



Proceedings of the
6th Educators' Symposium:
Software Modeling in Education at MODELS 2010
(EduSymp 2010)

Role Allocation and Scheme in Software Engineering Course Projects
Ghafour Alipour

10 Pages

Role Allocation and Scheme in Software Engineering Course Projects

Ghafour Alipour

Islamic Azad University, Hashtroud Branch
alipour@khayam.ut.ac.ir

Abstract: Role is a set of behavioural actions and responsibilities one takes in a specific situation. We play different roles with our family, at work as well as in other environments. Differences among the role definitions reach mainly from the different emphasis of the software development method itself. Agile software development methods define roles to enhance communication and to produce a better product. In this paper, we describe a role allocation and scheme method whose data were gathered in student's projects in software engineering course in the university.

Keywords: Role, Leading Group, Maintenance, Software engineering, DotNetNuke

1 Introduction

A team consists of at least two people who are working towards a common goal, where each person has been assigned a specific role to perform and where a completion of the mission requires some form of dependency among group members [Hpy00] (based on [Dyr84]). In the case of a software project, a team is a group of individuals who get together in order to produce a software product.

One accepted agreement upon roles in software teams is that each software team should have a project leader. However, just setting this role is not sufficient and additional issues should be addressed. Among other questions we should ask: What are the specific characteristics and responsibilities of the individuals? What other roles should be defined in a software team? What is the role of role assignment? These are only some of the questions with which this work deals with. The article is based on the analysis of projects developed by students at the Islamic Azad University, Hashtroud Branch.

In an academic environment the academic coach plays the customer role, defines the project requirements, refines them gradually as the project proceeds, and provides feedback and assessment. The academic coach also plays the coach role, and checks the main development activities that are correctly performed on time. Usually most of the development work is done when the academic coach is not present. In this paper another aspect of roles is illustrated. Specially, we advocate the idea that the responsibility for the software progress and success should be transferred to the students. It does not imply that the significance of the academic coach is reduced. This perspective gives the academic coach a better way to measure and evaluate the development process.

The work presented in this paper is part of a research that we have conducted in 2006 about the teaching of software development methods. During this research, DotNetNuke open source project, the most widely adopted Web Content Management Platform for building web sites and web applications on Microsoft .NET, was developed with the students. The research field is a project-based course in the field of web programming and in Extreme Programming (XP).



In Section 2 we describe approach towards the theme of roles. In Section 3 we present our role allocation and scheme to be implemented in software teams. In Section 4 we go down deep into the details of our role allocation and scheme as a measurement tool. We conclude the paper in Section 5.

2 Role approaches

A Role is a set of behavioral actions and responsibilities one takes in a specific situation. We all have many roles in our life. Ali, for example, has a teammate role, a mentor role, and maybe additional life roles. When Ali works on a software project together with his colleagues, he has a teammate role with respect to that project; when he teaches his friend how to solve a mathematical equation, he has a mentor role.

A relevant question to be asked at this stage is: How can one manage all these roles? The answer is quite simple. We do not have all our roles concurrently. Rather, in each real life situations we behave according to our role in that specific situation. Sometimes, the role is inherited naturally, like our child role, sometimes it is obtained as part of our profession as teachers or employees. In all these situations, we are able to solve conflicts that the role raises.

3 Roles in student teams

Agile software development methods define roles that aim to enhance communication and to produce a better product. Differences among the role definitions result mainly from the different emphasis of the software development method itself [Hsm02]. In what follows, we describe a role allocation and scheme we used in students software teams, mentioned above in the research environment. The scheme is based on 12 roles grouped into four major groups:

- Leading Group
 - Coach coordinates and solves group problems, checks the web forum and responds on a daily basis and leads development sessions.
 - Tracker measures the group progress by tests score and tasks estimations and manages studio boards and the group diary.
 - Methoder guides the teammates how to follow the principles of the software development method and is responsible for the method implementation.
- Customer Group
 - Customer tells customer stories, makes decisions pertaining to each iteration, and provides feedback and defines acceptance tests.
 - Acceptance tester works with the customer to define and develops acceptance tests, learns and instructs test-driven development.
- Maintenance Group
 - Presenter plans, organizes and presents iteration presentations, demos, and time schedule allocations.
 - Documenter plans, organizes and presents the project documentation, process documentation, user's guide, and installation instructions.
 - Installer plans and develops an automated installation kit and maintains the development environment infrastructure.
- Code Group
 - Designer maintains current design works to simplify design, searches for refactoring

tasks and ensures their proper execution.

