World Academy of Science, Engineering and Technology
International Journal of Computer and Systems Engineering
Vol:6, No:11, 2012

Software Development Processes Maturity versus Software Processes and Products Measurement

Beata Czarnacka-Chrobot

***Abstract*—**Unsatisfactory effectiveness of software systems development and enhancement projects is one of the main reasons why in software engineering there are attempts being made to use experiences coming from other engineering disciplines. In spite of specificity of software product and process a belief had come out that the execution of software could be more effective if these objects were subject to measurement – as it is true in other engineering disciplines for which measurement is an immanent feature. Thus objective and reliable approaches to the measurement of software processes and products have been sought in software engineering for several dozens of years already. This may be proved, among others, by the current version of CMMI for Development model. This paper is aimed at analyzing the approach to the software processes and products measurement proposed in the latest version of this very model, indicating growing acceptance for this issue in software engineering.

***Keywords*—**CMMI for Development (1.3), ISO/IEC standards, measurement and analysis process area, software process measurement, software product measurement.

I. Introduction

AS indicated by the studies of Software Engineering Institute (SEI) (see e.g., [1], [2], [3]), measurement of software processes and products remains a neglected area of software engineering in which „there is still a significant gap between the current and desired state of measurement practice" [1, p. 29]. Results of these studies are in agreement with the conclusions coming from the surveys having been carried out by the author of this paper among Polish developers of dedicated software systems that unequivocally indicate that methodic approaches to the software product and process measurement are still being used relatively rarely (for more details see [4]). While software product and process measurement is an essential factor promoting effective management of software development and therefore – effective execution of software systems development and enhancement projects [5, Part 6]. It is of significance that hardly can be overestimated since such projects cannot boast about high effectiveness: as indicated by the results of the Standish Group study, success rate for such projects has never gone beyond 37% [6]. This US institution estimates that in case of more than 40% of software application development and enhancement projects the planned time of product delivery is exceeded by nearly 80% on average and the estimated budget - by approx. 55% on average [7]. This leads to the substantial financial losses, on a worldwide scale estimated to be hundreds of billions of dollars yearly, sometimes making even more than half the funds being invested in such projects. That's why „there is still much that needs to be done so that organizations use measurement effectively to improve their [software – BCC] processes, products, and services." [1, p. 29].

No wonder that the measurement of software processes and products is an area of software engineering, which L. Buglione and A. Abran not long ago used to describe as „rather immature in terms of knowledge maturity" [8, p. 84]. In their opinion, this area does not yet meet criteria of the so-called rule of general acceptance, formulated by the Project Management Institute (PMI) in PMBOK (Project Management Body of Knowledge) and then adopted by the IEEE (Institute of Electrical and Electronics Engineers) Computer Society in SWEBOK (Software Engineering Body of Knowledge). According to this rule: "generally accepted means that the knowledge and practices described are applicable to most projects most of the time, and that there is widespread consensus about their value and usefulness" [9, p. 3]. According to these authors, a factor promoting change of this *status quo* is developing formalized approaches to the software process and product measurement that would constitute source of knowledge allowing the above mentioned rule to be accomplished over time also by this specific area of software engineering.

That's why over the last couple of years the works have been intensified that aimed at describing the best practices concerning software process and product measurement within the existing models, including first of all Capability Maturity Model Integration (CMMI) for Development (CMMI-DEV) [10], or in the form of new standards, first of all ISO/IEC (International Organization for Standardization/International Electrotechnical Commission) standards, relating to this subject matter to various, also very detailed one, degree (see e.g., [11], [12], [13], [14]). As a result they gave rise to a variety of formal approaches, treating this very area from different perspectives, and *de facto* being the effect of several dozens of years of pursuit and development of sufficiently objective and reliable methods of proceedings in the discussed field and filling the gap in the theory as well as in the practice of software engineering. It is worth noting that to a greater and greater degree they concentrate on measurement of software

Beata Czarnacka-Chrobot is with the Department of Business Informatics, Warsaw School of Economics, Al. Niepodleglosci 164, 02-554, Warsaw, Poland (e-mail: bczarn@sgh.waw.pl).

product, which as a matter of fact is a result of software processes being undertaken, while only recently such approaches were criticized for concentrating almost entirely on the process alone.

