

ROLES AND TEAMS IN A METHOD ENGINEERING FRAMEWORK

Dagher, L.M, Department of Software Engineering, University of Technology, Sydney,
ldagher@it.uts.edu.au

Henderson-Sellers, B, Department of Software Engineering, University of Technology,
Sydney, brian@it.uts.edu.au

Serour, M.K, Department of Software Engineering, University of Technology, Sydney,
mserour@it.uts.edu.au

Abstract

Using a situational method engineering(SME) approach, we analyze the degree of support for people, the roles they play and how they work together in teams in one specific SME repository (that of OPEN). Since the successful use in practice of an SME-constructed methodology depends upon the existence of appropriate method chunks in the OPEN repository, we provide new method chunk descriptions to provide additional support for the existing OPEN repository in the area of people and teams.

Keywords: Methodology, Situational Method Engineering (SME), Software Development, Team, Team Building

1 INTRODUCTION

Formal methodologies for software development have traditionally been rigid and hard to modify. They have typically supported the software development activities at a technical level but been less supportive at the people level. While a methodology such as DSDM (Stapleton, 1997) does include significant support for the personnel element, it does not provide the technical detail for, say, object-oriented development that a rigid and formal methodology such as RUP (Kruchten, 1999) does. Other people focussed methodologies such as those currently labeled as “agile” also do not offer much of the more extensive managerial and technical support that is needed for large projects. In this paper, we commence with a flexible, yet formal, methodological approach based on the ideas of situational method engineering or SME (Kumar and Welke, 1992). SME is used as a means to engineer a method for a specific situation (Brinkkemper, 1996; Ter Hofstede and Verhoef, 1997; Henderson-Sellers, 2003) by selecting appropriate method fragments from a methodbase or repository and assembling them to create the highly customized methodology. This contrasts with prepackaged methodologies, which are severely limited in their flexibility and adaptability to non-standard situations (Constantine and Lockwood, 1994). Using SME, the successful use of the methodology then depends upon (i) the construction of a project-specific methodology from the method chunks and (ii) the availability within the repository of an adequate spread of method chunks. In particular, we ask whether one specific method chunk repository (that of OPEN) contains sufficient support for teams, both in terms of groups and individuals. To correct the identified weaknesses, we make suggestions for additions to the OPEN methodological repository of method chunks in this highly important area of people, the roles they play and how these are managed through high quality teamwork (Martin, 2001). These additions will enable organizations to adopt new/or enhanced method chunks specifically engineered to support teamwork.

2 OPEN

Object-oriented Process, Environment, and Notation (OPEN) is the longest established third-generation, public domain, full lifecycle, process-focussed, methodological approach that was designed for the development of software intensive applications, particularly object-oriented and component-based development (Graham *et al.*, 1997; Henderson-Sellers *et al.*, 1998). OPEN is a flexible object-oriented software development framework; it is not a process *per se*.

From this process metamodel, an organizationally-specific process (instance) can be created using the concepts of Situational Method Engineering. This process instance is created by choosing, from the OPEN repository of process components (method chunks), specific method chunks and then configuring these for the specific organization. This approach permits a flexible and fully configurable process to be generated from the framework, a process that is specifically attuned to individual organizational needs. Project-specific processes are then the realization or enactment of this documented, organizationally-specific process.

Unlike other OO software processes, OPEN is defined by a metamodel equivalent to the M2 level in the OMG standard (OMG, 2001). The OPEN metamodel defines the following five main high level classes of method chunks (Figure 1):

- **Work Product:** ‘A Work Product is anything of value that is produced during the development process’ (Firesmith and Henderson-Sellers, 2002). Work Products are the result of producers (people) executing Work Units and are used either as input to other Work Units or delivered to a client. Pragmatically, they also include externally-supplied (e.g. by user) pre-existing artifacts used as inputs to Work Units.
- **Producer:** ‘A Producer is responsible for creating, evaluating, iterating and maintaining Work Products’ (Firesmith and Henderson-Sellers, 2002).
- **Work Unit:** A Work Unit is defined as a functionally cohesive operation that is performed by a Producer. There are three major classes of Work Unit: Activity, Task and Technique.
- **Language:** A Language is defined as a medium for documenting a Work Product.
- **Stage:** A Stage is defined as an identified and managed duration within the process or a point in time at which some achievement is recognized.

