The 5th International Scientific Conference
eLearning and Software for Education
Bucharest, April 09 - 10, 2009
eLSE 2009

AGILE CMMI FOR E-LEARNING SOFTWARE DEVELOPMENT

Grigore ALBEANU *

***Abstract**: The paper describes the impact of Agile approach on e-Learning content developed by organizations implementing CMMI methodology. It is shown that new ICT developments ask for an agile updating of some practices in order to implement low cost projects.*

***Keywords:** Agile, CMMI, e-Learning, software, design.*

I. INTRODUCTION

Quality assurance for integrated software development is an important task nowadays [6, 18, 19]. There is a large variety of methodologies and technologies, and selecting those having the great impact on improving quality of the software process and the software products delivered to customers is a challenging task. Software quality has to be considered not only from software engineering point of view, but also related to quality of design and quality of conformance. The conformance is influenced by quality attributes like: understandability, completeness, conciseness, portability, consistency, maintainability, testability, usability, reliability, structuredness efficiency and security [4, 5].

According to Stracke [27], "Quality development covers every kind of measurement, assurance, optimisation and continuous improvement of the quality within given system". In this paper we are interested in explaining how agile and capability and maturity integration models can be used to develop high level quality components including and not limited to content, simulation units (SU), learning content management systems (LCMS), learning managemen systems (LMS), virtual campus management systems(VCMS).

Recently, according to [10, 11], new quality influential factors were identified: reusability, interoperability, transparency, variability, and extensibility.

The *reusability* can be obtained as least as mentioned in the following scenarios:

- Using a general application resource;
- Deriving the new product from the existing one;
- Using the similarities between different components, operating systems etc.

The software *interoperability* is defined according to ISO/IEC 2382-01 [17], and asks for supporting communication among different components without major intervention by a software engineer.

For e-learning, the *interoperability of assessments* is also important. According to Barchino et al [13], any assessment model will be flexible, will permit formalization and automatic processing, must be reusable (reusability), will be resistant to technical changes and conversion problems (interoperability and sustainability), must cover the whole assessment process (completeness), will be able to describe the semantic meaning of learning objects, will be reproducible, will provide neutrality, and will assure an increased compatibility with available standards and specifications.

The *transparency* is a new factor and is related to the software implementation. It is accepted that “a transparent software development environment means that the enterprise sets its own standards for application quality or security and then monitors compliance with those standards throughout the development lifecycle” according to [23].

The *extensibility* is a new system design principle where the implementation takes into consideration future growth, and can be seen as a systemic measure of the ability to extend a system and the level of effort required to implement the extension. This is consistent with agile methodology asking for customer implication in all phases of an iterative development.

In the following, we provide details about the Agile methodologies for software development (the second section), CMM and CMMI software development models adapted for E-learning components (the third section), and the Agile-CMM and Agile-CMMI approaches (in the fourth section). Finally, concluding remarks and future developments will be outlined.

II. AGILE METHODOLOGIES

During last decades, new software development methods approaches have been proposed including agile software development methodologies with emphasize on: (1) short releases of working versions; (2) an iterative development process; and (3) less documentation and formality [1, 9, 20, 26].

Agile methodologies generally promote a project management process that encourages frequent inspection and adaptation. Agile methods emphasize face-to-face communication over written documents (working in same location) or in different locations but having video contact daily, communicating by videoconferencing, voice, e-mail etc.

Using Agile methodologies is possible to provide a better understanding of what is called “quality profile”, when considering the ISO/IEC 19796-1 standard for learning, education and training.

The following agile methods are suitable when dealing with E-learning component-based software development [1, 9, 10, 21]: Extreme Programming, Scrum, Crystal approach, Feature driven development, the rational unified process, dynamic software development, adaptive software development, open source development, Agile-CMM, Agile-CMMI etc. In the following we details only extreme programming, adaptive software development and open source software development, which are best suited to a dynamic evolution as e-learning experiences nowadays.

