Evidence-Based Timelines for Agile Project Retrospectives – A Method Proposal

Bjarnason, Elizabeth; Regnell, Björn

Published in:
Agile Processes in Software Engineering and Extreme Programming/Lecture Notes in Business Information Processing

DOI:
10.1007/978-3-642-30350-0_13

2012

Evidence-Based Timelines for Agile Project Retrospectives – A Method Proposal

Elizabeth Bjarnason and Björn Regnell

Department of Computer Science, Lund University, Lund, Sweden
{elizabeth.bjarnason,bjorn.regnell}@cs.lth.se

Abstract. Retrospective analysis of agile projects can support identification of issues through team reflection and may enable learning and process improvements. Basing retrospectives primarily on experiences poses a risk of memory bias as people may remember events differently, which can lead to incorrect conclusions. This bias is enhanced in project retrospectives which cover a longer period compared to iteration retrospectives. To support teams in recalling accurate and joint views of projects, we propose using an evidence-based timeline with historical data as input to project retrospectives. The proposed method was developed together with a large software development company in the telecommunications domain. This paper outlines a method for visualizing an evidence-based project timeline by illustrating aspects such as business priority, iterations and test activities. Our method complements an experience-based approach by providing objective data as a starting point for reflection and aims to support objective analysis of issues and root causes.

Keywords: agile, software process, retrospective, software visualization.

1 Introduction

Continuously improving through introspection is a recognized part of agile methods and is applied in, e.g. pairing, use of automated testing and in retrospectives [2, 5, 12]. Retrospectives are commonly performed after each sprint or iterations when the development team gathers to reflect on their way of working, to identify improvements and agree on modifications for the next iteration [5, 6]. This approach aims at enabling self-governing teams to respond quickly to changes, which may require modifying how they work [6]. In addition, retrospectives may have a therapeutic effect that can further support communication and interaction within the team [3], a highly-valued aspect of agile software development.

However, there are also challenges when applying retrospectives in an agile context. Self-governing development teams tend to focus primarily on short-term issues that directly concern their team. Drury et al. found that teams that only perform iteration retrospectives, not reflecting beyond each individual cycle, tend to focus on tactical decisions rather than long-term strategic issues with the risk of losing sight of the goals of the organization [6]. In addition, it has been found that efficient coordination and communication outside of the development team, e.g. with other

C. Wohlin (Ed.): XP 2012, LNBIP 111, pp. 177–184, 2012.

dependent teams, is a challenge in particular for large-scale agile software development [9, 15]. Once projects are completed project members may be re-assigned and may quickly forget the details since accurate memory recall of project events tend to decrease with elapsed time [1, 7]. Another issue is that project-level retrospectives often require multiple viewpoints to obtain a full picture of the project since many people with different roles and focuses are involved over time.

In this paper, we propose to use evidence-based timelines to address the above challenges. The proposed method is aimed at supporting fact-based memory recall by providing a project timeline based on time-stamped data mined from various systems and databases. This method was developed together with and is planned to be evaluated at a large software development company that operates in the telecommunications domain.

The rest of this paper is organized as follows. Section 2 describes previous work on retrospectives. The research approach is described in 3, while section 4 describes our proposed method. The method is discussed in the light of related work in Section 5. Finally, we conclude and describe future research in Section 6.

2 Retrospectives

Retrospectives are prepared to enable productive face-to-face meetings, where the whole team is encouraged to share experiences and then reflect on and analyse those experiences in order to identify important issues and agree on an action plan for improvements [1, 3, 5, 13]. Retrospectives often rely primarily on the participants' experiences of what has taken place. This focus on subjective opinions may turn retrospectives into emotional venting sessions rather than being constructive fact-based discussions [3, 6]. This relates to *memory bias,* one of the barriers to learning from post-mortem reviews identified by Zedwith et al.. Memory bias is caused by the fact that what we remember is selective, and that repression of memories can override potentially important and valuable information that could have been used to learn and improve future situations [20].

Collection of both subjective experiences and objective information is included to some extent in the post-mortem process described by Collier et al.. Presenting objective data to the project team was found to enable focusing on actual problems of a sizeable magnitude, rather than merely subjective opinions [3]. In addition, objective data was found to be useful, in combination with subjective information, in supporting group analysis and identification of root causes and suitable actions [3].