Each of these metaclasses has many subclasses from which one or more method chunk instances can be generated, which are, in turn, documented and stored in the OPEN repository. Thus, the OPEN repository contains each instantiated element, known as a process component or method chunk. Figure 1 illustrates the metaclasses of OPEN’s metamodel.

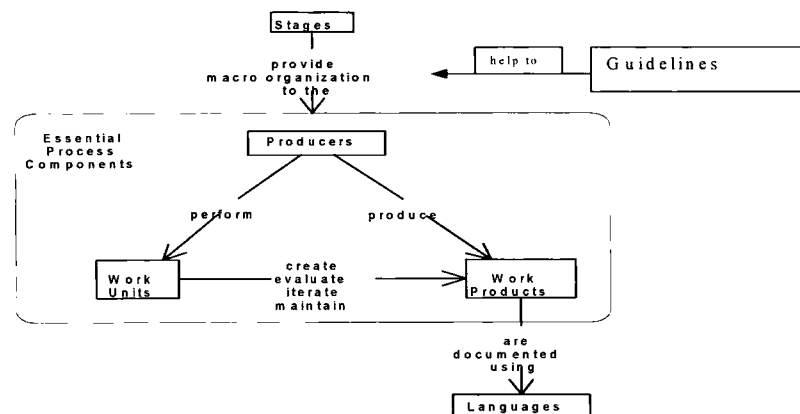


Figure 1 The five major metaclasses of OPEN’s metamodel (after Firesmith and Henderson-Sellers, 2002) © Addison-Wesley.

In this paper, we focus on the Producer metaclass. Producers include people who are assigned roles individually and then often grouped into teams in order to complete the work units to produce work products. A producer is said to be anything that produces (i.e. creates, evaluates, iterates or maintains) versions of one or more work product' (Firesmith and Henderson-Sellers, 2002). OPEN has two types of producers: Direct and Indirect. Direct producers are those who are responsible for the creation of one or more product. Direct producers are people, their assigned roles and software tools. Indirect producers are on a higher level: Teams and Organisations. An Organization (in OPEN) is then defined as consisting of one or more Teams. In addition, we focus on teams and the roles played by individuals in the teams. We seek weaknesses and omissions in the repository contents and thus enhance OPEN by augmenting it with missing method chunks.

3 TEAM BUILDING AND TEAM MATURATION

3.1 What is a Team?

The term 'team' or 'working team' is a concept pertaining to the task of getting a number of individuals with diverse skills working together towards a common purpose. According to the Macquarie Dictionary, a team is defined as 'a number of persons associated in some joint action'. Chowdhury *et al.* (2002) define a team as a collection of individuals thriving to accomplish a common goal for a set of tasks, while Katzenbach and Smith (1993) focus on the team's dedication to a common purpose. In summary, we can simply define team as a number of individuals working together towards a common purpose to achieve a common goal for a given project.

3.2 Team Roles

Belbin (1993, 2000) describes nine types of roles for individual team members. A unique feature of each individual role in this approach is a description not only of the strengths and contributions of each role but also its allowable weakness (Table 1).