The project life cycle in the extreme programming (XP) approach consists of five phases:

- *exploration* – few days to few weeks for content development and from few weeks to few months for software modules used to simulate experiments – customer wishes– project team familiarization;
- *planning* – setting the priority order, effort and first release – after few weeks or months depending on the size of the task;
- *iterations to release* - several iterations, functional tests, the whole system architecture is available and after last iteration the production can start;
- *production* – coding, unit/applet testing, extra functional testing, performance checking;
- *maintenance and death* – customer support tasks, new people and a modified team structure, the lesson/a package of lessons, a software component or the learning management system is full implemented, validated, installed and the documentation is complete.

The team members have different roles and tasks: programmers, customers, testers, trackers, coaches, consultants/educational staff, and the project manager. XP methodology can be used by teams having 3-20 people, which is the case for e-learning content development.

The Adaptive Software Development (ASD) is appropriate for developing complex large systems based on incremental, iterative development, with constant prototyping. Along three phases, called Speculate, Collaborate and Learn, the following steps are required: (1) project initiation, (2) adaptive cycle planning, (3) concurrent component engineering, (4) quality review and (5) final Q/A, and release. There is a learning loop, according to the sequence 2, 3, 4, 2, 3, 4, 2, 3, 4… in order to functional demonstrability assurance. The ASD focuses on developing large systems like a VCMS which include LMS/LCMS components.

The Open Source Software Development (OSSD) can be considered an agile approach based on sharing code freely among colleagues motivated by the need for robust code, faster development cycles, high standards of quality, reliability and stability, open standards and platforms, the need for shared costs and shared risks. The OSSD is the best one in the case of a small budget and time limits in providing access for learners/trainers.

Our experience shows that mixing OSSD with XP or ASD is a good strategy, not only for software development, but also for online course generation. Using OSS components will provide low-cost access to learning objects and difference teaching resources.

III. CMMI MODELS

There are many maturity models for managing systems to be used in different fields, including for software industry. The most recent is CMMI (The Capability Maturity Model Integration). CMMI is an extension of the CMM approach developed by SEI (Software Engineering Institute) at Carnegie Mellon jointly with DoD and industry. It was designed to integrate the proliferation of models like SW-CMM, SE-CMM, IPD-CMM and to be an aid for organizational and project process improvement.

As Glazer et al say in [16], CMMI is not a process standard, but a model having as central theme the process management. CMMI is necessary to be implemented (used as a learning tool, a communication tool, and a means to deal with challenging tasks by the organization), and the best practices to be used for growing the capabilities of the organization. The appraisal method for process improvement is SCAMPI (Standard CMMI Appraisal Method for Process Improvement) and uses a review/journal of the process story (containing the CMMI model used, the development practices of the organization, the improvements identified, and the ability in integration.

When considering online course design, CMMI can use the OCDMM approach described by Neuhauser [22]. The e-learning processes belong to five key process areas along the five maturity levels. The levels are called: Initial (L1), Exploring (L2), Awakening (L3), Strategizing (L4), Integrating best practices (L5). The processes are classified to belong to: Components and appearance (K1), Individualized and personal (K2), Use of Technology (K3), Socialization and Interactivity (K4), and Assessment (K5).

During L1 maturity all information are available in text, including the syllabus and course information, the access is limited and controlled by instructor or course manager, only e-mail communication is used, and the assessment is campus based, not online. According to L2, the organization provides online notes, offers blended courses and uses fonts and colors, the access is controlled by instructor, and learners make use of search engines, library databases and communicate by e-mail, if socialization is provided the discussions are instructor-led (messenger conference), the assessment address the evaluation of papers sent by e-mail. A L3 e-learning based educational entity will integrate lectures with links and discussions, the content being HTML based and using Powerpoint files. The communication is based on e-mail, the web resources will be discovered (people involved being comfortable with ICT), the discussions are instructor controlled, the contact is quite frequent and students participate more and more. The assessment is based on test pools, papers sent to the instructor and the access of students to a LCMS. In order to fit the OCDMM level 4, the contents will be well-structured and based on good/optimal learning objects (high quality units), audio, video, animation and simulation units are available when necessary to describe phenomena, experiments and getting attention. The access is based on a learner-instructor partnership paradigm; students are able to disseminate knowledge from Web resources, to generate discussion, they use collaborative tools for variable size groups. The assessment process for L4 educational units is based on versatility of projects, peer-review, on-line working.

