
M4: The level of increasing support for the communication with the client.
M5: The level of increasing support for solving conflicts with the client.

Conclusion

WIKLIDO addresses a problem of particular significance for the success of software acquisition processes. Although the initial feedback from the application of the method was very positive, more experimental research is necessary to assess the method in more objective terms. Such experiments are planned to be carried out in the industrial environment.

The paper presented a brief introduction to WIKLIDO and then explained the criteria to be used during experimental evaluation of the method.

The proposed metrics are mostly qualitative which reflects that client–provider cooperation involves various soft aspects. The basic mechanism for gathering experimental data are questionnaires. The data will be collected in a series of real life experiments.

REFERENCES

[1] Babbie E., The Basic of Social Research, Third Edition, Wadsworth Publishing (2005)
[2] Dąbrowski P., Jakubowski M., Górski J., Wyrzykowski K., Supporting client-provider cooperation Turing acquisition of eMedSolution hospital management system for the Swissmed hospital (in Polish), Ogólnopolski Przegląd Medyczny, No. 12 (2007)
[3] Eclipse Process Framework Project, EPF Composer (2009), http://www.eclipse.org/epf/
[4] IBM, Rational Unified Process v7.01 (2008), http://www-01.ibm.com/software/awdtools/rup/
[5] IEEE Std 1062-1998 Recommended practice for software acquisition (1998), http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?reload=true&tp=&isnumber=16011&arnumber=741938&punumber=5977
[6] IEEE Std 828-2005 Standard for Software Configuration Management Plans (2005), http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?tp=&isnumber=32241&arnumber=1502775&punumber=10048
[7] IEEE Std 830-1998 Recommended practice for software requirements specifications (1998), http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?tp=&isnumber=15571&arnumber=720574&punumber=5841
[8] Information Systems Audit and Control Association, Control Objectives for Information and Related Technology 4.1 (2007), http://www.isaca.org/Content/NavigationMenu/Members_and_Leaders/COBIT6/Obtain_COBIT/Obtain_COBIT.htm
[9] Information Technology Infrastructure Library v3 (2007), http://www.itil-officialsite.com/home/home.asp
[10] ISO 12207:2008 Systems and software engineering - Software life cycle processes (2008), http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=43447
[11] Madeyski L., Test-Driven Development. An Empirical Evaluation of Agile Practice, Springer-Verlag Berlin Heidelberg (2010)
[12] McConnell S., Rapid Development, Microsoft Press (1996)
[13] OMG, Software Process Engineering Metamodel Specification, Version 2.0 (2005), http://www.omg.org/docs/ptc/07-11-01.pdf
[14] OMG, Unified Modelling Language Superstructure, Version 2.2 (2009), http://www.omg.org/spec/UML/2.2/Superstructure/PDF/
[15] SEI, Capability Maturity Model® Integration for Acquisition Version 1.2 (2007), http://www.sei.cmu.edu/pub/documents/07.reports/07tr017.pdf
[16] Solingen R. van, Berghout E., The Goal/Question/Metric method: A practical guide for quality improvement of software development, McGraw-Hill (1999)
[17] Thomas W., Mail Surveys. Improving the Quality, Sage Publication (1995)
[18] Wyrzykowski K., Górski J., Modelling client-provider cooperation In software acquisition processes (In Polish). I Małopolska Konferencja Zarządzanie Projektem. Kraków 2007

***Authors**: prof. dr hab. inż. Janusz Górski, mgr Krzysztof Wyrzykowski: Gdansk University of Technology, Software Engineering Department, ul. Gabriela Narutowicza 11/12, 80-952 Gdańsk; {jango,kwyrzyk}@eti.pg.gda.pl.*