Due to a large substantive differentiation of various approaches to measurement, in the execution of this process in practice it is worth to use model of proceedings that not only recommends execution of measurement but also offers general set of guidelines to the users, certain scheme of proceedings assumed to simplify and/or sort this process. As an example of model that could be helpful in the execution of the measurement process one should present CMMI-DEV model and this being due to at least four reasons: (1) on the basis of the development of this model one may observe evolution of the approach to the issue of measurement in software engineering, (2) in this model the effects that can be obtained thanks to measurement are clearly appreciated, (3) approach to measurement, characteristic of this model, brings very beneficial measurable results, (4) this model is considered useful and relatively often used in software engineering practice, as suggested by the SEI studies [1, p. 18].

Hence the goal of this paper is the analysis of the approach to the software products and processes measurement proposed in the latest version of the CMMI for Development model. Therefore in the section 2 the paper presents general classification of formal approaches to measurement in software engineering. Section 3 is devoted to synthetic characteristics of the CMMI-DEV model while section 4 – of the so-called measurement and analysis process area of the CMMI-DEV model. Section 5 features conclusions concerning mainly evaluation of the described approach usage in practice from the perspective of software process and product measurement.

II. Classification of Formal Approaches to Measurement in Software Engineering

On the basis of the classification proposed by K. Richins [15, p. 3.259], formal approaches to the software process and product measurement may be divided into (for more details see [11]):

- Approaches officially recognized as international standards, which to lower or higher degree relate to measurement. Since the measurement processes in software engineering are being attributed higher and higher significance, a considerable progress has taken place over the last few years in the standardization of approaches to this issue, which may be proved, among others, by the international norms having been published jointly by the ISO and IEC. The ISO/IEC norms concerning measurement in software engineering may be specified by bringing it to the following categories characterized by the increasing level of detailness:
  1) Standards including the measurement, concerning:
     - measurement process (ISO/IEC norms: 12207, 15288, 14598)
     - software process measurement (ISO/IEC norms: 90003, 15504)
     - software product measurement (ISO/IEC norms: 25000, 25001, 25030).
  2) Standards dedicated to the measurement, concerning:
     - measurement process (ISO/IEC 15939 norm)
     - software product measurement (ISO/IEC norms: 9126, 25020, 25021, 14143).
  3) Standards dedicated to particular methods of software product functional size measurement (ISO/IEC norms: 20926, 20968, 24570, 19761, 29881).
- Approaches setting directions for proceedings thus ensuring proper course of the measurement process, which do not hold official status of international standard yet as a rule are successfully used in practice in given application areas. Development of this type of structuralized approaches is being supported by the initiatives such as e.g., Software Engineering Measurement and Analysis (SEMA) Group, undertaken within the SEI, or Measurement Working Group (MWG) performing within the International Council on System Engineering (INCOSE), as well as by various government directives. As indicated by the SEI studies [1, p. 18], approach being used in practice to the largest extent in this type of measurement proves to be the so-called Measurement and Analysis (MA) Process Area (PA) of the CMMI-DEV model [10, pp. 175-190].

III. Synthetic Characteristics of the CMMI for Development Model

According to N. Fenton, considerable influence on the awareness of the significance of measurement and its understanding and therefore on the use of measurements in the practice of software engineering has been made by the Software Process Assessment/Software Process Improvement (SPA/SPI) model called Capability Maturity Model (CMM), developed by the SEI in the 1980s [16, pp. 88, 109-110]. This results from the fact that in this particular approach as well as in other approaches that had stemmed from it, being relatively popular due to commercial motivation (e.g. US Department of Defense requires companies putting for their contracts to be minimum on the maturity level 3 of the CMMI model), measurement is considered significant to achieving subsequent, higher levels of maturity or capability.

The CMM model was commissioned by the US Department of Defense in response to the problems concerning effectiveness of the execution of software ordered for this institution. It was supposed to be used for the assessment of the usefulness of its potential providers by qualifying them into one of five levels of maturity [16, p. 109]. In spite of its popularity it had not avoided the criticism though – first of all due to a five-degree scale and methods of assessment used as well as lack of fully convincing evidence that organizations from higher level of maturity can develop better software products [16, p. 111]. Hence attempts had been made to develop other models, including various ISO/IEC standards (see e.g., [11], [12], [13], [14]). However, works on enhancing the rules introduced by the CMM had been going and are still going on, and one of the main manifestations of this was replacing a group of models built on it with the integrated

CMMI (CMM Integration) model and its evolution that followed.