An educational entity fits L5 when is able to develop learning objects, multiple kind of inputs is possible, the presentation is intuitive, the navigation being effortless. Interactive learning aids are available and used, learning objects were designed in such a way to match to student needs and interests, the learning profile is considered in order to provide great individualized learning. All current ICT tools will be used for communication, experiment, learn, to interact and socialize. The assessment is mature consisting in providing multiple assessments for student performance and course improvement, and feedback for effective self-learning is assured.

For software organizations designing and implementing LMS/LCMS, or complex learning objects, the CMMI can extend the eL-CMM, described by Albeanu [3].

IV. AGILE CMMI FOR E-LEARNING

Recently, there are a special interest in implementing CMMI using a combination of agile methods (see Alegria & Bastarrica [7], and their references), or to study the compatibility of CMMI and agile methods, as discussed by Fritzsche & Keil [15]. Experience was reported about projects that use agile methods with certain adjustment in order to achieve CMMI level 2 or 3 [8, 28]. Glazer et al [16] show that even using both agile and CMMI approaches will remain problems to be solved by other models/methodologies. Also, it was shown that it is possible to implement CMMI using a combination of agile methods [2, 8, 12, 13, 14]. This is possible when thinking about CMMI structure and CMMI practice areas.

In the following we describe the compatibility of XP and SCRUM approaches [25] to a large numbers of KPAs of CMMI in a level by level manner.

When speaking about the second level of CMMI (called managed), we identified the following compatibilities:

- ✓ The requirements management is largely supported by on-site customer presence and by continuous integration. The instructor and the students participate for story-cards elaboration.
- ✓ The project planning is also supported if XP is applied correctly, by the so called planning game and small releases (by asking the customer about prioritization). There is a release game and an iteration game.
- ✓ Project monitoring and control is supported by short iterations method and small releases.
- ✓ Supplier agreement management is not addressed by agile methods. The project team will ask for support to the organization.
- ✓ Measurement and analysis is partially supported by XP and largely supported by SCRUM.
- ✓ Process and product quality assurance is supported by XP using pair coaching, and it is not supported by SCRUM.
- ✓ The configuration management is largely supported by XP and not supported by SCRUM. XP uses collective ownership approach in supporting software configuration management, pair programming, tests, and customer collaboration.

In order to fit CMMI level 3 (called defined), using the agile methods, the KPAs should be supported:

- ✓ Requirements development is partially supported by both XP (story cards and functional tests, task cards) and SCRUM.

- ✓ Processes for the technical solutions are supported by XP, but no SCRUM. Prototyping, refactoring, and iterative development are methods used as technical solutions. Coding standards and pair programming are used as implementation solutions [1, 24].
- ✓ The product integration is not supported by SCRUM, but it is largely supported by XP by continuous integration and direct customer involvement.
- ✓ Also, the verification KPA is not supported by SCRUM. However, XP uses intensive testing, a test-first approach, peer reviews (supported by pair programming, refactoring, and the collective code ownership principle).
- ✓ The validation is supported both by XP and SCRUM. In XP, the validation is assured by customer participation and frequent releases.
- ✓ Both XP and SCRUM are not able to support organizational process focus. XP is applied only to projects not to organizations. Also, a strictly discipline imposed in the organization is in conflict with XP.
- ✓ The organizational process definition is not addressed both by XP and SCRUM.
- ✓ However the organizational training is supported. For instance, by XP, training is supported during exploration phase, by pair programming and coaches. There will be no record to certificate the training results.
- ✓ The integrated project management is supported by SCRUM more then by XP. There is an intensive communication and XP integrates and coordinates developers, customer, testers and management.
- ✓ The risk management is largely supported by agile methodologies. XP enforces the identification and analysis the risks during the planning phase. The risk mitigation is assured by using the short iteration approach. The simply human error can appear, but the chance to be detected is big according to the pair programming paradigm, and continuous integration approach.
- ✓ The Integrated teaming KPA is fully supported by agile methods. In XP there are well-defined roles and some approaches supporting the team operation, like: pair programming, collective ownership of the cod, focus on cooperation and communication.
- ✓ The integrated supplier management is not considered nor by XP end SCRUM.
- ✓ Decision analysis and resolution is conflicting with agile methods and is not supported.
- ✓ Finally, the organization environment for integration is supported at the project level but not at the organizational level.