- Unit tester learns about unit testing, establishes an automated test suite, guides and supports others in developing unit tests.
- Integrator establishes an integration environment, publishes rules pertaining to the addition of new code using the test suite.
- Code reviewer maintains source control, establishes and refines coding standards guides and manages the team's pair programming.

The following principle guides the formulation of this scheme: all the roles together should cover as many as possible of those practices that we wish our students to implement throughout the project development. The importance of this principle is illustrated by the integrator role. Teammates may be aware of the importance of continuous integration. However, this practice may be applied more properly if one of the team members actively pushes the team in this direction. Accordingly, we refer to roles as practice representatives.

As part of implementing our role allocation and scheme, we aimed at evaluating it and using it for evaluating the students' work as well as the software development process. Deriving quantitative metrics during development is part of software project management [PCS96, PM77]. As it turns out, the above role allocation and scheme provides systematic quantitative metrics during the development process which reflects the current status at each development phase. In what follows we elaborate on some related activities conducted with the students and the kind of data that they enable to derive.

4 Measures and role activities

We present data and analysis which students developed in teams of 10–12 students and how role activities were used as a measurement tool. Based on an examination of the role activities from the academic coach perspective, three main measures were defined.

- Role Time Measure (RTM)
- Role Communication Measure (RCM)
- Role Management Measure (RMM)

RTM measures the ratio between the times invested in development tasks relative to the time invested in role activities. RCM measures the level of communication in the team at each development stage. This measure evolves since each role holder needs to communicate with the other team members in order to perform his/her individual role efficiently. RMM measures the level of the project management. Since the role allocation and scheme aimed at covering all management aspects, maximal level is obtained when all role holders provide maximum role performance.

The three measures are on-going indicators derived from students' input at each of the following activities, an examination of students' electronic forums and their personal role reports.

Although XP emphasizes face-to-face communication among developers, several adaptations should be made when it is applied in a university environment. One adaptation, for example, is the addition of electronic communication via an electronic forum during the week between the face-to-face required team sessions. Data of students XP teams, all together 7 teams that presented in this paper was gathered by electronic forum of a 10-students team during a 14-week semester. The scheme of 12 roles was changed to fit that team: the customer was also the end user, and a student was documenter and presenter, all together 10 roles.



In the following section 4.1, the first two activities of the role allocation and scheme are presented.

4.1. Launching the role allocation and scheme

In the first meeting of the semester, the academic coach sits together with 10–12 students who do not know each other but should start working together and carrying out many tasks related to software development. Some of these tasks are about the actual development of software; others are not connected directly to the production of code. One of the tasks of the second type is role assignment. The first activity that is related to role assignment initiates students' relationship as teammates.

Activity 1: first acquaintance. Time: first meeting of the semester. **Task description:** You are going to develop a software product in a team of 10 students. Write down 10 roles that should be performed as part of your project. **Individual work (10 min):** Students write down their lists. **Discussion (20 min):** Students discuss their suggestions and try to generate an agreed list. **Summary:** The academic coach presents the role allocation and scheme. **Home activity:** The students are asked to prepare a personal prioritized list of roles they would prefer to perform during the semester.

During their work in Activity 1 the students actually start their acquaintance. They hear other students' ideas in general and their perception of roles in particular. They start noticing who are the knowledgeable students, who are the students tending to lead and manage, who are the students preferring being quiet, and who are the students contacting after the meeting. At the same time, the way the students handle the discussion and express ideas and feelings informs the academic coach about the nature of the team and of its individuals. Though the students receive at the end of this activity the pre-prepared role allocation and scheme, in most cases these roles as it turns out, have been suggested by them during the activity. This fact increases the students' confidence. Activity 2 describes the roles allocation.

Activity 2: roles distribution. Time: second meeting. **Discussion (10 min):** how should be assigned roles to teammates? **Task description (20 min):** Distribute the roles among the teammates. At the end of the task, suggest a list of role-teammate pairs. **Discussion (15 min):** Discuss how many percentages of your time you should dedicate to the performance of your personal role and how many percentages should be devoted to the accomplishment of your development tasks? **Reflection (after the meeting):** Express your opinion about the process of roles assignment in your team. How do you conceive of your role? What input would you expect to get from the other teammates with respect to your role?

