Exploring Workflow Mechanisms and Task Allocation Strategies in Agile Software Teams

Zainab Masood*, Rashina Hoda, Kelly Blincoe
SEPTA Research, Department of Electrical and Computer Engineering,
The University of Auckland, Auckland, New Zealand
zmas690@aucklanduni.ac.nz,
{r.hoda,k.blincoe}@auckland.ac.nz

Abstract. Task allocation is considered an important activity in software project management. However, the process of allocating tasks in agile software development teams has not received much attention in empirical research. Through a pilot study involving mixed open-ended and closed-ended interviews questions with 11 agile software practitioners working within a software development organization in India, we explain the process of task allocation as including three different mechanisms of workflow across teams: team-independent, team-dependent, and hybrid workflow; and five types of task allocation strategies: manager-driven, team-driven, individual-driven, manager-assisted and team-assisted. Knowing these workflow mechanisms and task allocation strategies will help software teams and project managers make more effective decisions around workflow and task allocation

Keywords: Task Allocation; Workflow; Allocation Mechanism; Agile Software Teams; Task Allocation Strategies;

1 Introduction

Successful project completion depends on how well and effectively the project activities are planned and managed throughout [1]. Primary project management activities include managing resources, task allocation, and tracking time and budget in the best possible way [2]. Several studies have researched task allocation in global and distributed software development using traditional or agile methods [3, 4, 5, 6]. A limited number of studies have assessed task allocation mechanisms practiced by Free/Libre Open Source Software (FLOSS) development teams; however, they did not cover commercial projects [7]. Overall, task allocation in agile software teams, which are meant to be self-organizing [9, 12], has not be studied.

*Corresponding Author: Building 903, 386 Khyber Pass Road, Newmarket Auckland 1023, New Zealand. Tel: +64 9 923 1377

We conducted a pilot study involving face-to-face interviews with 11 agile practitioners from three teams in a software organization in India. Thematic analysis [8] was performed to derive the different types of workflow mechanisms and task allocation strategies from the interview data. We identified three workflow mechanisms: team-independent, team-dependent, and hybrid workflow. We also identified five types of task allocation strategies: manager-driven, team-driven, individual-driven, manager-assisted and team-assisted. Identifying these mechanisms and strategies helped understand the flow and forms in which tasks arrives to the team and the basis on which tasks are classified and allocated.

2 Related Work

In traditional software development, the project manager plays a key role in task allocation and management and overall decision making. With the evolution of agile methods, software teams are meant to be self-organizing with high levels of autonomy, teams empowerment and mutual decision making in their everyday work [10, 12] including project management activities such as task allocation [11, 12]. In practice, however, agile teams are seen to display varying levels of autonomy as they gain experience of functioning in a self-organizing way [11]. How the varying levels of autonomy influence task allocation is not well understood. In particular, it is unclear how work flows to and within the team, how tasks are allocated on an individual level, and what are the different types and autonomy levels of task allocation in agile teams.

The research on task allocation in software teams has been largely dominated by distributed contexts in global software development. Imtiaz et al. in their recent survey-based study identified "functional area of expertise and phase-based" task allocation as the most common way of allocating tasks global software development [5]. Other studies, e.g. [4, 6], explored task allocation in distributed agile software development contexts through literature review and proposed models indicating further studies as a promising area of research. Crowston et al. 2007 [7] demonstrated the possible mechanisms of tasks allocation in community-based Free/Libre Open Source Software (FLOSS) development in self-organized volunteer teams. Their findings support self-assignment as one of the common ways of assigning tasks adopted by FLOSS teams. However, not much has been explored in the literature about task allocation mechanisms outside the FLOSS domain and specifically for commercial software development. Overall, much remains to be understood about how work flows to and within agile teams and how they practice task allocation.

3 Research Method

Our pilot study involved mixed open- and closed-ended interview questions with 11 agile practitioners. The overarching research questions were:

RQ1: How does work flow in agile teams?
RQ2: How does task allocation happen in agile teams?