This integrated model, in its 1.1 version of 2002, allowed for adjusting to the needs of given organization – as it made it possible to assess both particular processes being accomplished within it as well as an organization as a whole. These results from the fact it is available in two variants: continuous (CMMI Continuous Representation - CMMI-CR) and staged (CMMI Staged Representation - CMMI-SR). Although version CMMI 1.3 is the one that is valid at the moment, both representations, however, are included to it [10].

The structure of the currently valid version of CMMI is based on the concept of the so-called constellation, understood as „a collection of CMMI components that are used to construct models, training materials, and appraisal related documents for an area of interest" [10, p. 450]. At the moment three areas of interest are being taken into consideration: product development, product acquisition and services related to a product. Version 1.3 of CMMI for Acquisition [17], CMMI for Development ([10], [18]), and CMMI for Services [19] were released in November 2010. First of the constellations developed by SEI, and at the same time being most interesting from the point of view of the subject matter discussed in this paper, is CMMI for Development.

CMMI for Development is a set of the so-called best practices that serve as a point of reference for activities regarding product development and maintenance, mostly in software engineering (but also e.g., in system engineering or in hardware engineering). These practices cover the entire product's lifecycle: from the conception to delivery and maintenance, while particular emphasis is being put on works that are indispensable to develop and maintain complete product.

Staged representation of the CMMI-DEV model is designed for comprehensive assessment of organization's software process and this by qualifying organization to one of the five *maturity levels* that are briefly characterized in Table I.

Analysis of the conditions which an organization must meet in order to achieve subsequent levels of maturity indicates that in the CMMI-DEV model a considerable importance is given to measurement of software processes and products: the higher the maturity level, the stronger orientation towards quantitative approach. This means that what is of fundamental significance are metrics while selection of objects and attributes to be measured depends on the information available on each of the levels.

All maturity levels in the CMMI-SR model, except lowest one, are characterized by the specified set of the so-called process areas (PA). Process area is understood as „a cluster of related practices in an area that, when implemented collectively, satisfies a set of goals considered important for making improvement in that area" [10, p. 449]. Organization being assessed is qualified to the given maturity level provided that it satisfies all goals determined by all process areas that are characteristic not only of given level but of lower levels as well. All process areas are common to the staged as well as continuous representation whereas in the CMMI-SR variant they are organized according to maturity levels while in the CMMI-CR variant the same process areas were grouped according to the so-called category of process areas, comprising categories such as: process management, project management, engineering, and support (for more details see [10, p. 33]).

TABLE I
SYNTHETIC CHARACTERISTICS OF MATURITY LEVELS IN THE CMMI-DEV MODEL – STAGED REPRESENTATION

Name of maturity level	Synthetic description
Level 1: Initial	In the organization the procedures according to which the projects are performed either do not exist at all or they do exist but are not in fact performed – hence the processes usually proceed chaotically and *ad hoc* while success depends on the competence of employees and their dedication. Such organization often provides functioning software products yet as a rule with time and budget overruns.
Level 2: Managed	Organization is obliged to implement process management procedures with regard to: project planning, managing its requirements and configuration, procedure of ensuring quality of processes and products as well as **measurement and analysis**. However there is no formal definition of processes. Project management is orientated towards monitoring functionality, cost and time of execution. Procedures carried out within particular projects may differ significantly.
Level 3: Defined	Processes are defined in the form of the set of formal general-organizational standards being executed and enhanced that allows for attaining consistency within the entire organization. Projects are performed according to organization's own defined processes which, however, are developed on the basis of adaptation procedures based on general-organizational set of standard processes. Main differences between this and the preceding level consist in the range of set standards, accuracy of processes description and in **the use of measures**: on the discussed level, processes are managed with the use of detailed **measures of the process and product.**
Level 4: Quantitatively managed	Organization defines **quantitative objectives** with regard to processes and quality of products which it then follows and monitors within the projects performed, thanks to which it can keep enhancing them; it also uses them as criteria in the processes management. Quantitatively defined objectives are based first of all on client's needs. Hence organization monitors **measures of process and quality of product** that are collected in general-organizational repository designed for this purpose. For selected subprocesses detailed measures of process execution are analyzed with the use of statistical methods in order to identify *extraordinary* causes of differences, their improvement and preventing them in the future. Key difference between this and the preceding level is **quantitative predictability** of processes execution – on the defined level, processes are usually predictable only qualitatively.
Level 5: Optimizing	Processes corresponding with projects executed by an organization are being constantly optimized on the basis of **quantitative management** of the *expected* differences in their execution. Technological innovations and enhancements are being introduced continuously. **Quantitative objectives** of processes enhancement are determined, reviewed and corrected so they are adapted to the changes in business objectives of an organization.