Furthermore, Baird et al. observed that accurate recall of events becomes harder as time elapses [1]. The timing aspect has been reported as the main reason why project retrospectives rarely take place [7]. While 3 to 12 months after project delivery is suggested as the best time for such a retrospective, by then people tend to be tied up in new projects and a lot of the details of the previous project have been forgotten [7]. Jorgensen et al. discuss similar issues and state that project retrospectives based on subjective opinions are very likely to be biased, which in combination with simplified analysis leads to a high risk of drawing incorrect conclusions [8]. To combat this, the advice is to combine experience with knowledge (i.e. actual facts) and to use statistical techniques, in combination with being aware of the biases [8].

3 Research Approach

Our proposed method has been developed in collaboration with one of our partner companies which operates in the telecommunications domain. The company has around 4,000 employees and is faced with the challenge of developing software for a market that rapidly changes. This requires an ability to quickly adapt to change and to ensure a short time to market in order to keep up with competitors. The company wanted to evaluate their agile software development process with the goal of further improving the lead time and development efficiency. The current process assessments are performed per organizational unit and conducted on individual development teams, and not on the entire project life cycle, which includes several handovers between different teams and units. A new method was needed to assess the full development cycle from initial feature request through development in self-governing cross-functional teams to system integration and testing, and customer acceptance testing. A typical software release project contains around 200-250 new features. The total lead time from feature request until customer acceptance ranges from 9 weeks to 2 years. The feature development teams consist of 1 to 40 developers and testers. The company had three high-level goals for the new method:

i. The people involved in the full development cycle should be encouraged and motivated to learn and improve from the findings of the assessment.
ii. The assessment should take place after project completion (to get a full picture) and the effort required from participants (who have moved on to other assignments) must be reasonable to the individuals.
iii. Comparison analysis of several features needs to be facilitated, in order to identify common patterns, good practices etc., and enable organization-wide improvements.

To meet these high-level goals the concept of project longitudinal retrospectives for individual features was selected. This allows for collaborate reflection and learning for all the roles involved in the full life cycle of a feature, i.e. the main development team, and the maintenance team(s) and system-level functions (e.g. system architecture and system verification) with which the main development team interacts. In order to support comparison of multiple features, a structured and common format for the retrospectives, both in how they are prepared, performed and reported was required. For this reason, and to support memory recall and minimal preparation time for the participants, we selected to prepare a timeline with relevant time-stamped data from the available systems. These pre-prepared feature timelines *visualize* the *evidence* gathered from various systems and, thus, provide memory prompts and enable reflecting on past events without requiring much preparation of the participants. The timelines are intended to be used as the starting point for project retrospectives.

A number of meetings were held with representatives from the different units, i.e. business, software development and system verification, to discuss and review the method as it was developed. The researchers designed the proposed method, which was produced iteratively over a period of 1-2 months with regular feedback from the company. A feature timeline was produced for an example feature by extracting

time-stamped data from systems used for project- and scope- management, and for software development. Over several iterations with intermediate reviews, the *aspects* to include, the data (or *evidence)* to extract for each aspect, and how they are to be *visualized* was agreed with the company representatives. This initial desktop validation [19] was considered successful by the company representatives, and the method is at the time of writing planned to be evaluated at the company.

4 Creating Evidence-Base Timelines

Method outline. The proposed method includes four parts as input to a retrospective: goals, aspects, evidence, and visualization. *Goals* are defined for the retrospectives in order to focus on strategic improvement areas. Based on these goals, the *aspects* that are to be covered at the retrospective meetings and visualized in the timelines are then defined. The aspects are preferably selected with an eye to what data can be extracted. Both goals and aspects can be defined for continuous reflection (and, thus allow long-term comparison) or to assess issues specific for a certain project. Individual retrospectives can be aligned by defining common goals and aspects.

