A Proposal for Improving Project Coordination using Data Mining and Proximity Tracking

Elizabeth Bjarnason and Håkan Jonsson

Lund University, Sweden
{elizabeth, hakan.jonsson}@cs.lth.se

Abstract. Coordination is an important success factor for a development project. Communication gaps, e.g. between product owners shaping the requirements and testers verifying the developed software can result in wasted effort and unsuccessful products. We propose improving the communication between project members with recommendations of whom to interact with and what to discuss based on link prediction in multi-layered proximity-based social graphs based on data mined from project repositories. We plan to explore and validate these ideas through prototyping and by applying a design-science approach in collaboration with an industrial partner.

Keywords: communication, data mining, social networks, machine learning

1 Introduction

Developing software is a knowledge-intense activity that greatly relies on communication [20]. Development projects struggle with information overload [10] and managing huge amounts of complex and ever-changing information spread over multiple roles and repositories, e.g. requirements- and test-management systems. Time is wasted on locating relevant information [16], and coordinating a project is a challenge [3, 4].

Distances between individuals and teams can negatively affect communication frequency and quality [5]. Apart from physical distance that reduces the rate of direct communication [1, 4], there are also cognitive and psychological distances that can cause communication gaps and misunderstandings. Good communication practices address distances and improve coordination [3, 5, 18]. The information flow can be enhanced by collaboration tools that improve artefact navigation [16] and awareness of ongoing activities [18]. Furthermore, identifying communication patterns can enable improving the communication and even predict the risk of quality issues [20].

In this short paper, we outline an idea for improving project collaboration and requirements communication by a conversation recommendation application that prompts, e.g. a product owner and a tester to discuss related issue reports and requirements. The recommendations are based on data mined from project repositories and from tracking people's physical interaction and proximity with others. The approach could support requirements coordination in larger projects for the following scenarios:

Interaction Recommendations. When users are in close proximity to each other the application identifies related tasks associated with these users, and suggests a conversation around these, e.g. similar requirements, issues reports or test cases.

adfa, p. 1, 2011.

Coordination Recommendations. A user is provided with recommendations on who to coordinate with based on related tasks and previous physical encounters. For example, a product owner is prompted to contact a certain tester whom is defining test cases related to requirements currently discussed with a customer.

Orientation of New Team Member. A new tester can browse other team members and their tasks and those related to her tasks are highlighted, thus indicating possible sources of requirements information.

Project Meeting Support. The application can provide a list of items for the detected participants to discuss based on their ongoing tasks, thus highlighting current requirements topics. The application can also outline meeting notes containing the proposed topics and the set of participants.

Human resource management. Managers can use the application to identify key resources in their organisation. For example, individuals that are critical to the requirements-test coordination in the organisation, so called *information brokers* [8].

Organisation and office space management. The application can detect which people and teams that interact, or should interact, on a regular basis, and can suggest how to organise the staff and how to plan office space. For example, place product owners close to the teams and testers that they frequently interact with.

Related work is described in Section 2. We outline our proposal in Section 3 and our future plans in Section 4.

2 Background and Related Work

Our research proposal is based on providing recommendations derived from information about connections between project members. This information consists of physical proximity, information derived from mining various computer systems used by these engineers, and connections between the artefacts stored in these systems identified by natural language processing techniques.

2.1 Supporting Direct Interaction with Physical Proximity Tracking

Tracking people's physical location, and their proximity to others and to devices poses interesting possibilities. Proximity tracking has been researched in the context of social contacts (e.g. Facebook) and software development teams. In both cases Bluetooth via mobile phones was used due to not requiring additional communication infrastructure.

Eagle and Pentland deployed the first system in an office setting using mobile phones for proximity sensing with the purpose of introducing serendipitous contact formation [9]. Similarity of user profiles was used to recommend people to connect to.

Corral et al. used proximity measures to track and identify software engineering activities through the use of a mobile application and tagging of physical devices [7], e.g. computers with Bluetooth dongles. Corral et al. identified work sessions from this proximity data through detection of patterns for, e.g. pair programming.

Similarly Jonsson and Nugues used Bluetooth-enabled mobile phones to capture meeting participation based on proximity to others and/or to a room-specific mobile phone [12]. The participants' social identities were used through a features of the Proximates system [12]. Proximates matches a physical device's identifier (BT MAC id) with user identities of, e.g. Facebook, and provides functionality for scanning and logging interactions. These features were also applied in a reminder application [13] that prompts the user when in physical proximity to social contacts for which a reminder has been set, e.g. repay loan, discuss holiday plans etc.

Discussion. Physical proximity is a very promising concept for tracking engineers' direct interactions including one-to-one and group meetings. The Proximates functionality would enable connecting the user of a mobile phone to information found in software engineering repositories, e.g. issues and tasks currently assigned to this person. The ability to identify meeting participants could be used to detect and monitor important interaction and communication points for a project.