During Activity 2 teammates are requested to formulate a process by which to distribute the roles among themselves. It is obvious that such a discussion, when it is conducted at the beginning of the semester, can contribute to communication and acquaintance processes. Furthermore, the role assignment itself has a direct influence on team communication. For example, in one of our teams, students suggested that roles should be assigned not according to appealing and wish, but rather, according to the area that one is not skilled in. The students explained that by this way the team members will have to communicate with other teammates, who are more knowledgeable about their roles to perform them properly. In addition, the students articulated that in this way each team member will benefit from learning a new topic.

The reflection at the end of the activity is very important. Students express their thoughts according to the role distribution process and provide the academic coach with a feedback. In addition, students started defining their role by expressing how they conceive it, and started establishing relationship with other team members that can support their role. As it is illustrated in the several students' quotes:

- “Everyone says which roles he (or she) prefers and then we distributed them. I think it is a pretty good way to do it, since each one received a role that he or she wished.”
- “. . . Considering there are 12 people in the team who do not know each other, it is difficult to assess in advance who fits best for each role. . .”
- “I am the unit tester. I think that this role is very important since code writing is not really a big wisdom. [. . .] In addition I will have to ask each of the teammates to check his or her parts . . .”
- “I’m the customer. I believe it is a major role, that is responsible for launching the entire project, by defining more precisely what the project is . . .”

In Section 4.2 additional activities, carried out during the development process, which support the project management are presented.

4.2. Project management by roles

In the following are highlighted some of the project stages from the role perspective. The activities in this part are performed on a regular basis; thus, enabling the academic coach to be aware of the project progress for better assessment of students work.

Instead of providing the students with a detailed requirement document, the students are presented with a short paragraph which describes the subject of the project. The student who owns the customer role starts telling customer stories by extending a brief description. The second meeting ends after the stories are sorted into three decks by the customer and the stories of the first iteration of the first releases are decided (once again, by the customer).

The third meeting of the semester is mostly dedicated to first design, formulation of development tasks, task allocation, time estimation, and load balance [Bek00]. Activity 3 starts in the third meeting.

Activity 3: stand-up meetings. Time: third meeting and on. Task description (10 min): *While standing, every student should describe briefly, in about 1 minute, what she/he has done during the last week with respect to the software project, and what he or she intends to do in the coming week.*

As it turns out, when students talk during the stand-up meeting they refer both to their development tasks and to their role. More specifically, they use this meeting to tell about what they have accomplished, about their role performance and about their expectations from the other teammates with respect to their personal roles. For example, one of the trackers said during one of the stand-up meetings that he asked everyone (in the electronic forum) to send him the number of hours they worked on the project during the last week but, unfortunately, not everyone responded. During the semester there are students who ask the academic coach to clarify their role definition since they do not understand it clearly, need further information about it or just want to further understand the academic coach's expectations. This kind of questioning leads to discourses about roles. Activity 4 demonstrates the individual dimension of roles. It can be performed with respect to other roles as well.

Activity 4: discourses. Task description (20 min): *Hasan wrote me about his role as installer. Since our project is DotNetNuke based portal, he wants to know if a*

publish mechanism is enough for the first version, if not, what other options exist. We all will benefit if we discuss this matter together.

Students appreciate such discourses since they receive answers and share information. Furthermore, such discussions provides the students with the feeling that they can raise any subject they wish, and that all the team members will discuss it in order to help them sort it out.

At the sixth meeting of the semester, a week before the presentation of Iteration 1 to the customer, Activity 5 is performed. It aims at helping students preparing their presentation. Since students usually do not have previous experience in presenting software products, they benefit a lot from this activity.

Activity 5: *towards presentation. Time: sixth meeting. Task description (20 min):*
Next week you will have 1 hour to present Version 1 of your product to the customer. Let's think together what the presentation should contain, how much time will be dedicated for each part, and how personal roles will be expressed.

There are three presentations to the client during the semester, at the seventh, eleventh and fourteenth (the last) meetings. After each presentation there is a feedback process that takes place during the meeting and a personal web-reflection that aims at encouraging the students to evaluate their work in the last period of time. As part of this reflection students are requested to evaluate their role performance as well as to grade it. Activity 6 outlines this reflection. Since this activity invited the students to go through a personal evaluation process, only the academic coach can see the summaries and evaluations that students present on the Web.

