EFFICIENT TEAM BUILDING FOR ON-TIME PROJECTS

Dan Benţa[1]
Ştefan Ioan Niţchi[2]
Darius Şuta[3]
Remus Felix Pop[4]

Abstract

When we think to software development projects we consider that a strong set of requirements is defined and team members always follow initial planning. Well, this was a specific orientation in traditional software developments methods and methodologies. In a current competitive business environment this attitude leads to fail in software development projects. This paper aims to highlight the importance of agile approach focused on meetings and team building and to give examples based on our experience in software development in industrial field for financial and energy sectors. This paper is the result of collaboration between university (business informatics, psychology, and finance) and business environment (internal software development projects).

Keywords: agile software development, team/group building, CMMI, project management

Introduction

Rapid evolution in IT&C leads to performance if modern techniques and methods are customized and adapted by companies. In terms of software development methods, we pointed in our previous works [6], [7] a set of techniques with strengths and weaknesses. Traditional methods as Waterfall Model [30], Prototyping Model [31], Spiral Model [8], Incremental Model [27], and Rapid Application Development (RAD) Model [28] are no longer used in complex software development projects. New trends lead to agile [3] software development methods and we consider that SCRUM [32], [17] methodology best fits customer needs and changing initial requirements for rapid development and project delivery. As described in [24], [29] agility is "*the ability of to both create and respond to change in order to profit in a turbulent business environment*".

Developed to improve chances of business success, best practices defined by Capability Maturity Model (CMM) in Capability Maturity Model Integration (CMMI) [15] process cover topics that include collecting and managing requirements, formal decision making, measuring performance, planning work, handling risks, and more. Company maturity to

[1]Dan Benţa is PhD Candidate at Babeş-Bolyai University of Cluj-Napoca, Faculty of Economics and Business Administration, Romania. E-mail: dan.benta@econ.ubbcluj.ro
[2]Ştefan Ioan Niţchi, PhD, is Professor at Babeş-Bolyai University of Cluj-Napoca, Faculty of Economics and Business Administration, E-mail: stefan.nitchi@econ.ubbcluj.ro
[3]Darius Şuta is student at Babeş-Bolyai University of Cluj-Napoca, Faculty of Psychology and Educational Science, Romania. E-mail: sutadarius@yahoo.com
[4]Remus Felix Pop is PhD Candidate at Babeş-Bolyai University of Cluj-Napoca, Faculty of Economics and Business Administration, E-mail: pop_remus_felix@yahoo.com

develop software projects is essential in CMMI approach as it aims to determine organizational process capability and process maturity.

As well described in [26] an agile project is composed of a series of iterations of development. Iterations are short intervals of time, usually two to four weeks, during which the project makes progress. Adapted for our software development projects, developers implement individual features that have value to customers every iteration. These features are called user stories [26]. In this way, an efficient team building and meetings are mandatory and assure a proper project evolution.

Our main interests are in software development projects, risk management and team building. In next section conceptual background is presented following with our perspective, personal classifications, comments and results from our experience in software development projects, work meetings, group dynamic and products' economical efficiency.

Company maturity to develop software projects

In their research paper [29], authors perform studies on how CMMI could be used in assessing agile software development or in a situation in which the organization is planning to change its processes towards agility and propose an process for assessing agile software development with CMMI as presented in Figure 1.

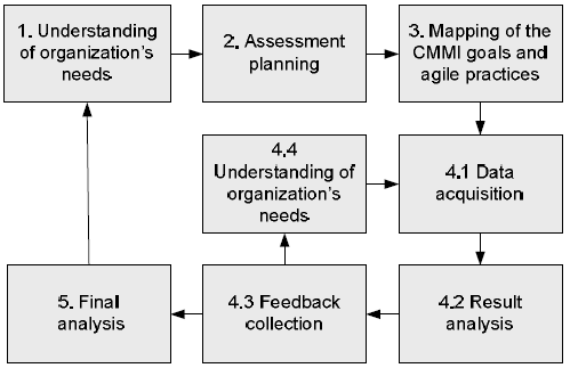

Figure 1 – Assessment Process[5]

Evolution and history of CMMs are presented in Figure 2 and current approaches for CMMI are:

- CMMI for Acquisition, Version 1.3;
- CMMI for Development, Version 1.3;
- CMMI for Services, Version 1.3.

