

Motivation for Self-Assignment: Factors Agile Software Developers Consider

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@auckland.ac.nz, k.blincoe@auckland.ac.nz

Abstract Self-assignment is a fundamental to self-organizing teams, however, the factors that motivate agile developers to self-assign tasks are not known. In a pilot study of a software organization in India, we conducted interviews with 12 agile developers. Using descriptive statistical and thematic analysis, we identified three groups of factors motivating the self-assignment: task-based, developer-based, and opinion-based factors. We found that agile developers give precedence to task-based and developer-based factors over opinion-based factors. Furthermore, majority of the participants preferred to self-assign tasks that give them an opportunity to learn new technology, tools, and domains. This knowledge can be applied to let team members better select tasks of their interest and to conduct future studies into this topic.

Keywords: *self-assignment; influencing factors; motivation*

I. INTRODUCTION

Motivation is a critical determinant of project success in software projects [1, 2]. Several motivating factors considered by managers and developers have been explored in literature and industry that help achieve project goals in various contexts and settings [3, 4, 5]. We address the research question: “*What factors motivate agile team members to self-assign tasks?*” To answer this, we conducted a pilot study using interviews with 12 agile developers.

We found that developers consider a set of motivating factors which fell into three groups: *task-based factors* such as technical complexity of the task, business priority of the task, and customer demand (irrespective of priority); *developer-based factors* such as technical ability to perform the tasks, previous experience with similar tasks, and opportunity to learn new technology, tools, or domains; and *opinion-based factors* such as opinions of the team members and opinions of the manager.

II. RELATED WORK

Motivation in software engineering influences productivity, software quality, and project success [1, 2]. For software engineers and developers, some of the motivational factors identified are clear goals, personal interest, job satisfaction, technical challenges, personality of developers and level in career [1, 2]. Factors such as expertise, autonomy, and variety of work play a significant role in task allocation decisions in distributed software development (DSD) [3, 5]. Managers in DSD take into account factors like technical

knowledge, expertise in business, and proximity to client when allocating tasks [4].

Agile developers are meant to practice self-assignment [6]. Agile teams have been reported to pick up tasks of their own choice rather than being allocated by a manager. Self-assignment of tasks is also observed as a practice in open source software projects [7]. However, the factors that motivate agile developers to self-assign tasks are not known.

III. RESEARCH METHOD AND DESIGN

As a part of a larger study, we conducted a pilot study involving 12 agile practitioners from a software organization in India. Face-to-face interviews were conducted on site in India. Interviews were recorded and transcribed for statistical and thematic analysis [9].

The interview questions were structured into three sections to collect participant demographics, project and team information; and task allocation related information through open-ended and close-ended questions with opportunities for providing examples and discussions. An example of an open-ended question is: *What are the most important things you take into consideration when self-assigning tasks? Does it vary? How/when?*

We sent an invitation to the Agile community members in India. A digital organization developing different software solutions and offering the most number of participants and teams was selected. Participants were performing various roles such as tech lead, assoc. tech lead, senior software engineer or software engineer. The age of the participants varied from 25 to 35 years with overall working experience between 2 and 10 years and agile experience between 1 and 7.5 years.

We analyzed the close-ended questions using descriptive statistical analysis to understand motivating factors. The factors fit into three main groupings: *task-based*, *developer-based*, and *opinion-based factors*. This grouping helped us to understand the implications of these factors. The open-ended questions fed the qualitative thematic analysis [9] and enabled a discussion of underlying reasons and emergence of new factors.

IV. FINDINGS

Task-based Factors included *technical complexity* which covers the technical aspects of the task such as technologies involved; *business priority of the task* which gauges the

importance of a task in terms of business outcomes and priorities; and *customer demand (irrespective of priority)* which refers to what the customer wants and distinguishes it from the task's business priority e.g. customer can demand a particular task be done despite it being low on business priority for various reasons.

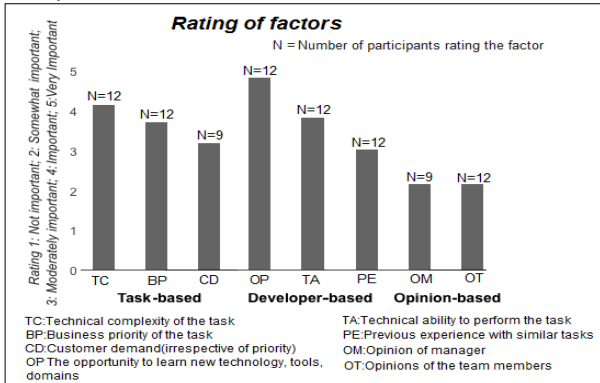


Figure 1 Motivational Factors for Self-Assignment – Average Ratings by Developers

Figure 1 indicates that, within task-based factors, technical complexity is considered as an important factor with an average score of 4.2 out of 5. For customer demand (avg. 3.2), two of the developers considered this similar to the business priority factor (avg. 3.7) and one stated the client never comes up with such demands and so this does not apply to their current business model.