Table 1. Team, Project Contexts and Participants Demographics (TS: Team Size, SP#:Participants; TX: total experience in years; X: agile experience in years; ATL:Assoc.Tech Lead; TL:Tech Lead; SSE: Senior Software Engineer)

Team	TS	Software Method	Project Area/ Context	SP#	Role	Age Group	TX	AX
T1	10-15	Scrum	Digital Marketing/ Features & Maintenance	SP1	TL	31-35	10-11	6-7
				SP2	SE	21-25	2-2.5	1
				SP3	ATL	26-30	4-5	4-5
				SP4	SE	21-25	2.5	2.5
T2	5-10	Scrum	Analytics/ Features	SP5	TL	36-40	7	7
				SP6	SSE	26-30	4	2
				SP7	TL	31-35	7.5	7.5
T3	15-20	Kanban	Cloud Services/ Migration & Enhancement	SP8	TL	31-35	5.5	5-6
				SP9	ATL	26-30	4	2
				SP10	SSE	21-25	3.5	1
				SP11	ATL	26-30	4.5	2

3.1 Participant Selection and Description

An invitation to participate was sent out to members of the Agile software community of India. The company willing to offer a maximum number of teams and participants was selected. Eleven software practitioners from three agile teams working in this digital technology company were included (one additional participant was later dropped since they were the sole representative of a fourth team.) Participants were experienced software practitioners and were using agile methods, either Scrum or Kanban, including key agile practices such as Daily Team Meetings, Release and Iteration planning, Pair Programming, Review meetings and Retrospectives. Teams were collaborating with off-shored customers or product teams in the USA through Google Hangout, Skype or Webex. The project management tool used by all teams was Jira. Team, project and participants' details are profiled in Table1. Data Collection and Analysis

We conducted face-to-face interviews lasting 30-40 minutes with each participant using a combination of open- and close-ended questions about their current projects applying agile methods. Initial questions gathered participants' demographical data, details related to the project, team and the agile methods used. Most other questions focused on task allocation process e.g. how, when and from whom the teams receive the tasks and how the tasks are allocated among the teams and the individuals. These were mostly open-ended questions to allow a range of answers, with some choices being given to facilitate the interviewees during the interview.

All the interviews were recorded with detailed notes taken during the interview. Interview data was transcribed and analyzed manually using thematic analysis [8] to derive the

common themes, i.e. patterns of workflow mechanisms and task allocation strategies common across the participants. This was led by one of the authors and supported by the other two through careful reviews and discussions.

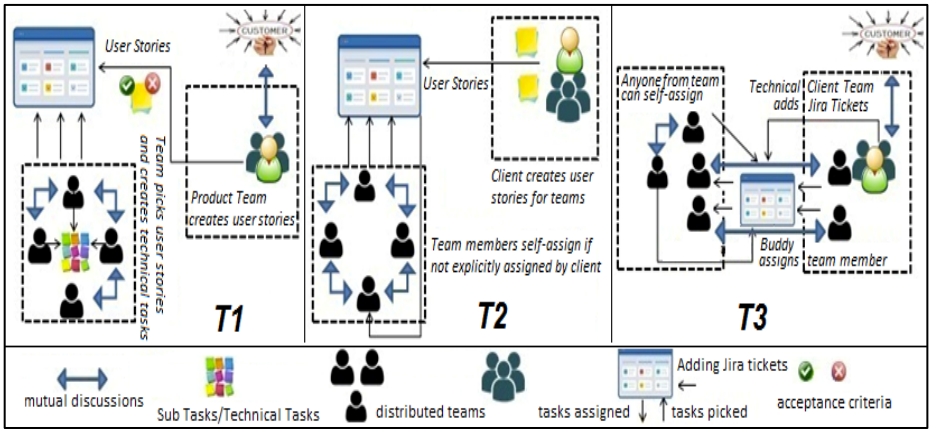

Fig. 1. Teamwise Task Allocation Mechanisms (T1: team independent workflow; T2: team dependent workflow; T3: hybrid workflow)

4 Findings

In answer to RQ1, we identified three distinct workflow mechanisms (illustrated in Fig.1.) that describe how the teams receive the work from the relevant stakeholders: team-independent, team-dependent, and hybrid workflow. Additionally, in answer to RQ2, we found five different task allocation strategies based on how tasks were allocated within the team: individual-driven, manager-driven, team-driven, manager-assisted and team-assisted.