We are interested in CMMI for Development as our objective is to deliver software development projects in predefined costs, time and quality.

[5][Pikkarainen and Mäntyniemi, 2006], Minna Pikkarainen and Annukka Mäntyniemi, An Approach for Using CMMI in Agile Software Development Assessments: Experiences from Three Case Studies, VTT Technical Research Centre of Finland, SPICE 2006 conference.

In the current marketplace, maturity models, standards, methodologies, and guidelines exist that can help an organization improve the way it does business. However, most available improvement approaches focus on a specific part of the business and do not take a systemic approach to the problems that most organizations are facing. By focusing on improving one area of a business, these models have unfortunately perpetuated the stovepipes and barriers that exist in organizations. CMMI® for Development (CMMI-DEV) provides an opportunity to avoid or eliminate these stovepipes and barriers. CMMI for Development consists of best practices that address development activities applied to products and services. It addresses practices that cover the product's lifecycle from conception through delivery and maintenance [13].

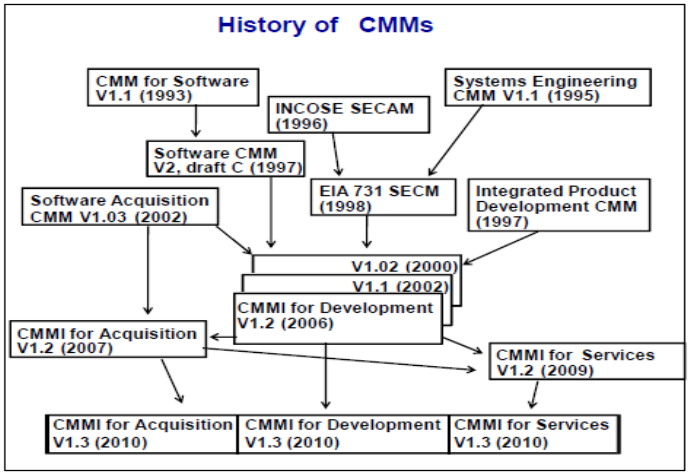

Figure 2 – History of CMMs[6]

Capability Maturity Model Integration for development helps organizations to achieve project objectives by improving practices and measures the business and company maturity in such projects. CMMI® for Development is not actually a method for development as traditional or agile one; it is an approach that helps organizations to improve their techniques in the project life cycle.

As presented in [14], [12], in CMMI are used level to describe an evolutionary path recommended for an organization that wants to improve the processes it uses to develop and maintain its products and services. CMMI supports two improvement paths (Figure 3):

- First path enables organizations to incrementally improve processes corresponding to an individual process area (or process areas) selected by the organization, so it is a continuous representation and is associated with capability level;
- Second path enables organizations to improve a set of related processes by incrementally addressing successive sets of process areas, so it is a staged representation and is associated with maturity level.

[6][EIA, 2002], Electronic Industries Alliance. Systems Engineering Capability Model (EIA/IS-731.1). Washington, DC, 2002.