BELBIN Team-Role Types	Contributions	Allowable Weaknesses
PLANT	Creative, imaginative, unorthodox. Solves difficult problems.	Ignores incidentals. Too pre-occupied to communicate effectively.
COORDINATOR	Clarifies goals, promotes decision-making, delegates well.	Can often be seen as manipulative. Off loads personal work.
MONITOR EVALUATOR	Sober, strategic and discerning. Sees all options. Judges accurately.	Lacks drive and ability to inspire others.
IMPLEMENTER	Disciplined, reliable, conservative and efficient. Turns ideas into practical actions.	Somewhat inflexible. Slow to respond to new possibilities.
COMPLETER FINISHER	Painstaking, conscientious, anxious. Searches out errors and omissions. Delivers on time.	Inclined to worry unduly. Reluctant to delegate.
RESOURCE INVESTIGATOR	Extrovert, enthusiastic, communicative. Explores opportunities.	Over - optimistic. Loses interest once initial enthusiasm has passed.
SHAPER	Challenging, dynamic, thrives on pressure. The drive and courage to overcome obstacles.	Prone to provocation. Offends people's feelings.
TEAMWORKER	Co-operative, mild, perceptive and diplomatic.	Indecisive in crunch situations.
SPECIALIST	Single-minded, self-starting, dedicated. Provides knowledge and skills in rare supply.	Contributes only on a narrow front.

Table 1. The Belbin Role Types.

3.3 Team Building

Team Building is a concept that commences when an organisation finds the need to have teams to accomplish the organisational objectives. Tuckman (1965) claims, that the team building activity is based on a set of consecutive stages as part of a team development lifecycle. Ciaburri (1998) asserts that a team requires information skills and thinking power, which is developed through team building activities. Indeed, the term ‘Build’, according to the Oxford English Dictionary (2001), is ‘to use as a base from which you can make further progress’. An effective team’s foundation is thus centred on the activity of team building (Adair, 1986).

Adair (1986) identifies three major areas that form the foundation of a team: selecting team members, team review and team consultancy. The skills of team leaders have been suggested (Nador, 2001) as being:

1. Work reflects organisational success
2. Choose the right people
3. Identify Team Roles
4. Maintain Project Plan
5. Develop Shared Understanding

The purpose of team building is to provide learning and feedback for improvements to be implemented for the next project. Team learning is a process that extends throughout the life of a working team. Team members are encouraged to look back and reflect upon their achievement through the team experience, what lessons have been learned and which of these can be applied to the next project (Kerth, 2001). Feedback is one way of building a good team, an effective team ‘takes high energy and deliberate focus on fundamental principles’ (Rohlander, 1999). Rohlander (1999) discriminates between good and bad feedback. Teams need to be experienced in order for learning to take place from both types of feedback. If feedback is part of working teams, it will identify areas of improvement.

3.4 Team Maturation

It has been shown that team building is a continuous cycle (Adair, 1986) throughout the life of a working team. Tuckman (1965) argues that team building consists of five stages, which are Forming, Storming, Norming and Performing and Adjourning (Table 2), which together form the Team Development Lifecycle (Figure 2).

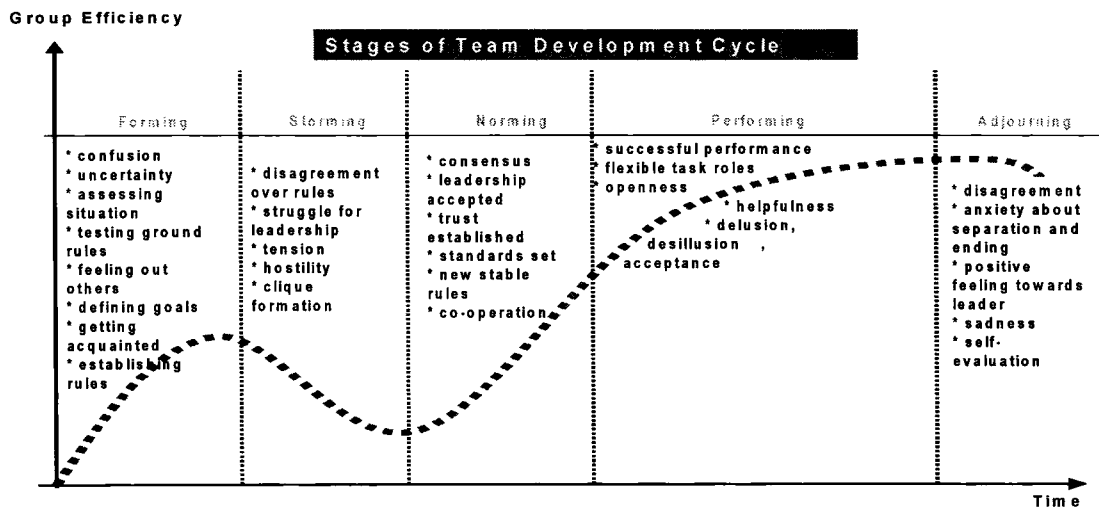


Figure 2. Team Development Lifecycle.