Source: Author's own analysis based on: [10, p. 27-29].

Continuous representation of CMMI-DEV model is designed for autonomous assessment of particular process areas that are classified into one of four *capability levels*, concisely characterized in Table II.

TABLE II
SYNTHETIC CHARACTERISTICS OF CAPABILITY LEVELS IN THE CMMI-DEV MODEL – CONTINUOUS REPRESENTATION

Name of capability level	Synthetic description
Level 0: Incomplete	A process that either is not performed or is partially performed. One or more of the specific goals of the process area are not satisfied and no generic goals exist for this level since there is no reason to institutionalize a partially performed process.
Level 1: Performed	A process that accomplishes the needed work to produce work products; the specific goals of the process area are satisfied. Although capability level 1 results in important improvements, those improvements can be lost over time if they are not institutionalized.
Level 2: Managed	A performed process that is planned and executed in accordance with policy; employs skilled people having adequate resources to produce controlled outputs; involves relevant stakeholders; is monitored, controlled, and reviewed; and is evaluated for adherence to its process description.
Level 3: Defined	A managed process that is tailored from the organization's set of standard processes according to the organization's tailoring guidelines; has a maintained process description; and contributes process related experiences to the organizational process assets. A critical distinction between capability levels 2 and 3 is the scope of standards, process descriptions, and procedures. At capability level 2, the standards, process descriptions, and procedures can be quite different in each specific instance of the process (e.g., on a particular project). Another critical distinction is that at capability level 3 processes are typically described more rigorously than at capability level 2. A defined process clearly states the purpose, inputs, entry criteria, activities, roles, measures, verification steps, outputs, and exit criteria. At capability level 3, processes are managed more proactively using an understanding of the interrelationships of the process activities and detailed measures of the process and its work products.

Source: Source: [10, pp. 24-25].

Table III compares the four capability levels versus five maturity levels.

TABLE III
COMPARISON OF CAPABILITY AND MATURITY LEVELS

Level	Capability Levels (CMMI-CR)	Maturity Levels (CMMI-SR)
Level 0	Incomplete	
Level 1	Performed	Initial
Level 2	Managed	Managed
Level 3	Defined	Defined
Level 4		Quantitatively Managed
Level 5		Optimizing

Source: [10, p. 23].

In the current version of the CMMI-DEV there are 22 process areas that can be isolated (for more details see [10, p. 11]). Each of the PA is assigned one or several the so-called specific goals which on the other hand are assigned specific practices. Version 1.1 of the model was already widened with general goals and practices which are not assigned to any individual PA as the same goal and its practices are linked with many PA. Whether given area is executed is proved by achieving all specific goals assigned to it along with general goals that are linked with them while achieving such goals in case of the set of process areas characterizing given maturity level (and lower levels as well) means achieving this very level. In other words, maturity level (in CMMI-SR) is determined on the basis of the achieved specific and general goals linked with the each predefined set of process areas while capability level (in CMMI-CR) – on the basis of general goals linked with specific process area.

IV. MEASUREMENT AND ANALYSIS PROCESS AREA OF THE CMMI FOR DEVELOPMENT MODEL

One of the 22 process areas isolated in CMMI-DEV model is an area of Measurement and Analysis (MA) [10, pp. 175-190]. Developing of MA area was made in close cooperation with specialists from Practical Software and Systems Measurement Support Center, ISO and International Function Point Users Group (IFPUG). In CMMI-SR representation it was assigned to the maturity level 2 (see Table I) while in CMMI-CR representation it was classified into the category of support process areas. Isolating this area explicitly proves to managers the execution of the measurement process in order to enhance software development processes is necessary – predecessors of CMMI models lacked such a clear, autonomous and consistent approach while scope of the discussed PA is at the moment significantly larger.