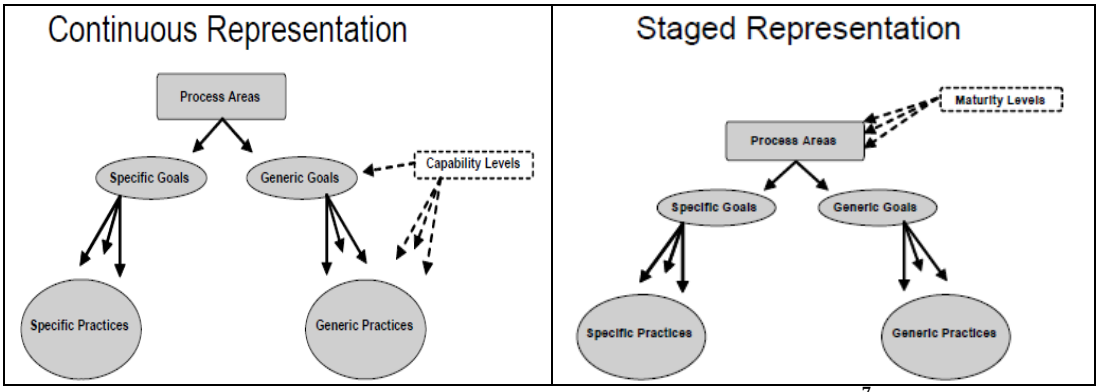

Figure 3 – Structure of the Continuous and Staged Representations[7]

The six capability levels, designated by the numbers 0 through 5, are [14]:
0. Incomplete - is a process that either is not performed or partially performed
1. Performed - is a process that satisfies the specific goals of the process area. It supports and enables the work needed to produce work products.
2. Managed - is a performed (capability level 1) process that has the basic infrastructure in place to support the process
3. Defined - is a managed (capability level 2) process that is tailored from the organization's set of standard processes according to the organization's tailoring guidelines, and contributes work products, measures, and other process improvement information to the organizational process assets
4. Quantitatively Managed - is a defined (capability level 3) process that is controlled using statistical and other quantitative techniques
5. Optimizing - is a quantitatively managed (capability level 4) process that is improved based on an understanding of the common causes of variation inherent in the process.

There are five maturity levels, each a layer in the foundation for ongoing process improvement, designated by the numbers 1 through 5 [14]:
1. Initial - At maturity level 1, processes are usually ad hoc and chaotic
2. Managed - At maturity level 2, the projects of the organization have ensured that processes are planned and executed in accordance with policy; the projects employ skilled people who have adequate resources to produce controlled outputs; involve relevant stakeholders; are monitored, controlled, and reviewed; and are evaluated for adherence to their process descriptions.
3. Defined - At maturity level 3, processes are well characterized and understood, and are described in standards, procedures, tools, and methods. The organization's set of standard processes, which is the basis for maturity level 3, is established and improved over time.
4. Quantitatively Managed - At maturity level 4, the organization and projects establish quantitative objectives for quality and process performance and use them as criteria in managing processes.

[7][CMMI2, 2006], CMMI Product Team, Improving processes for better products, CMMI® for Development, Version 1.2, CMU/SEI-2006-TR-008, ESC-TR-2006-008, Carnegie Mellon University, 2006

5. Optimizing - At maturity level 5, an organization continually improves its processes based on a quantitative understanding of the common causes of variation inherent in processes.

The capability levels of a process area are achieved through the application of generic practices or suitable alternatives to the processes associated with that process area [14].

Organizations can achieve progressive improvements in their organizational maturity by achieving control first at the project level and continuing to the most advanced level—organization-wide continuous process improvement—using both quantitative and qualitative data to make decisions [14].

Projects are unique and have their own trajectory. We consider that in a software development project life cycle is very important cu accept new requirements. Capability to adapt is more efficient than considering that initial requirement plan cannot be changed.

From our experience in financial field we consider that adapting to new requirements; additional initial project is a cheaper solution that involves risks and potential losses in a less manner than giving up the initial project and funding a new project. We take into consideration one of the main marketing ideas in financial institutions, according to which the financial institution must successfully agree their clients' financial needs through measures designed to identify new needs, to be able to reshape financial products and services. The financial institutions must have a functional and flexible organizational structure that allows continuous adaptation to customers' financial needs. Also in the process of adapting to new projects' requirements, they have to consider that the competition in their sector is in constant growth.

Group/Team building in achieving projects' goal

We consider that team building is a decisive factor in order to succeed in software development projects. An agile approach is mandatory to meet requirements that change over the project life cycle. Besides team training, group development is essential to produce innovative ideas and to deliver projects in predefined costs, time and quality.