Developer-based Factors included *opportunity to learn new technology, tools, and domains* which refers to task that will help developers learn new technology; *technical ability* which captures individual's ability to perform tasks with technical knowledge, expertise and skills; and *previous experience with similar tasks* [3].

Based on the results (Fig.1), it can be inferred that developers rate the opportunity to learn new technology as the most important factor with an average score of 4.8.

Opinion-based Factors included *opinions of manager* which refers to an individual considering their manager's opinion as a significant factor while picking up tasks; and *opinions of team members* which refers to how the choice of a task is made based on the opinions of other team members.

Participants rated opinions of the team members and managers as somewhat important (avg. 2.2). Three participants did not provide any rating to opinion of manager as with their current task allocation method the manager directly assigned the tasks.

Other Factors that we identified in the open-ended responses included *task size, deadline, and deference to juniors*. One participant stated the size of the task and completion time were 'other factors' they considered (i.e. if the task required significant effort and the time to complete is short they would not pick it up). Another participant mentioned deference to juniors as a factor, since senior developers sometimes back off from a task and let the junior developers utilize the opportunity to learn.

V. DISCUSSION

We identified that while the opinions of managers and team members are considered, the final assignment decision

is based on factors related to the developer and the task itself such as technical ability to perform and technical complexity of task respectively. This can be considered indicative of high levels of autonomy – a desirable trait for self-organizing agile teams [6, 8]. Two factors specified by the participants, task deadline and size, also can be categorized as task-based factors. These factors can be investigated in the future, as they may account for cases where managers move from the self-assignment model to direct task allocation [8]. The other factor brought up by one of the participants, deference to juniors, can be included in future studies for further analysis.

Another implication of these findings is the inter-play between these factors and the trade-offs that developers perform to reconcile conflicting priorities. Practitioners mentioned situations that were context-driven and could not be considered in isolation e.g. familiar tasks are normally not preferred but if the priority of the task is high, it overshadows personal preference, similar to results from [3].

Furthermore, developers may not always self-assign tasks depending on these factors alone e.g. one participant stated that even if the developers are technically skilled to complete a task, they still may not self-select if it does not fall within their technical comfort zone.

VI. CONCLUSION

Through a pilot study of motivating factors for self-assignment involving 12 agile software developers, we found that *task-based* factors and *developer-based* factors are given higher priority over *opinion-based* factors while self-assigning tasks. While these factors are prioritized, self-assignment decisions are often context-driven. Developers may deviate from their usual practice depending on circumstances and the trade-offs between different factors e.g. developers may give precedence to familiar tasks if the business priority of those tasks is high. We will incorporate the additional factors identified in an extended survey.

REFERENCES

- [1] T. Hall, H. Sharp, S. Beecham, N. Baddoo, and H. Robinson, "What do we know about developer motivation?," IEEE Software, vol. 25, no. 4, pp. 92–94, Jul. 2008.
- [2] S. Beecham, N. Baddoo, T. Hall, H. Robinson, and H. Sharp, "Motivation in software engineering: A systematic literature review," Information and Software Technology, vol. 50, no. 9–10, pp. 860–878, Aug. 2008.
- [3] S. Imtiaz and N. Ikram, "Dynamics of task allocation in global software development," Journal of Software: Evolution and Process, 2016.
- [4] M. S. Filho, P. R. Pinheiro, and A. B. Albuquerque, "Task allocation in distributed software development aided by verbal decision analysis," in Software Engineering Perspectives and Application in Intelligent Systems. Springer Nature, 2016, pp. 127–137.
- [5] B. Tessem and F. Maurer, "Job satisfaction and motivation in a large agile team," in Agile Processes in Software Engineering and Extreme Programming. Springer Nature, pp. 54–61.
- [6] R. Hoda and L. K. Murugesan, "Multi-level agile project management challenges: A self-organizing team perspective," Journal of Systems and Software, vol. 117, pp. 245–257, Jul. 2016.
- [7] K. Crowston, Q. Li, K. Wei, Eseryel, U. Yeliz, and J. Howison, "Self-organization of teams for free/libre open source software development," Information and Software Technology, vol. 49, no. 6, pp. 564–575, Jan. 2007.
- [8] R. Hoda and J. Noble, "Becoming Agile: A Grounded Theory of Agile Transitions in Practice." IEEE International Conference on Software Engineering, Buenos Aires, 2017.
- [9] V. Clarke, V. Braun, "Thematic analysis," in Teo, T., ed. (2014)