2.2 Identifying Communication Patterns with Social Network Analysis

Mining and analysis of social networks is an active research topic that is used for several applications including prioritisation of e-mails [21] and enhancing collaboration [8, 20]. Social networks have been constructed from sources such as e-mail communication [2, 14, 21] and software engineering repositories, e.g. systems for managing tasks [8] and issues [20]. This information can be used to address information overload and enhance communication by suggesting relevant information points, e.g. people.

Yoo et al. propose a technique for automatically prioritising e-mails by user-specific priorities derived from personal social networks [21]. Their technique is designed to enable training of the algorithms per individual rather than for a whole organisation, thereby addressing privacy issues. Networks are constructed by clustering nodes, e.g. according to recipient lists. Importance of contacts is derived from *centrality measures*.

Wolf et al. derive task-based communication between engineers from project artefacts such as source-code changes, and issue reports [20]. Their tools and methodology for constructing social networks have been applied in an industrial project environment and used for predicting software build failures based on communication patterns.

When individuals belong to multiple social networks these can be analysed using *multi-layer social network* techniques. For example, Magnani and Rossi propose a model and an extension to standard measurements for analysing such networks [17].

Discussion. The techniques for constructing social networks from repositories could be used to identify people working on related tasks, information hubs etc., which is relevant for our research. We plan to extend on existing work by automatically deriving recommendations. Concerning personal integrity, Yoo's work on personal network data is promising due to addressing this by limiting data mining to individual users.

2.3 Deriving Useful Connections with Machine Learning Techniques

Borg proposes using *machine learning* and *information retrieval* techniques for supporting engineers in navigating the large and volatile information landscape of software development projects [6]. In particular, Borg has applied these techniques to support automated allocation of issue reports and to provide recommendation support for impact analysis. New issue reports have been allocated to development teams by training multiple classifiers on the textual content of previous issue reports and then combining these when analysing new (incoming) issues. Similar prediction accuracies as for the manual allocation were obtained, thus saving time in performing this task while retaining the same quality level of the outcome. For change impact analysis, Borg uses traditional information retrieval techniques to locate textually similar issue reports. These are then used as input to identify a set of artefact linked to these (related) issue reports. These artefacts are weighted using measurements of textual similarity, relative distance and centrality in the derived network of artefacts, thereby providing a ranked list of artefacts according to probably of being impacted by the new change. Initial evaluations show that the algorithms predict 30-40% of previously reported impact when considering 5-10 of the first recommendations and that the approach could motivate and support engineers in performing and in validating change impact analyses.

Erman et al. apply similar techniques for identifying related test failures when analysing large amounts of test results from automatic test cases in a multi-branch project environment [11]. In this work, test case name, error message and test environment context were weighted and failed test executions were clustered based on cosine similarity in the vector space model [19] using these three components. The technique is now used in production at the case company for which it was developed. Examples of supported scenarios include cross-referencing failures across branches and assessing if a problem is global or local, i.e. isolated to one branch.

Discussion. In our context of enhancing communication, we could apply these approaches and use machine-learning and information-retrieval techniques for locating related entities. By identifying related issues or tasks, potentially useful contacts can be found that may have information that could facilitate the current work. In addition, identifying other potentially relevant artefacts could provide pointers to additional repositories to mine. Furthermore, Borg's clustering techniques may be relevant to apply, since our case company also performs extensive automatic testing in a multiple branch environment. By clustering related test failures the teams, projects or individuals responsible for the connected branches may be relevant contacts.

3 Solution Proposal

The proposal will be investigated through a design-science approach [15] by iteratively constructing a prototype and evaluating it through case studies at a development company. The initial focus is to improve coordination from the testers' perspective.

The developed system will consist of a server and an application that runs on the users' mobile phones. The server will interface a number of project repositories for test cases, issues etc. and connect users' devices with their user ids for these repositories using Proximates (see Section 2.1). Social networks will be constructed from information mined from the project repositories using similar techniques as are described in Section 2.2. These *multi-layer networks* will contain information on both existing entities and connections, and potential connections identified using *machine learning* and *information retrieval* techniques similar to those in Section 2.3. In addition, the physical encounters, as tracked by Bluetooth, will be represented in an additional network layer.

Beneficial interactions and topics relevant to two or more *proximate* users will be derived based on *link predictions* in the constructed *social graph* and on the principle of triadic closure in sociology. These will then be suggested via the end-user application. For example, when registered users are noted as being physically close, the social network will be queried to identify if these users have related topics that could be discussed, if so an interaction is recommended. Similarly, if a user at a project meeting requests suitable topics, the social network is filtered for the identified participants and relevant topics are ranked and presented.

Key coordinators can be identified from social graphs by *centrality analysis*. For example, *betweeness centrality* can identify members critical to the information transfer and links or nodes that need to be reinforced through redundancy. Furthermore, *eigencentrality* can be used to identify influential key members. This information can help management in adjusting office seating and organisational structures.