The stages of Group development are best described in Bruce W. Tuckman's article "*Developmental Sequence in Small Groups*" (1965), in which he synthesized about 50 articles on different type of group formation**.** The stages identified in these articles are separated into those descriptive of social or interpersonal group activities and those descriptive of group-task activities. The author proposes 4 stages for the group development in both the social realm and the task realm, in the social realm. These stages are testing-dependence, conflict, cohesion, and functional roles. In the task realm, they are orientation, emotionality, relevant opinion exchange, and the emergence of solutions. For the proposed stages the author coined the well known terms, *forming, storming, norming* and *performing*. The most relevant type of group for us is the human relations training-group (T-group), in which the aim is to create an atmosphere where individuals can interact in a less defensive and more productive manner and to be aware of the dynamics

underlying such interaction. The goal in this type of group is the development of interpersonal sensitivity. This type of group is usually formed out of students or corporation executives, and a trainer or leader, lasting for about three to six months [34].

The proposed distinction between the group as a social entity and the group as a task entity is similar to the distinction between the task-oriented functions of groups and the social-emotional-integrative functions of groups, both of which occur as simultaneous aspects of group functioning [2], [16], [18], [25] in [34]. *In T-groups, the task is a personal and interpersonal one in that the group exists to help the individuals deal with themselves and others* [34].

The first stage in the realm of group structure in the model described by Tuckman is labeled as *testing* and *dependence*. The first term, "testing", refers to an attempt by group members to see what kind of behavior is accepted in the group, by its members based on their reaction and the reaction of the trainer (if one is present). The first stage of group development is described by several authors, [23], [5], [10], [11], and [9], as being one best characterized by dependence which refers to a *strong expression of dependency needs by the members toward the trainer, and attempts at group structuring to work out authority problems by the quick acceptance of and dependence on such arbitrary norms. The first stage of task-activity development is labeled as task orientation, in which group members attempt to identify the task in terms of its relevant parameters and the manner in which the group experience will be used to accomplish the task. The group must decide upon the type of information they will need in dealing with the task and how this information is to be obtained. In orienting on the task, one is essentially defining it by discovering its basic rules. Thus, orientation, in general, characterizes behavior in both interpersonal and task realms during this stage* [34].

Regarding this first stage, we consider that it aims to identify each member qualities and how those qualities may be used in accomplishing the projects' overall objective. Achieving the objective involves tasks and resource allocation to each team member based on their previous experiences. If it is possible, lesson learned reports should be considered in order to avoid repeating a same mistake twice. Each member' previous experience can influence the project evolution and implementation. This stage is characterized by little work and a variable amount of emotionality, during which the members are concerned with defining the directions the group will pursue. In this stage interpersonal problems are taken care of with dependence, while task problems are met with task-oriented behavior. Orientation, testing and dependence constitute the group process defined by Tuckman as *forming* [34].

The second phase in group development is known as intergroup conflict. In this phase, group members become hostile toward one another and toward the trainer as a means of expressing their individuality and resisting the formation of group structure. Interaction between the members is uneven and fighting within the group is common. The lack of unity is an easy noticeable feature of this phase. *There are characteristic key issues that polarize the group and boil down to the conflict over progression into the 'unknown' of interpersonal relations or regression to the security of earlier dependence*. Characteristic to the second stage of task activity development is emotional response to task demands.

Here, *group members react emotionally to the task as a form of resistance to the demands of the task on the individual, that is, the discrepancy between the individual's personal orientation and that demanded by the task. This task stage will be most evident when the task has as its goal self-understanding and self-change, namely, the training-group tasks, and will be considerably less visible in groups working on impersonal, intellectual tasks. In both task and interpersonal realms, emotionality in response to a discrepancy characterizes this stage. However, the source of the discrepancy is different in the different realms* [34].

The conflict and polarization behaviors serve as a resistance to group influence and task requirements and it is labeled by the author as *storming*. Our opinion is that in this stage each member expresses his own opinion and ideas regarding the project, roles and task allocation. Please note that this stage is dominated by complaints and conflicts in order to avoid future communication problems in the team. The communication gaps between the members will only be filled in the next stage where communication will become more efficient.