The goal of the measurement and analysis process is „to develop and sustain a measurement capability used to support management information needs" [10, p. 175]. Therefore integration of activities covered by measurement and analysis into one process area is aimed to support:

- objective planning, including project estimation;
- monitoring factual course of the project with regard to its correspondence with plans and goals;
- identifying and solving various problems concerning the measured process that arise;
- developing bases designed for including measurement into subsequent processes.

The MA PA involves the following activities [10, p. 175]: (1) specifying goals of measurement and analysis so that they are aligned with the identified information needs and project, organizational, or business objectives; (2) specifying measures, analysis techniques, and mechanisms for data collection, data storage, reporting, and feedback; (3) implementing the analysis techniques and mechanisms for data collection, data reporting, and feedback; (4) providing objective results that can be used in making informed decisions and taking appropriate corrective action.

Measurement and analysis process area supports all process areas of the CMMI-DEV model by providing set of practices specific to it; those practices being helpful in sorting measurement needs and goals in a way so that objective results of measurement, necessary to make rational decisions and right corrective action mostly with regard to projects and therefore to organizations, would be achieved. Specific

practices (SP) are grouped with regard to two specific goals (SG) of the discussed PA [10, p. 176]:

- SG 1: Align measurement and analysis activities, including: SP 1.1: Establish measurement objectives, SP 1.2: Specify Measures, SP 1.3: Specify data collection and storage procedures, SP 1.4 Specify analysis procedures;
- SG 2: Provide measurement results, including: SP 2.1: Obtain measurement data, SP 2.2: Analyze measurement data, SP 2.3: Store data and results, SP 2.4: Communicate results.

Execution of practices specific to the first goal enables to get answer to the question about measurement goal, its subject, the way in which it is going to proceed as well as the way of using data obtained. As a result these activities allow for developing consistent measurement and analysis plan. As for the specific practices assigned to the second goal – they simply need to be performed [20, p. 21]. It is also worth stressing that in the CMMI-DEV model there are no specific methods indicated with the help of which measurement of particular attributes of software processes and products should be carried out; instead only framework structure of such proceedings is determined. Analogous situation applies to other process areas. Thus from this point of view CMMI-DEV model is so general that it can be called metamodel – what's important is that the goals of the given PA, both specific and general, are achieved.

Measurement and analysis process area is also associated with a certain set of the above mentioned general goals of the CMMI-DEV model, achieving of which is enabled by effective execution of general practices (see [10, pp. 63-125]). D. Goldenson and others [20, pp. 21-23] used to stress that general practices on the one hand serve to institutionalize measurement and analysis process as well as to enhance capabilities with which these activities are performed in the course of a product's lifecycle, on the other hand, however, they indicate general practices being strongly dependent on measurement. Among them are first of all those concerning: monitoring and control of a process, gathering information, supporting enhancement, determining quantitative goals for the process, stabilizing subprocesses, ensuring continuous enhancement of processes and correction of the main causes of problems. Additionally, with the use of correctly defined and executed measurement process one may prove profitability of SPA/SPI processes and therefore also of activities supporting them, into which the discussed area belongs. What's more, every process area is to a lower or greater degree dependent on the correct execution of measurement and analysis that help to enhance these areas significantly while practices of MA area go on in different ways as an organization achieves goals of the subsequent PA of the CMMI-DEV model. Process areas of the CMMI-DEV model, explicitly linked with measurement and analysis, are displayed in Table IV.

As indicated in Table IV, process areas linked with MA PA belong first of all to the category of project management (although MA PA itself belongs to the category of support process areas). Most among those PA are assigned to the maturity level 2 yet MA supports also process areas of the maturity level 3 and 4.