Forming consists of defining tasks, understanding team problems, establishing team vision and barriers to team formation. Tuckman (1965) compares the forming stage to the period of direction and reliance articulated by young children. Individuals transform into members of a team, thus shifting their mindset from individual to a team member. A study conducted by Sheard and Kakabadse (2002) illustrated team forming as ‘clearly stating the goals, clarifying priorities and establishing communication amongst team members’. The results of the study indicate that out of the nine factors, group dynamics was the main key factor of this stage.

Storming is the most difficult stage of the team building model. This stage involves a full understanding of the tasks that each member needs to accomplish. The tasks are identified and assigned to team members. At the storming stage, team members further understand their surrounding environment, leading to conflict and resistance. The results of the study by Sheard and Kakabadse (2002) found that self-awareness and leadership were the two main factors experienced during the storming stage. Tuckman (1965) asserts that, ‘teams that do not experience the storming stage tend to be divided and less creative’.

Norming is the stage at which team members can manage and resolve conflict. Team members develop relationships and avoid conflict, reducing the amount of information shared. This is problematic, especially for software development teams reluctant to share their code with other members. The skills and talents of each team member are not distributed to support team thinking. Indeed, Sheard and Kakabadse (2002) found the key factor to be clear communication throughout this stage.

Performing is the stage of developing relationships and the working team is an effective unit. The working team is progressing through implementation of problem solving skills and implementing changes. An effective team at this stage should show signs of collaborative work, competent team members, good leadership and unified commitment. The study by Sheard and Kakabadse (2002) reportedly found that all factors work collaboratively accomplishing tasks.

Adjourning is the stage during which evaluation on teamwork and team members is undertaken, with the team reflecting upon their achievements. As soon as the project is completed, the team dissolves to form another team or start a new project.

Table 2. The Stages of Tuckman's Team Maturation Model

4 CURRENT OPEN SUPPORT FOR TEAMS

The examination of OPEN shows some existing support for working teams through the high level process elements (method chunks) that consist of Work Units, Producers and Work Products (Figure 1). Although OPEN states types of teams, team roles and team structure of a software development, it does not specify any team building mechanisms. Team building is ‘the key to successful team building is to identify complementary skills in its members and also be aware of their weaknesses’ (Firesmith and Henderson-Sellers, 2002, p.92). This forms the basis of this paper to contribute to the OPEN repository, by adding method chunks to support working teams for situational method engineering.

Existing OPEN Tasks are (1) Choose project team; (2) Identify project roles and responsibilities; and (3) Develop education and training plan together with two major Techniques (not discussed here) of Team Building and Role assignment.

Roles are both important to team building and maturation *per se* and are also well supported already in OPEN. As noted earlier, OPEN supports a wide range of both direct producers (individuals) and indirect producers (teams). Examples of such supported kinds of teams are Business Engineering Team; Deployment Team; Environment Team; Process Team; Software Development Team; Test Team; and Web Development Team. Teams themselves may be stand alone or be composed of two or more subteams (Firesmith and Henderson-Sellers, 2002).

5 PROPOSED OPEN SUPPORT FOR TEAMS

Prior to introducing new work units into the OPEN repository, a conceptual model needs to be added as a method chunk to this repository. The conceptual model proposed by (Dagher, 2004) is a model

that represents the states of team building that a team is likely to go through, based on Tuckman's theory. The Team Building Activity is based on the theory of Tuckman's (1965) stages of Team Building (Table 2), emphasising that organisations need guidelines to support development (Hilburn, 2000). Also, the required tasks to accomplish the team building activity are stated below.