4.1 Team Workflow Mechanisms

Team Independent Workflow: In this workflow, the tasks are defined irrespective of the team location (US, India). Tasks comes to both teams from Product Owner mostly in form of user stories during sprint planning meeting. Members of all teams individually pick and break user stories into technical tasks. The work allocation is done by volunteering for tasks through mutual discussions. For example, one participant explained:
"They[Product Team] bring whole description of the ticket[user story]...Everyone is in sprint planning meeting, every developer I should say and then ticket by ticket we volunteer, they do not assign any name." SP1, Tech Lead.

Team Dependent Workflow: Client defines the tasks for respective teams (US, India) separately as user stories during fortnightly iteration planning meeting. Before sprint

planning meeting, the team (T2) go through their stories and team members allocate the tasks either individually or through mutual consensus. SP7 described the workflow as follows:
"Client creates user stories then one day before sprint planning we [T2] go through stories which are meant for India team and we pick whatever we want to do." SP7, Tech Lead

Hybrid Workflow: Team T3 was seen to follow multiple workflow mechanisms, but tasks are typically allocated during a monthly release from the USA technical team, who collaborates with the client. For a few members of the team, the USA team creates Jira tickets with a set priority and complexity level. As specified by SP9:
"Now that teams have been divided so they have to work according to the tasks that are assigned to those particular teams only so it's not like that X team can work on team Y cards." SP9, Associate. Tech Lead
For other team members, work comes as features with a defined priority and release date from the USA team. These features are selected by the Tech Lead in USA, who breaks them into tasks and sub-tasks and allocates them to their 'buddy' programmer in India.
"*So the client decides the criticality and to which release these [cards] will belong so once the lead has decided that then pair[buddy] can pick up." SP10, Sr. Software Engineer*

4.2 Task Allocation Strategies

In **Manager-driven Task Allocation**, the manager/client/technical-lead allocates tasks to the team members with names against the tasks as stated by a participant, where the 'buddy' was a senior Tech Lead in the USA: *"Nowadays I am given task by my buddy." SP11, Assoc. Tech Lead*

In **Manager-assisted Task Allocation**, tasks are allocated with some assistance from the manager/client/technical-lead to the team members. As a technical lead, SP1 mentioned assisting team member with picking tasks: *"'Hey [name] you should do this [task]', let say he is new and he doesn't know [so] I help him, 'pick this one because this is lesser complex'." SP1, Tech Lead*

In **Team-driven Task Allocation**, the team discusses and mutually decides who will perform which task, for example: *"We are three people [in the team] so mutually decide who will do [what]." SP6, Sr. Software Engineer*

In **Team-assisted Task Allocation,** every team member self-assigns tasks with some assistance from fellow team members, for example*: "So any of the pair[s] can pick up [a task]." SP10, Sr. Software Engineer*

In **Individual-driven Task Allocation**, tasks are self-assigned i.e. selected and managed individually without any assistance from others. For example, SP4 quoted practicing self-driven allocation: *"Mostly we volunteer it." SP4, Software Engineer*

5 Discussion

We identified five task allocation strategies. Four of these strategies involve either the team as a whole or the manager/client in the task allocation process, making it evident that the task allocation mostly takes place through assistance or mutual discussions. In other words, task allocation strategies rely on collective decision making. A prior study [13] has shown that agile teams make effective decisions collectively compared to individual decisions, benefitting from collective knowledge and experiences.

Another aspect is that for high priority tasks all mechanisms agree on a common allocation method, i.e. tasks are directly allocated to a skilled and experienced person, an aspect supported by previous research [7].

Our study supports the different levels of autonomy evident on agile teams [11] as we found evidence of varying management approaches: manager-driven, manager-assisted and team-driven. Additionally, we also identified a new level: individual-driven task allocation.