The third group structure phase is labeled as the development of group cohesion. After a stage characterized by conflict and polarization is time for a stage that's characterized by the reduction of conflict, resolution of the polarized issues and establishment of group harmony, in this phase group norms and values emerge. *Group members accept the group and accept the idiosyncrasies of fellow members. The group becomes an entity by virtue of its acceptance by the members.* Harmony is of maximum importance, and task conflicts are avoided to insure it. The third stage of task activity development was labeled as the *open exchange of relevant interpretations.* In the training-group context, this takes the form of discussing oneself and other group members, since self and other personal characteristics are the basic task inputs [34].

This stage has been labeled as *norming*. From our approach, this stage generates final roles and norms that helps flourish, tensions has been eliminated within the group and the members can now focus more efficiently on the task at hand. The communication boundaries have been passed and each member can now express freely own ideas with no fear of being judged in case of a mistake.

The fourth developmental phase of group structure is known as functional role-relatedness. *The group, which was established as an entity during the preceding phase, can now become a problem solving instrument. It does this by directing itself to members as objects, since the subjective relationship between members has already been established. Members can now adopt and play roles that will enhance the task activities of the group, since they have learned to relate to one another as social entities in the preceding stage. Role structure is not an issue but an instrument which can now be directed at the task. There is some tendency for T-groupers to emphasize the task aspects of the final stage, namely, the emergence of insight into the interpersonal process. In doing this, it is made implicit that the group as a social entity characterized by task-oriented role-relatedness makes the emergence of such insight possible by providing support and an opportunity for experimentation and discovery.* The group is described by [10] in [34] *as becoming a work organization which provides member support, mutual*

acceptance, and has strong but flexible norms. In task-activity development, the fourth and final stage is identified as the emergence of solutions. It is here that we observe constructive attempts at successful task completion. In training-group context, these solutions are more specifically insight into personal arid interpersonal processes and constructive self-change. Here, as in the three preceding stages, there is an essential correspondence between group structural and task realms over tune. In both realms, the emphasis is on constructive action, and the realms come together so that energy previously invested in the structural realm can be devoted to the task, this stage was named by the author as performing [34].

In our case, performing is by far the most efficient stage, concerning the task, here results are visible, and the members are task oriented with few interpersonal conflicts. Each member is allocated to the task where it has maximum efficiency and in case he finishes his task before the deadline he may assist his team mates on their parallel tasks or on non depending tasks from their Project Management Plan (Figure 4). All team members are focused to deliver the project in predefined costs, time and quality.