Activity 6: *evaluation. Time: seventh, eleventh and fourteenth meetings. Reflection (after each presentation):* 1. *Describe your work on the project during the last release from your role perspective. Address problems and achievements.* 2. *Grade your role performance on a 0–100 scale.*

The last activity presented in this article is requested by the academic coach periodically when she/he observes that it is needed. This need is raised mainly in cases when the academic coach feels that some roles are not carried out properly. Similarly to Activity 6 it also asks the students to summarize their role activities. However, as opposed to the personal manner of Activity 6, in Activity 7 the students are requested to publish their responses in the electronic forum and to provide feedback to the summaries presented by the other teammates.

Activity 7: *periodical summary. Time: periodically. Task description: All students are requested to summarize their role activities in the electronic forum. All students are requested to read the other team members' summaries and to give them feedback.*

Here are two statements in which the students tell their teammates how they expect them to attitude the procedures that their role establishes:

- “Hi, the attached file presents my plan how to execute the integration. Please read it and respond – we need to decide about it.”
- “The basic templates I've published for documentation are a bit oversimplified, yet I expected that they would be followed.”

4.3.Data and analysis

The measures according to data that was gathered by the electronic forums of 10-student teams in a 14-week 2007 fall semester was presented. With respect to RTM, collect data during the discussion of Activity 2 was started. At this stage most of the students predicted a RTM ratio of 30% (role)–70% (development), and only few predicted 40–60% or 20–80%. Students

talked about the RTM variation that is expected in some of the roles when comparing their role-peak weeks to their role-non-peak weeks. During the semester the students reported few times regarding their own RTM ratio. The semester average was 20–80% for all roles. Figure 1 shows RTM ratio for five 10-student team in 2007 fall semester. One exception was the coach who met the academic coach towards the presentation of the first iteration telling her or him that works 10–90% for 2 last weeks and felt that this is wrong. The academic coach discussed this issue with the students and then it came out that the coach did large parts of the other teammates’ roles. Discussing this matter with all students led to their improved understanding of the expectations and the scope of their role.

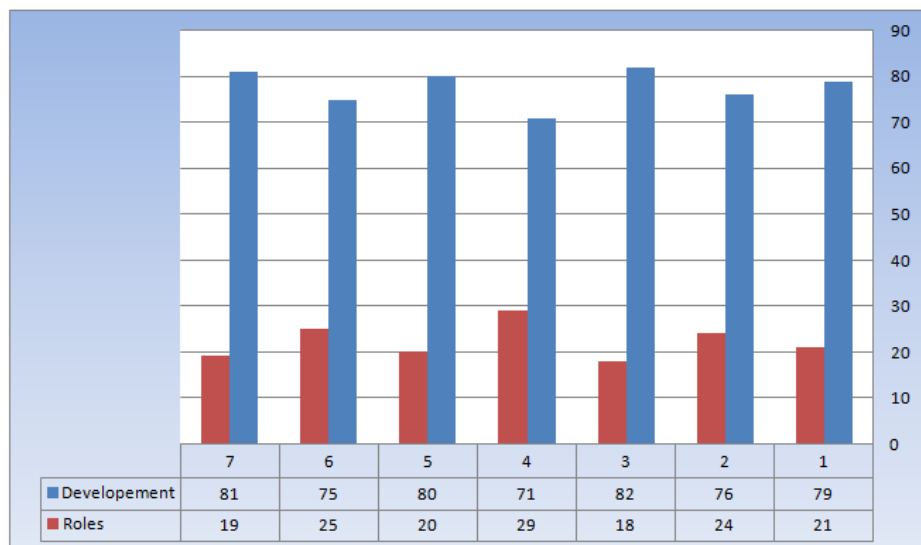


Figure 1. RTM of 10-student teams in 2007 fall semester

With respect to RCM, examining the electronic forum, 506 messages were sent during the semester, out of them 490 messages were sent by the 10 students. The rest of the messages were posted by the academic coach. Since there is load balancing in the part of development as part of the planning game, we refer to the number of messages as indicators for the role part. Table 1 provides their distribution among the role holders including their percentage from the total amount of messages.