With respect to the effectiveness of their current strategy, all the teams reported being satisfied, but some participants shared a few challenges, e.g. vagueness or missing clarity on tasks was the most commonly reported challenge. One participant (SP10) mentioned that with their current task allocation strategy (Team-assisted), work at times is not evenly distributed. Another participant (SP1) revealed drawbacks of picking tasks remotely. Since their client and the USA team are co-located they were perceived to have an advantage in picking tasks over SP1's India team. However, these challenges are not directly related to task allocation, rather, they are also linked to requirements clarity issues and the distributed nature of the team. This illustrates that task allocation is impacted by many factors.

This research study can serve as a basis for exploring other task allocation strategies and internal workflow mechanisms of agile teams. This pilot study included only 11 interviews from the same organization which signifies a limited dataset and context. Our larger study will interview more software teams and individuals representing different roles. Future work can focus on evaluating the effectiveness of the strategies.

6 Conclusion

This study presents a preliminary understanding of workflow mechanisms and task allocation strategies in agile teams. Clients typically provide high-level requirements as features or user stories to the agile teams who then break them down into technical tasks or sub-tasks by themselves or directly allocate them to team members. The team members then select them individually or through mutual discussions within the team. Allocation of tasks usually takes place during iteration or release planning. The findings of this study

demonstrate that there are multiple types of task allocation strategies practiced by agile teams based on what suits the completion of the work in the best possible way. A common mechanism found in a majority of the teams is that if the priority of the task is high, then the task is allocated to the most suitable person directly. Also on average, the practice most commonly followed is that the team members collaborate with each other and with their manager/client when assistance is needed.

References

1. Pinto JK, Slevin DP (1988) Critical success factors across the project life cycle. Project Management Journal, Vol. 19 No. 3, pp. 67-75.
2. Hoda R, Noble J, Marshall S (2008) Agile Project Management. New Zealand Computer Science Research Student Conference.
3. Lamersdorf A, Munch J, Rombach D (2009) A survey on the state of the practice in distributed software development: Criteria for task allocation. 2009 Fourth IEEE International Conference on Global Software Engineering. doi: 10.1109/icgse.2009.12
4. Filho MS, Pinheiro PR, Albuquerque AB (2015) Task allocation approaches in distributed agile software development: A Quasi-systematic review. Advances in Intelligent Systems and Computing. Springer Nature, pp 243–252.
5. Imtiaz S, Ikram N (2016) Dynamics of task allocation in global software development. Journal of Software: Evolution and Process. doi:10.1002/smr.1832
6. Mak D, Kruchten P (2006) Task coordination in an agile distributed software development environment. 2006 Canadian Conference on Electrical and Computer Engineering. doi: 10.1109/ccece.2006.277524
7. Crowston K, Li Q, Wei K, et al (2007) Self-organization of teams for free/libre open source software development. Information and Software Technology 49:564–575. doi: 10.1016/j.infsof.2007.02.004
8. Clarke V, Braun V (2014) Thematic analysis. Encyclopedia of Critical Psychology. Springer Science + Business Media, pp 1947–1952. doi:10.1007/978-1-4614-5583-7_311
9. Moe NB, Dings T, Dyb T (2008) Understanding self-organizing teams in agile software development. 19th Australian Conference on Software Engineering (aswec 2008). doi: 10.1109/aswec.2008.4483195
10. Highsmith JA (2009) Agile project management: Creating innovative products, 2nd edn. Addison-Wesley Educational Publishers, United States.
11. Hoda R, Noble J (2017) Becoming Agile: A Grounded Theory of Agile Transitions in Practice, IEEE International Conference on Software Engineering (ICSE2017).
12. Hoda R, Murugesan LK (2016) Multi-level agile project management challenges: A self-organizing team perspective. Journal of Systems and Software 117:245–257. doi: 10.1016/j.jss.2016.02.049
13. Drury M, Conboy K, Power K (2012) Obstacles to decision making in agile software development teams. Journal of Systems and Software 85:1239–1254. doi: 10.1016/j.jss.2012.01.058