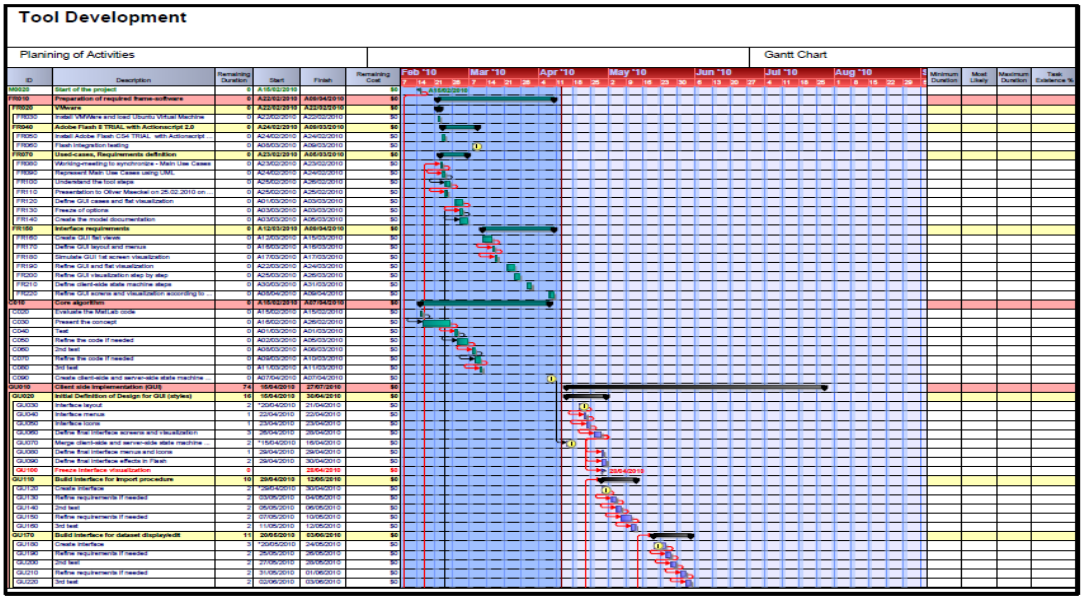

Figure 4 – Section in our Project Management Plan

Another important research on group development was conducted by [4]. Belbin proposed that five criteria must be fulfilled for the construction of an effective team; that each member contributes to achieving objectives by performing a functional role and a team role; that an optimal balance in other functional and team roles is needed, depending on the team's goals and tasks; that team effectiveness depends on the extent to which members correctly recognize and adjust to the relative strengths; that personality and mental abilities fit members for some team roles and limit their ability to play others, and; that a team can deploy its technical resources to best advantage only when it has the range and balance of team roles to ensure efficient team work [22]. In his research, Belbin focuses mostly on the roles team members play within the group. He describes eight

different types of roles, each one of them considered important for the group performance. He coined different names for the team roles, and described them in detail. In the mentioned article, the author distinguishes the following roles:

- Completer-finisher: He is conscientious, anxious and perfectionist. He also searches out errors and omissions, do to him the task is delivered on time. Without him the team would never finish the task on time;
- Co-ordinator (Chairman): He is mature, confident, he clarifies goals, promotes decision making, delegates well, inclined to be lazy and he takes credit for effort of a team;
- Implementer (Company worker): He is disciplined, reliable, conservative, and efficient and turns ideas into practical action. He is the one that plans the actions and manages the team but he is not the leader;
- Monitor-evaluator: He has an analytic intelligence, seeing all options, discerning, strategic, logical, sober, skeptical and cynical. His contribution is mainly the selection of ideas given within the group (similar to quality control);
- Plant: He is creative, imaginative, and unorthodox, he solves difficult problems, he is preoccupied with ideas and neglects practical matters; he has a strong ownership of ideas. He is inattentive to details and intolerant to criticism;
- Resource investigator: He is an extrovert, enthusiastic, communicative, explores opportunities, a diplomat and he develops contacts. He's main contribution is offering new development paths for the group. He is neither original nor a leader;
- Shaper: He is mobile, perseverant and dominant. In the absence of the chairman he is the one leading the team. He thrives under pressure, has the drive and courage to overcome obstacles;
- Team worker: He is a diplomat; he listens, builds, and averts friction. He is pleasant and uncompetitive avoiding situations that may entail pressure. His main contribution is team support.

For a team to function properly it must have all of the eight rolls, Belbin acknowledged that an individual may be able to operate effectively in more than one team role, thereby releasing an optimal team from any constraints on numbers. A conflict inside a team is defined as a confrontation of interests or incompatible activities that exist between the participants involved in social situations [19], [1]. [33] in [1] *emphasized three basic themes underlying common definitions of* conflict. *First, a conflict exists only if it is perceived as conflict by the actors involved. Second, there is a level of interdependence between the actors such that they have the ability to influence each other. Finally, in any conflict, scarcity of resources (such as money, power, and prestige) may generate tensions among the actors.* In terms of a conflict, we agree that a task oriented conflict produce benefits that helps a project to grow and generates new and innovative ideas and approaches. Opposite, a member oriented conflict is not project productive and may obstruct the completion of the performed task or may generate negative impact on project overall objective.