Table 1, Forum messages according to rules

Groups	Total	Percent	Original	Response
Leading	240	49%	91	149
Customer	33	7%	23	10
Maintenance	75	15%	35	40
Code	142	29%	70	72

As observed, the leading group was the most communicative (49%) while the customer group was the least (7%). Looking into the messages' details, the coach indeed sent a response message for almost every message that another teammate has sent, and both the coach and the tracker played also the role of the continuous integrator since they felt the need to do it for sake the project. There are forum messages that were initiated by the role holder and others that are responses to other role holders. Table 1 also shows for each role the division of original messages and responses to others. As observed, most of the messages sent by the less number-messages-sender students were original. The students who responded for many messages are actually the core group of students whose roles were the most performed during the semester.

With respect to RMM, the management level of the team, per each iteration, is calculated. For each iteration, Total number of messages during all weeks of the iteration, and examine their distribution into original messages and responses were summed. Table 2 shows the data for the three iterations during this semester. We note that the communication in the electronic forum in the first iteration, which is seven week long, has started only after the first three weeks of the semester.

Table 2, Management level for each iteration

Iteration	No. of Weeks	No. Messages	Original	Responses
1	4	236	84	152
2	3	147	88	59
3	4	107	59	48

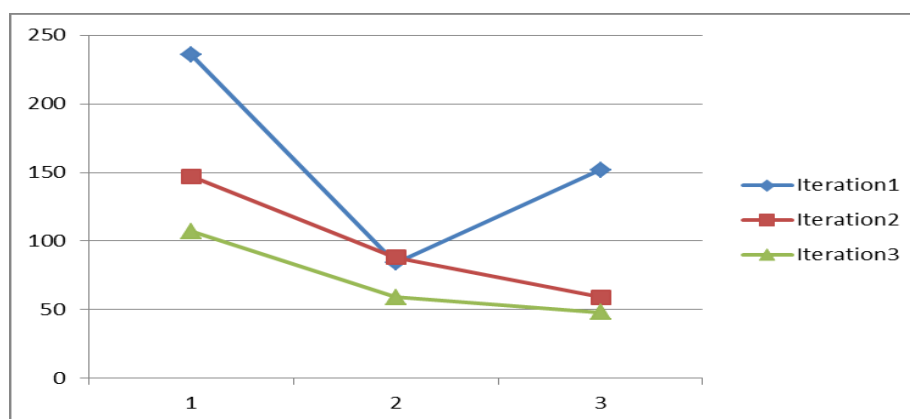


Figure 2. Weekly RMM per each iteration

Figure 2 shows weekly RMM for each iteration factored by the number of weeks. Thus, measuring RMM only with the number of messages per iteration per week, we conclude that iteration 2 is high managed since there is a reasonable ratio between original messages and responses, and total number of messages is high. The first iteration is the launching of the project and of the role allocation and scheme, and the last iteration is affected by the stress of the end of the semester. During the iteration 2, the students are more experienced with the project method and do their best.

5 Conclusion

Software development is a complex task that distributing the management of this task among teammates is suggested. According to which, each teammate has a special role, in addition to the development tasks, ease the project management by raising the accountability of all teammates in addition to an increased involvement and commitment. Our measurement model is not final, that is, it may be changed, updated and refined if needed.

This paper focused on the actual activities conducted with respect to the introducing of the role allocation and scheme to the students and on the way the scheme relates to the project management. Specifically, three main measures are introduced to cope with management, time and communication aspects, and data of a team in a semester was presented and analysed.

Bibliography

- [Hsm02] J.Highsmith, Agile Software Developments Ecosystems, Addison-Wesley, 2002.
- [Bek00] K. Beck, Extreme Programming Explained: Embrace Change, Addison-Wesley, 2000.
- [Dyr84] J.L. Dyer, Team research and team training: A state of the art review, Human Factor Review, The Human actors Society Inc., 1984.
- [Hpy00] W.S. Humphrey, Introduction to the Team Software Process SM, Addison-Wesley, 2000.
- [PCS96] K. Pulford, K. Combelles, S. Shirlaw, A quantitative Approach to Software Management , The AMI Handbook, Addison-Wesley, 1996.
- [PM77] L. Putnam, W. Myers, Industrial Strength Software Effective Management Using Measurement, IEEE Computer Society Press, 1977.