TABLE IV
PROCESS AREAS OF THE CMMI-DEV MODEL RELATED TO THE MA PA

Process area	Maturity level (CMMI-SR)	Category of process areas (CMMI-CR)
Project Planning PA	2	Project management
Requirements Management PA	2	Project management
Project Monitoring and Control PA	2	Project management
Quantitative Project Management PA	4	Project management
Configuration Management PA	2	Support
Requirements Development PA	3	Engineering
Organizational Process Definition PA	3	Process management

Source: Author's own analysis based on [10, pp. 33, 176].

One of the process areas being strongly dependent on measurement and analysis is project planning. Its goal is to determine and maintain plans that define project-related activities. This area covers development and monitoring of agreed and approved project plan with adequate engagement of its stakeholders. Planning begins with defining requirements concerning product and project and it comprises first of all assessment of their attributes, determining necessary resources, developing works schedule, and identification and analysis of the project risk. These activities, as a rule carried out in an iterative way, lead to the development of a project plan that constitutes basis for the execution and control of project-related activities in accordance with the needs agreed with a client. It is often being revised which results from high changeability of client's requirements, inaccurate estimates, corrective activities and changeability in the course of the project process itself (for more details see [10, pp. 281-299]).

Another process area being strongly correlated with measurement and analysis is quantitative project management. The goal of this area is to provide quantitative management of defined process corresponding with the project in a way so that as a result of its execution the required quantitative goals concerning process quality and execution are achieved. These goals should be determined beforehand; in addition, within its framework one should: identify right subprocesses making up the defined process – based on benchmarking data on the process execution; choose those subprocesses that are to be managed using the selected measures, methods and statistical-analytical techniques; monitor project in order to keep track of how it adapts at a given time to quantitative requirements, with particular attention given to subprocesses managed with the use of the selected statistical methods and register data on quality management in the organization's repository designed for measurements. Quantitative goals regarding process execution concern both measures of process attributes (e.g., work effort, time of execution) as well as measures of product attributes (e.g., reliability). Key characteristic of quantitative project management is adequate credibility of estimates being obtained as well as ability to identify threat to the set quantitative goals coming out throughout project execution (for more details see [10, pp. 307-324]).

V. Concluding Remarks

As indicated by the presented considerations, in the current version of the CMMI-DEV model (1.3) great importance is being attached to the measurement of software processes and products. This is one of the fundamental reasons why in the organizations on the high maturity level of the CMMI model ([21, pp. 3-30], [22], [23]):

- costs and time of project execution is estimated more accurately thanks to the proper collecting of reliable benchmarking data: organizations on the maturity level 5 practically do not exceed estimates whereas organizations on the maturity level 1 go beyond the estimated time by 150% on average, and beyond the estimated costs – by nearly 200% o average;
- quality of end products increases due to the lower number of faults and this being thanks to control and assessment of the intermediate products quality being made at the earliest stages of product's lifecycle which allows for correcting them on a current basis;
- due to lower number of faults in intermediate products total costs of improving their bad quality decline: average cost of enhancing faults in organizations on the level 5 is approx. 4% of the total software development cost while in organizations on the initial level it amounts to over 50% of such costs;
- due to decline in the costs of enhancing bad quality of products, the costs of software development decrease too: average cost of developing one function is more than 3 times lower in organizations on the highest level comparing to that cost in organizations on the lowest level;
- due to decrease of the above listed costs per unit, work productivity increases;
- due to lower number of faults in products and thus reduction of time designed for correcting them, total time of product execution gets shorter too.

Studies by C.A. Dekker and B. Emmons [24, p. 22] indicate that what for companies on the maturity level 1 is one of the fundamental causes of difficulty in achieving the level 2 is lack of measurement procedures while measure of the size of products to be delivered is one of the measures that are leaved out the most often. On the other hand, studies carried out by M. Brown and D. Goldenson [25, pp. 131-133, 136] indicate that in practice problems with measurement of attributes of software products and processes occur regardless of the organization's maturity level: first – they are being noted on every level, second – their structure is very similar for each of the levels. At the top two maturity levels, comparing to the defined level one declares somewhat more difficulties with product measurement and less with process measurement. With regard to the attributes of those objects, high or medium level of difficulties with measurement was declared, depending on the attribute, by 70% to even 90% of the surveyed CIOs (Chief Information Officer), while those bigger difficulties concern, among others, measurement of the project risk, level of client's satisfaction, product quality, however estimation of its execution costs is included here as well.