Conclusions

We consider that collaborating in the sprint meetings is essential in a software development project using agile methods. A team building period and team accommodation with the project and with the product they need to develop is a first important step. Those meetings are design to fill the collaboration gaps between team members and to clear point what each member done, what each member has to do until next meeting. In this way, potential risks are identified and things that can go wrong in order to respect project` initial plan and schedule.

Agile software development methods can easily be integrated with project management for a proper project plan, task allocation and resource planning. Initial requirements change during the project life cycle and the implementation team should consider all the necessary changes. In our internship, we were involved in a software development project for risk management. Agile software development approach proved to be best choice and lead to a successful project. We managed to develop a software solution and to integrate it in company policy. Company culture and policy represented an essential aspect in our case. Initial requirements changed during the project life cycle and in many cases implementing an additional requirement plan proved to be more benefic than ignoring it.

As previously described, concerns in terms of team building and group development are not new but techniques must continually adapt to new market requirements in order to satisfy all customer needs. Our opinion is that group development is a mandatory feature in the project life cycle and evolution. An individual can obtain benefits from the group as the group can obtain benefits from each individual, previous experience of each team member combined in teamwork may produce better results with less effort then when performed individually, and this is particularly useful in complex projects. Interrelations within the group help the members to develop new skills and abilities or to improve the existing ones.

Besides company culture oriented on agile methods, human factor and group dynamic is very important, reason why team building must guarantee that no communication gaps or any constraints exist. Human behavior is an interesting approach and some authors [21] emulate the behavior of the humans in solving their problems through group counseling. *This is motivated by the fact that the human's thinking is, or should be, the most reasonable and influential, and group counseling is in essence a problem-solving technique.* The challenge of applying agile isn't in merely adopting the practices. The practices are simple. The real challenge arises in the collision between the company culture and policy, project team members, and agile. *Agile methodologies such as Scrum create transparency. Every deficiency that obstructs the best flow of work is singled out for examination* [26].

Acknowledgment

This paper was supported by Romanian National Authority for Scientific Research under the grant no. **PN2 92-100/2008 SICOMAP**.

Bibliography

1. Aritzeta, A. Ayestaran, S. Swailes, S. (2005). Team Role Preferences And Conflict Management Styles. The International Jounal of Conflict Management, Vol. 16, No. 2, 2005 pp. 157-182
2. Bales, R. F. (1953), The equilibrium problem in small groups, in T. Parsons, R. F. Bales and E. A. Shils (eds.), Working Papers in the Theory of Action, Free Press, 111-61.
3. K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R.C. Martin, S. Mellor, K. Schwaber, J. Sutherland & D. Thomas, Principles behind the Agile Manifesto, Available on-line: http://agilemanifesto.org/iso/en/ principles.html, [Accessed 05 Nov., 2010].
4. Belbin, R.M. (1981). Management teams: Why they succeed or fail. London: Heinemann.
5. Bennis, W. G., and Shepard, H. A. (1956), A theory of group development, Human Relations, 9, 415-37.
6. Dan Benţa, Ştefan Ioan Niţchi, SUCCEEDING IN SOFTWARE DEVELOPMENT PROJECTS, The Proceedings of Journal ISOM Vol. 4 No. 2 / December 2010, Journal of Information Systems & Operations Management, ISSN: 1843-4711, pp.9-19, 2010
7. Dan Benţa, 1st PhD. Report, Current state of research from PhD. thesis Software development and risk management in complex projects, Babeş-Bolyai University of Cluj-Napoca, Faculty of Economics and Business Administration, Business Information Systems Department, December, 2010.
8. B. Boehm, "A Spiral Model of Software Development and Enhancement" in ACM SIGSOFT Software Engineering Notes, ACM, 11(4), pp.14-24, August 1986.
9. Bradford, L. P., and Mallinson, T. (1958), Group formation and development, in Dynamics of Group Life, National Education Association, National Training Laboratories, Washington.
10. Bradford, L. P. (1964a), Trainer-intervention: case episodes, in L. P. Bradford, J. R. Gibb and K. D. Benne (eds.), T-Group Theory and Laboratory Method, Wiley, 136-67.
11. Bradford, L. P. (1964b), Membership and the learning processes, in L. P. Bradford, J. R. Gibb and K. D. Benne (eds.), T-Group Theory and Laboratory Method, Wiley, 190-215.
12. Mary Beth Chrissis, Mike Konrad, Sandra Shrum, CMMI for Development®: Guidelines for Process Integration and Product Improvement, 3rd Edition, Published Mar 10, 2011 by Addison-Wesley Professional. Part of the SEI Series in Software Engineering series.
13. CMMI Product Team, CMMI® for Development, Version 1.3, Improving processes for developing better products and services, TECHNICAL REPORT, November 2010
14. CMMI Product Team, Improving processes for better products, CMMI® for Development, Version 1.2, CMU/SEI-2006-TR-008, ESC-TR-2006-008, Carnegie Mellon University, 2006