New Activity: Team Building

Team Building is proposed either as a part of the existing OPEN project management activity (i.e. a subactivity) or as a stand-alone activity. The proposed Team Building activity with associated (appropriate) tasks and techniques may guide the organisation to build that team.

Team Building can be selected to utilise people resources during project management. Many organisations find that teams are not well equipped with the necessary resources in order to cater for their needs (Gordon, 2002). Organisations also need to provide these individuals with skills and education to work as a team. Through utilizing team skill and learning, team members can form relationships, overcome conflict and suggest alternative ways to accomplish their assigned tasks.

Task: Identify Roles (Existing)

This task is focused on identifying project roles. The identification process involves determining the roles for a project such as a Software Engineer, System Designer, Systems Analyst and Tester. These project roles should be assigned to the people of the team.

Task: Assess Team Members (Existing)

This task was introduced by Serour (2003) to assess team members in terms of their qualification, knowledge and skills. The main objective is to link the project roles to the appropriate team member.

Task: Assign Roles (New)

This task aims to assign the roles to people required for the project. Assigning roles may be a many-to-many relationship between project roles and people. For example, one team member can have many project roles and one project role can be performed by one or many team member(s).

Task: Review Progress (New)

The review or evaluation of a team is vital for the improvement of team performance. This task can be performed to achieve both introspective and retrospective, that is to evaluate the progress of the team during and after the project (Table 3).

Review daily process
These meetings allow team members to meet and discuss the projects progress, tasks and any issues
Resolve conflicts among team members
Conflict in teams may arise due to the diversity of people working together each day
Discuss project problems such as time, cost and resources (Human and Technological)
Accept possible improvements for example performance
New ideas from team members to tackle problems encountered or manage a task can be considered
Provide continuous feedback
During the course of the project, team members (including the team leader) enforce the positives and negatives of the project as each milestone is completed

Table 3. Objectives of the Task: Review Progress

6 DISCUSSION AND CONCLUSIONS

In the context of situational method engineering, we have focused on methodology fragments relevant to teams and team building. In particular, we have sought new method fragments to augment the methodbase or repository of the OPEN methodological framework.

The importance of teams in SME should encompass how people need to be organized and managed to carry out the tasks of a project. Support for this “people” element of a methodology has been noted for its absence in most OO methodologies. Using well-accepted theories of team building and roles played as verified in the studies by Belbin (1981, 1993, 2000), we have identified several new method fragments (process components) to support working teams for the OPEN methodological framework. These new method fragments encompass the various necessary supporting elements for both individuals within a team and also the management support for team building and team maturation. This study has identified one new Activity for inclusion in the OPEN repository, that of Team Building. This is supported by two new tasks (plus the enhancement of some pre-existing ones).

Using an SME approach (here based on OPEN) to create an organizationally-specific methodology, this study has underlined that roles and teams are vital to coordinate people to accomplish any given project. In this paper, this has been accomplished by illustrating a new addition to the OPEN process framework, to support Team Building. In practice team building activity provides the ability to coordinate people for a specific project according to their skill and resources.

7 ACKNOWLEDGEMENTS

This is Contribution number 04/09 of the Centre for Object Technology Applications and Research (COTAR).