The SEI data indicate that to achieve the highest maturity level an organization needs 7.5 years on average. Each organization, however, has to compare benefits coming from the implementation of the CMMI-DEV model against its costs. Main objection, however being raised to the discussed SEI proposal is its complexity and inflexibility and therefore need to create, often very extensive, bureaucratic apparatus to control application of CMMI practices. As a result use of the model requires considerable financial means which for small and medium sized organizations is a factor that significantly limits the possibility of using it. In addition, implementation of the model at first slows down the software process which often makes users feel reluctant to use it. Also, because of the complexity of the model, one may sometimes in practice lose sight of what its factual purpose is: high functionality and quality of software product that are to be delivered as a result of the execution of effective process, in turn getting theoretical consistency of the process with model maturity. What's more, there is still no convincing evidence of the existence of correlation between quality of process and quality of product and this is what is assumed in the SEI: "The quality of a system or product is highly influenced by the quality of the process used to develop and maintain it" [10, p. 5]. This premise, however, is widely accepted in software engineering environment, including the ISO/IEC as well.

Due to the above reasons solutions have appeared that were meant to be a counterweight to the excessive, according to some, formalization of the CMMI model: agile, adaptive methodologies, whose general assumption is effective development of properly functioning software thanks to concentrating on purely constructive activities and minimization of other activities. In these approaches measurement of software process and product key attributes too is of significance yet as strictly defined and constant are treated here costs and time of execution within which functioning products should be delivered, even if they do not cover complete functionality whereas in traditional approaches product size is relatively least flexible variable (this, however, does not apply to all cases and depends on client's priorities). However, in agile methodologies one resigns from benefits coming from disciplined and systematic use of good practices, or from the transfer of knowledge and skills between projects whose effect is that, among others - as indicated by the studies of B. Clark [26, p. 25] – change of maturity level by one level causes reduction of project's work costs by 4% to 11% on average. Thus some compromises have been sought, in case of which, however, weight is usually being attached too to the measurement as a method leading to high functionality and quality of software product as well as practical effectiveness of software process.

Summing up it should be stated that considerable progress in the development of the new formal approaches to measurement has taken place in recent years. On the basis of this observation Buglione and Abran used to stress that this constitutes „strong evidence of increased "generally accepted" recognition for a number of software measurement topics" [8, p. 91]. These works, however, can hardly be considered

finished – software engineering still faces many challenges related to measurement of software processes and products, becoming of special significance in recession and post-recession time alike.