15. CMMI | Overview, Available on-line: http://www.sei.cmu.edu/cmmi/ [Accessed on 02 of April, 2001]
16. Coffey, H. S. (1952), Socio and psyche group process: integrative concepts, Journal of Social Issues, 8, 65-74.
17. Mike Cohn, Succeeding with Agile Software Development using SCRUM, The Addison-Wesley Signature Series, 2010
18. Deutsch, M. A. (1949), A theory of cooperation and competition, Human Relations, 2, 129-52.
19. Deutsch, M. (1973). The resolution of conflict: Constructive and destructive processes. New Haven: Yale University Press.
20. Electronic Industries Alliance. Systems Engineering Capability Model (EIA/IS-731.1). Washington, DC, 2002.
21. M.A.Eita and M.M.Fahmy, Group Counseling Optimization: A Novel Approach, M. Bramer et al. (eds.), Research and Development in Intelligent Systems XXVI, DOI 10.1007/978-1-84882-983-1_14, © Springer-Verlag London Limited 2010
22. Fisher, S.G., Hunter, T.A., & Macrosson, W.D.K. (2001). A validation study of Belbin's team roles. European Journal Of Work and Organizational Psycology, 2001, 10 (2),pp 121–144
23. Herbert, E. L., and Trist, E. L. (1953), The institution of an absent leader by a student's discussion group, Human Relations, 6, 215-48.
24. J. Highsmith, Agile Project Management, Creating innovative products, AddisonWesley, 2004.
25. Jennings, H. H. (1947), Sociometric differentiation of the psychegroup and sociogroup, Sociometry, 10, 71-9.
26. Clinton Keith, Agile development in Agile game development with Scrum, Addison-Wesley, 2010, ISBN 0-321-61852-1, pp 13-32
27. C. Larman and V.R. Basili (June 2003). "Iterative and Incremental Development: A Brief History" in IEEE Computer (IEEE Computer Society) 36 (6), pp. 47–56, June 2003.
28. J. Martin, Rapid Application Development, New York: MacMillan, 1991.
29. Minna Pikkarainen and Annukka Mäntyniemi, An Approach for Using CMMI in Agile Software Development Assessments: Experiences from Three Case Studies, VTT Technical Research Centre of Finland, SPICE 2006 conference.
30. W. Royce, "Managing the Development of Large Software Systems" in Proceedings of IEEE WESCON, The Institute of Electrical and Electronics Engineers, pp.1-9, August 1970.
31. M.F. Smith, Software Prototyping: Adoption, Practice and Management, London: McGraw-Hill, 1991.
32. H. Takeuchi and I. Nonaka, The New Product Development Game. Harvard Business, pp.137–146, Review Jan/Feb 1986.
33. Thomas, K. W. (1992). Conflict and negotiation processes in organizations. In D. Dunnette& L. M. Hough (Eds.), Handbook of industrial & organizational psychology (2nd ed.. Vol. 3, pp. 651-717). Palo Alto, CA: Consulting Psychology Press.
34. Tuckman, B. W., (1965). Developmental Sequence in Small Groups. Psychological Bulletin, Volume 63, Number 6, pp 284-99