References

- Adair, J. 1986. ‘Effective Teambuilding’, Gower publishing Company Limited, England, 216pp.
- Belbin, R.M. 1981. ‘Management Teams’. Why They Succeed or Fail’, Butterworth-Heinemann, Oxford, England, 171 pp.
- Belbin, R.M. 1993. ‘Team Roles at Work’, Butterworth-Heinemann, Oxford, England, 141 pp.
- Belbin, R.M. 2000. ‘Beyond the Team’, Butterworth-Heinemann, 1st Edition, Oxford, England, 121pp, URL: <http://www.belbin.com/belbin-team-roles.htm> accessed on the 30/1/2004
- Borodzicz, E. 2002. ‘Individual and Group Learning in Crisis Simulations’, *Journal of Contingencies and Crisis Management*, 10(3): 139-148.
- Brinkkemper, S. 1996. ‘Method engineering: engineering of information systems development methods and tools’, *Journal of Information Software Technology*, 38(4): 275-280.
- Chowdhury, S., Endres, M and Lanis, L.W. 2002. ‘Preparing students for success in team work environments: The importance of building confidence’, *Journal of Managerial Issues*, (14)3: 346-359.
- Ciaburri, L. P. 1998. ‘Building the team or team building? There's a big difference’, *Rough Notes*, 41(1): 84-85.
- Constantine, L.L and Lockwood, L.A.D. 1994. ‘One size Does Not Fit All: Fitting Practices to People’, *Reprinted from American Programmer*, 7(12): 30-38.
- Dagher, L.M. 2004. ‘Introducing support for team building into software engineering: a case of enhancing the OPEN Process Framework’, *thesis for the degree of Bachelor of Science (Honours)*, Faculty of Information Technology, University of Technology, Sydney, 70pp.
- Firesmith, D.G and Henderson-Sellers, B. 2002. ‘The OPEN Process Framework’, Addison Wesley
- Gordon, J. 2002. ‘A Perspective on Team Building’, *Journal of American Academy of Business, Cambridge*, 2(1): 185-188.

- Graham, I., Henderson-Sellers, B and Younessi, H. 1997. 'The OPEN Process Specification', Addison-Wesley.
- Henderson-Sellers, B., 2003. 'Method Engineering for OO systems development', *Communications of the ACM*, 46(10): 73-78.
- Henderson-Sellers, B., Simons, A.J.H and Younessi, H. 1998. 'The OPEN Toolbox of Techniques', Addison-Wesley.
- Hilburn, T. 2000. 'Integrating the Personal Software Process (PSP) across the Undergraduate Curriculum', In: Proceedings of the ASEE/IEEE Frontiers in Education Conference
- Katzenbach, J.R. and Smith, D.K. 1993. 'The Wisdom of Teams: Creating the High-performance Organization', Boston, Harvard Business School, URL:
<http://tlt.its.psu.edu/suggestions/teams/about/definition.html> accessed on the 14/5/2003
- Kerth, N., 2001. 'Project Retrospectives: A Handbook for Team Reviews', Dorset House, 288pp.
- Kruchten, Ph., 1999. 'The Rational Unified Process: An Introduction', Addison-Wesley, Reading, MA, USA
- Kumar, K and Welke, R.J., 1992. 'Methodology engineering: a proposal for situation-specific methodology construction', Challenges and Strategies for Research in Systems Development (eds. W.W. Cotterman and J.A. Senn), J. Wiley, Chichester: 257-269.
- Martin, R.C., 2001. 'Agile Processes', URL:www.agilealliance.org accessed on the 4/02/2004
- Nador, S. 2001, 'The Fine Art of Team Selection', *Canadian HR Reporter*, October, 14(18): 15-19.
- Oxford English Dictionary., 2000. Oxford University Press, edited by Miranda Steel, 795pp.
- OMG, 2001. Unified Modelling Language Specification, version 1.4. formal/01-09-68 through 80 (13 documents). Object Management Group
- Pinson, L., Immonen, S and Ylitalo, J. 2000. 'Description of a team lifecycle', MUSCLES Technical Document', Task 2.4.3
- Rohlander, D. G. 1999. 'Effective Team Building', *IIE Solutions*, 31(9): 22-23.
- Serour, M. K., 2003. 'The Effect of Intra-Organisational Factors on the Organisational Transition to Object Technology', a *PHD Thesis, University Technology Sydney*
- Sheard, G. A and Kakabadse, P. A. 2002. 'From loose groups to effective teams, the nine key factors of the team landscape', *Journal of Management Development*, 21(2): 133-151.
- Stapleton, J. 1997, Dynamic Systems Development Method, Addison-Wesley, Harlow, UK, 163pp
- Ter Hofstede, A.H.M and Verhoef, T.F. 1997. 'On the feasibility of situational method engineering', *Information Systems*, 22(6/7): 401-422.
- Tuckman, B. W., 1965. 'Development sequences in small groups', *Psych Bulletin*, 63(6): 384-399.