REFERENCES

[1] M. Kasunic, "The state of software measurement practice: results of 2006 survey", Software Engineering Institute, Carnegie Mellon University, Pittsburgh, 2006, pp. 1-67.
[2] D. Goldenson, "Understanding CMMI measurement capabilities & impact on performance: results from the 2007 SEI state of the measurement practice survey", CMMI Technology Conference, November 14, 2007; http://www.sei.cmu.edu/library/assets/measurement-survey2007.pdf (6.06.2012).
[3] D. Goldenson, J. McCurley, and R. Stoddard, "Use and organizational effects of measurement and analysis in high maturity organizations: results from the 2008 SEI state of measurement and analysis practice surveys," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, Technical Report CMU/SEI-2008-TR-024, 2009; http://www.sei.cmu.edu/library/abstracts/reports/08tr024.cfm (6.06.2012).
[4] B. Czarnacka-Chrobot, "Analysis of the functional size measurement methods usage by Polish business software systems providers", in Software Process and Product Measurement, A. Abran, R. Braungarten, R. Dumke, J. Cuadrado-Gallego, J. Brunekreef, Eds., Proc. of the 3rd International Conference IWSM/Mensura 2009, Lecture Notes in Computer Science, vol. 5891, Springer-Verlag, Berlin-Heidelberg, 2009, pp. 17–34.
[5] ISO/IEC 14143 Information Technology – Software measurement – Functional size measurement – Part 1-6, ISO, Geneva, 1998-2007.
[6] Standish Group, "CHAOS manifesto 2011", West Yarmouth, Massachusetts, 2011.
[7] Standish Group, "CHAOS summary 2009", West Yarmouth, Massachusetts, 2009, pp. 1-4.
[8] L. Buglione, A. Abran, "The software measurement body of knowledge", Proceedings of 1st Software Measurement European Forum (SMEF), Rome, 2004.
[9] Project Management Institute, A Guide to the project management body of knowledge, PMBOK 2000.
[10] CMMI Product Team, "CMMI for Development, Version 1.3," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, Technical Report CMU/SEI-2010-TR-033, 2010; http://www.sei.cmu.edu/library/abstracts/reports/10tr033.cfm (6.06.2012).
[11] B. Czarnacka-Chrobot, "The ISO/IEC standards for the software processes and products measurement", in New Trends in Software Methodologies, Tools and Techniques, H. Fujita and V. Marik, Eds., Proc. of the 8th International Conference SOMET'2009, Frontiers in Artificial Intelligence and Applications, vol. 199, IOS Press, Amsterdam-Berlin-Tokyo-Washington, 2009, pp. 187-200.
[12] B. Czarnacka-Chrobot, "Standardization of software size measurement", in Internet – Technical Development and Applications, E. Tkacz, A. Kapczynski, Eds., Advances in Intelligent and Soft Computing, vol. 64, Springer-Verlag, Berlin-Heidelberg, 2009, pp. 149-156.
[13] B. Czarnacka-Chrobot, "Standardization of software functional size measurement methods", in Advanced Information Technologies for Management, J. Korczak, H. Dudycz, M. Dyczkowski, Eds., Proceeding of Scientific International Conference AITM'2009, Wroclaw University of Economics Research Papers, no 85, Wroclaw 2009, pp. 40-50.
[14] B. Czarnacka-Chrobot, "The effectiveness of business software systems functional size measurement", Proceedings of the 6th International Multi-Conference on Computing in the Global Information Technology (ICCGI 2011), 19-24 June 2011, Luxemburg City, Luxemburg, Constantin Paleologu, Constandinos Mavromoustakis, Marius Minea, Eds., International Academy, Research, and Industry Association, Wilmington, Delaware, USA, 2011, pp. 63-71.
[15] K. Richins, "Measurement in CMMI", Proceedings of a Seminar On Metrics, International Council on System Engineering (INCOSE), Hampton, Virginia, October 23-24, 2001.
[16] N. E. Fenton, "Ensuring quality and metrics of software" [„Zapewnienie jakości i metryki oprogramowania"], in Software engineering in IT project [Inżynieria oprogramowania w projekcie informatycznym], extended 2nd edition, J. Górski, Ed., Mikom, Warsaw 2000.
[17] B. Gallagher, M. Phillips, K. Richter, and S. Shrum, "CMMI-ACQ: guidelines for improving the acquisition of products and services", 2nd Edition, Addison-Wesley, Boston 2011.
[18] M. Chrissis, M. Konrad, and S. Shrum, "CMMI: guidelines for process integration and product improvement", 3rd Edition, Addison-Wesley, Boston 2011.
[19] E. Forrester, B. Buteau, and S. Shrum, "CMMI for Services: guidelines for superior service, 2nd Edition, Addison-Wesley, Boston 2011.
[20] D. Goldenson, J. Jarzombek, and T. Rout, "Measurement and analysis in Capability Maturity Model Integration Models and Software Process Improvement", CrossTalk, July 2003, pp. 20-24.
[21] D. L. Gibson, D. Goldenson, and L. Kost, "Performance results of CMMI-based process improvement", Software Engineering Institute, Carnegie Mellon University, Pittsburgh, August 2006.
[22] D. F. Rico, "ROI of software process improvement: metrics for project managers and software engineers", J. Ross Publishing, February 2004.
[23] Software Engineering Institute, http://www.sei.cmu.edu/cmmi/why/benefits (6.06.2012).
[24] C. A. Dekkers, B. Emmons, "How function points support the Capability Maturity Model Integration", CrossTalk. The Journal of Defence Software Engineering, February 2002, pp. 21–24.
[25] M. Brown, D. Goldenson, "Measurement and analysis: what can and does go wrong?", 10th IEEE International Symposium on Software Metrics, September 2004.
[26] B. W. Boehm, R. E. Fairley, "Software estimation perspectives", IEEE Software, November/December 2000.