In addition to providing useful functionality to the users, the system will track their physical interactions including identified meetings. The users' responses to interaction recommendations will also be tracked in order to train the system and to evaluate the underlying algorithms used to derive recommendations.

The prototype will be designed and evaluated for a development team at a large development company. This team consists of product owner, developers, testers, project managers etc., and interacts with other development teams and testing units both on- and off-site. The impact on requirements communication and its effect on project coordination, e.g. lead times and software quality, will be evaluated. The research will also consider the ethical and legal aspects of tracking and using information connected to individuals and their privacy, e.g. in selecting and displaying information, and in storing it. This is also an important aspect in securing participants for the evaluation.

4 Summary and Future Plan

We propose improving project coordination by providing recommendations for what to discuss with project members who are physically close. These recommendations will be based on information mined from project repositories using a combination of techniques from social network analysis and machine learning. In addition to enhancing communication and collaboration between engineers this research will provide a platform for studying team interactions.

We plan to investigate and evaluate these ideas in close collaboration with an industrial partner with a focus on improving requirements communication towards test engineers. A prototype will be iteratively developed and evaluated through use in a development project. Apart from evaluating the impact on requirements-test coordination, the research is expected to yield new insights and contributions into how team members in general and test engineers in particular interact and collaborate, and how to improve development processes in order to enhance project coordination.

5 References

1. Allen T. Managing the Flow of Technology. Cambridge, MA, MIT Press, 1977.
2. Bird C, et al. Mining email social networks. Proceedings of the 2006 International Workshop on Mining Software Repositories. ACM, 2006.
3. Bjarnason E, et al. Challenges and Practices in Aligning Requirements with Verification and Validation: a Case Study of Six Companies. Empirical Softw Engin. 19.6: 1809-1855, 2014.
4. Bjarnason E, Sharp H. The Role of Distances in Requirements Communication: a Case Study. Requirements Engineering, pp. 1-26, 2015.
5. Bjarnason E et al. A Theory of Distances in Software Engineering. Inf & Softw Tech, 2015.
6. Borg M. Embrace your Issues: Compassing the Software Engineering Landscape using Bug Reports. Proc.of 29th ACM/IEEE Int.Conf. on Automated Softw. Engin. 2014.
7. Corral L et al. DroidSense: a Mobile Tool to Analyze Software Development Processes by Measuring Team Proximity. Objects, Models, Comp., Patterns. Springer, pp. 17-33, 2012.
8. Damian D et al. The Role of Domain Knowledge and Cross-Functional Communication in Socio-Technical Coordination. IEEE ICSE 2013.
9. Eagle N, Pentland AS. Social Serendipity : Proximity Sensing and Cueing. MIT Technical report. 2004.
10. Eppler MJ, Mengis J The Concept of Information Overload: A Review of Literature from Organization Science, Accounting, Marketing, MIS, and Related Disciplines. The Information Society 20.5,pp. 325-344, 2004.
11. Erman N et al. Navigating Information Overload Caused by Automated Testing - a Clustering Approach in Multi-Branch Development, IEEE 8th Int. Conf on Software Testing, Verification and Validation (ICST) pp.1-9, 13-17, April 2015
12. Jonsson H, Nugues P Proximates–A Social Context Engine. Evolving Ambient Intelligence. Springer International Publishing, pp. 230-239, 2013,
13. Jonsson H et al. Proximity-Based Reminders using Bluetooth, IEEE Int Conf on Pervasive Comp and Comm Worksh (PERCOMW), pp.151-153, 2014.
14. Gomes LH, et al. Improving Spam Detection Based on Structural Similarity. SRUTI 5, 2005.
15. Hevner AR et al. Design Science in Information Systems Research. MIS Q. 0276-7783, vol 28, issue 1, pp. 75-105, 2004.
16. Karr-Wisniewski P, Lu Y. When More is Too Much: Operationalizing Technology Overload and Exploring its Impact on Knowledge Worker Productivity. Computers in Human Behavior 26.5, pp. 1061-1072, 2010.
17. Magnani M, Rossi L The ML-Model for Multi-Layer Social Networks. Advances in Social Networks Analysis and Mining (ASONAM), 2011 Int Conf on. IEEE, 2011.
18. Nguyen T et al. Global Software Development and Delay: Does Distance Still Matter?. IEEE Int Conf on Global Softw Eng (ICGSE 2008), 2008.
19. Salton G et al. A Vector Space Model for Automatic Indexing. Communications of the ACM 18.11 (1975): 613-620, 1975.
20. Wolf T et al. Mining Task-Based Social Networks to Explore Collaboration in Software Teams. IEEE Software, 26.1, pp. 58-66, 2009.
21. Yoo S et al. Mining Social Networks for Personalized Email Prioritization. Proc 15th ACM SIGKDD Int Conf on Knowledge discovery and data mining. 2009.