When consider the fourth and five levels of CMMI, any KPAs contain conflicting objectives with Agile approaches. This is a consequence of the project oriented agile methodology. It is clear from above that fitting CMMI level 2 is an easy task for organizations based on agile methodologies. Also the level 3 can be easy attainable. It is practical impossible to attain the level 4 (quantitatively managed) and 5 (optimizing).

However, “a key to achieving more agility with the CMMI is to realize that the practices are primarily advisory or indicative only”, as Anderson (2005) says. A full comparison of CMMI and Agile was realized by Glazer et al [16]. One of the results states that: ”agile methods have flourished in a domain of low cost of failures or linear incremental cost of failure”. Or this is the case of e-Learning software, not only content for learning and training, but also for the software platform which, in some cases, requires updating according to the evolution of ICT, mainly web-based paradigms and open source software.

It is sure that organizations developing content for e-learning will be organized on teams based on projects in an Agile approach.

The instructor and some students will be included in the project team in order to capture the requirements and to provide quality releases. The main reason is the clear orientation of Agile methods on people and allowing people to determine the development technology and processes. Also, the team will include web programmers, multimedia experts, pedagogical experts, etc.

Depending on particular interests, the customer can be represented by one or more members of e-Learning team which is responsible with central e-learning service management, e-Learning material creators, advisers, and trainers. They will talk about their requirements related to Web 2.0 (or E-Learning 2.0) resources and practices in their future e-learning environment.

V. CONCLUSIONS

In this paper we had investigated the impact of Agile methods on CMMI model of capability and maturity. It was shown that fitting CMMI level 3 is possible by organizations working on low-medium projects or component based software developed in a prototype-iterative approach. The benefits of using XP methodology for such projects are detailed. It was claim that XP is a good choice for e-Learning components development, not only interactive content, but also platforms which are able to be changed according to the ICT evolution.

BIBLIOGRAPHY

[1] Abrahamsson P. Salo O., Ronkainen J. & Warsta J., *Agile Software Development Methods. Review and Analysis*, VTT Publications, 478, 2002.

[2] Ahern M.D., Clouse A. &Turner R., *CMMI Distilled: A Practical Introduction to Integrated Process Improvement* (2nd ed.), Addison Wesley, 2003.

[3] Albeanu G., Quality indicators and metrics for capability and maturity in E-learning, *Proceedings of eLSE 2007 – The International Scientific Conference XXI Strategies, Section E-learning and software for education*, Bucharest, April 12-13, 2007, “Carol I” National Defence University Publishing House, Bucharest, 2007, pp. 165-176.

[4] Albeanu G., E-learning metrics, *Proceedings of ICVL 2007: Virtual Learning-Virtual Reality (editors: M. Vlada, G. Albeanu, D.M. Popovici)*, Bucharest Publishing House, 2007.

[5] Albeanu G., Some aspects about quantity and quality in virtual learning and teaching, *OptimumQ*, Vol. XVII(2006), No. 4, pp. 1-6.

[6] Albeanu G. & Popentiu Fl., Total Quality for Software Engineering Management. In Pham Hoang (ed.), *Handbook of Reliability Engineering*, pp. 567-584. 2003.

[7] Alegria J.A.H. & Bastaricca M.C., Implementing CMMI using a Combination of Agile Methods, *CLEI Electronic Journal*, Vol. 9(2006), No. 1, pp. 7-22, http://www.clei.cl/cleiej/papers/v9i1p7.pdf (available, February 2009).

[8] Anderson D.J., Stretching Agile to fit CMMI Level 3, *Agile Conference*, Denver 2005, Microsoft Corporation, 2005.

[9] Andersson D., *Software Development Methods. A Little Orientation in the Jungle of Choices*, IT University of Göteborg, 2006.

[10] Averian A. & Albeanu G., Agile Component-Based Development and Software Quality Assurance, *OptimumQ*, 2008 (in press).

[11] Averian A. & Albeanu G., Quality Assurance for Agile Component-Based Software Development, *The 15th International Conference on Reliability and Quality in Design*, San Francisco, 2009 (submitted).

[12] Awad M.A., *A Comparison between Agile and Traditional Software Development Methodologies*, Report, The University of Western Australia, 2005.

[13] Barchino R., Gutiérrez J.M., Otón S., Martínez J.J., Hilera J.R. & Gutiérrez J.A., E-Learning Model for Assessment, *IADIS Virtual Multiconference on Computer Science and Information Systems, MCCSIS 2006*, http://www.iadis.org/Multi2006/Papers/15/S025_EL.pdf (available, February 2009).

[14] Davis C., Glover M., Mazo J. & Opperthauser D., An Agile Approach to Achieving CMMI, AgileTek Whitepaper, http://www.agiletek.com/ images/AgileTek/pdf/an_agile_approach_to_achieving_cmmi.pdf (available, February 2009).

[15] Fritzche M. & Keil P., Agile Methods and CMMI: Compatibility or Conflict? *e-Informatica Software Engineering Journal*, Vol 1(2007), No. 1, http://www.e-informatyka.pl/e-Informatica/attach/Issue1/Vol1Iss1Art1 eInformatica.pdf (available, February 2009).

[16] Glazer H., Dalton J., Anderson D.J., Konrad M. & Shrum S., *CMMI or Agile: Why Not Embrace Both!* CMU/SEI-2008-TN-003, Carnegie Mellon – Software Engineering Institute, 2008.

[17] ISO, ISO/2382/01: *Data Processing - Vocabulary. Part 01: Fundamental Terms,* Second Edition, 1984.

[18] Kan S.H., Basili V.R. & Shapiro L.N., Software Quality: An Overview from the Perspective of Total Quality Management, *IBM Systems Journal*, Vol 33(1994), No. 1, pp. 4-19.

[19] Kenett R.S. & Baker E.R., *Software Process Quality. Management and Control*, Marcel Dekker, Inc., New York, Basel, 1999.

[20] Miller G., Agile Software Development for the Entire Project, *The Journal of Defense Software Engineering*, Vol. 18(2005), No. 12, pp. 9-12.

[21] Minna Pikkarainen M. & Huomo T., *Agile Assessment Framework*, ITEA, 2005.

[22] Neuhauser C., A Maturity Model: Does it Provide a Path for Online Course Design, *The Journal of Interactive Online Learning*, Vol. 3(2004), No. 1.

[23] Reflective, *Software transparency*, http://www.reflectivecorp.com/ reflective.jsp?navigation=48.

[24] Salo O., *Enabling Software Process Improvement in Agile Software Development Teams and Organisations*, VTT Publications, 618, 2006.

[25] Schwaber K., *Agile Project Management with Scrum*, Microsoft Press, 2004.

[26] Stojanovic Z., Dahanayake A., Sol H., Modeling and Architectural Design in Agile Development Methodologies, In: *Proceedings of the 8th CAISE/IFIP8.1 International*

Workshop on Evaluation Methods in System Analysis and Design EMMSAD '03, Velden, Austria, pp. 180-189.

[27] Stracke C.M., *Quality Standards for Quality Development in e-Learning: Adoption, Implementation and Adaptation of ISO/IEC 19796-1*, QED, 2007.

[28] Xu Y., Lin Z. & Foster W., Agile Methodology in CMM Framework: an Approach to Success for Software Companies in China, *Global Information Technology Management (GITM) Conference*, Calgary, Canada, June 2003, http://zlin.ba.ttu.edu/papers/Outgoing/GITM-ITC-5.pdf, (available, February 2009).

[*] *Ph.D. Prof.,* Spiru Haret *University, Bucharest*