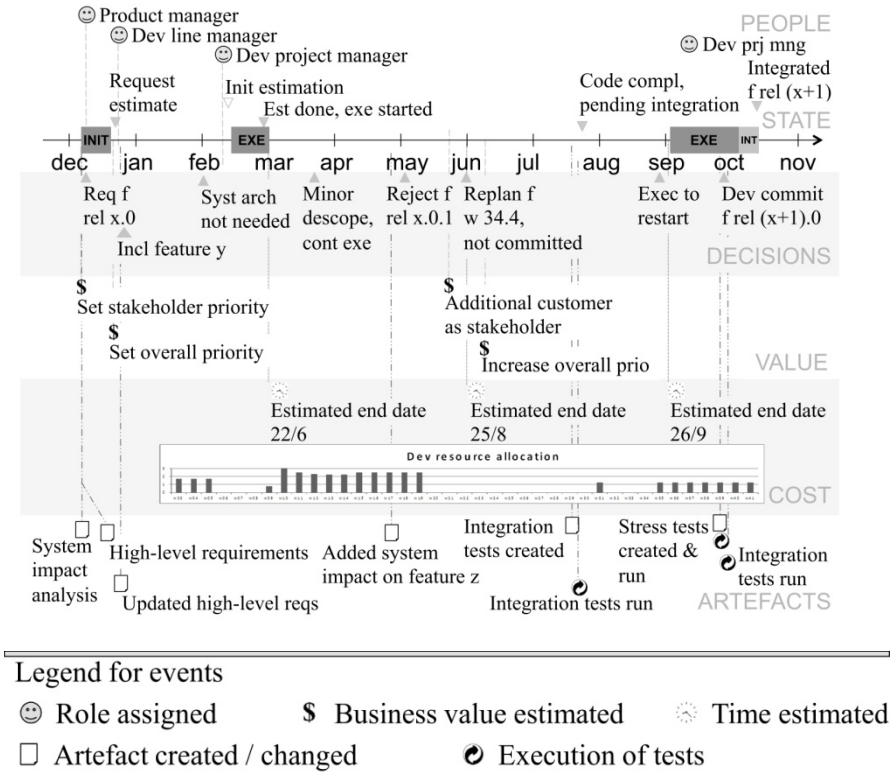

Fig. 1. An example of an evidence-based timeline for a feature

When the set of aspects to include are agreed with relevant parties, *evidence* is collected in the form of time-stamped data extracted from various available systems. The project life cycle is then *visualized* by displaying the collected evidence along a timeline. The timeline is distributed in advance to the retrospective participants together with a set of selected issue reports which form a basis for discussions at the retrospective meeting.

At the retrospective, the project history is *visualized* by posting the prepared timeline on the wall and using this as a basis for discussion and analysis. The overall timeline and the included *aspects* are first presented to orient the participants before going into detailed analysis per time period. The different aspects and relationships between them are investigated and discussed from the perspective of how they affect the issues covered by the *goals* defined for the retrospective. Missing or incorrectly shown events are elicited from participants. In addition, the participants contribute with explanations and underlying root causes for phenomena observed in the timeline. Clarifications, corrections and additional information are added to the timeline at the meeting, thus producing an updated and jointly agreed picture of the feature history as an outcome of the retrospective meeting. Over time, multiple timelines are produced using the same template, thus simplifying comparison analysis.

Timelines in context. The main retrospective *goal* for our partner company was to assess lead time with focus on communication and decisions throughout the development process. The following six *aspects* were selected to be covered by the retrospective: (1) project state (e.g. development iteration, integration, system testing), (2) decision points, (3) business value, (4) development cost (estimated and actual) and planning (e.g. estimated and actual delivery time), (5) creation and modification of specific artefacts (e.g. requirements, test cases), and (6) role assignments. *Evidence* for these aspects were gathered from various systems available at the company, e.g. databases for scope management, project planning and tracking, requirements and test cases, wiki pages, document management systems, code repository etc.. The time-stamped data was then *visualized* per aspect along a timeline, see Figure 1. For the aspects (3) – (5), namely business value, development cost and planning, and artefacts, the different events are represented by icons that illustrate the type of event, e.g. role assigned, business value or development time estimated. The evidence is grouped according to aspect, each of which is visualized in a swim lane. The aspects of project state and decision points are placed in direct proximity to the timeline axis, while the events of all other aspects are related to the timeline axis by dashed lines. This is to simplify identification of simultaneously or sequentially occurring events by displaying them in proximity to each other. For example, Figure 1 reveals that the decision taken in May to reject the feature for one release was preceded by discovery of impact on another feature (Artefacts), and followed by removing the development resources (Cost). Thereafter, an additional stakeholder was identified, the priority of the feature was increased (Value) and the execution was restarted (Decision & Cost).

For the retrospective meetings at our partner company, a similar approach to involving key roles for project history day [3] has been selected. In our case, we

decided to include roles responsible for managing the development team (product manager, project manager, software line manager, architect), and representatives from system verification and system architects. These roles may also invite other persons with specific technical competence and relevant experience, e.g. developers or testers. In all, we expect around 6-8 participants per feature excluding the moderator(s).

5 Discussion

Visualization of timelines can support more efficient processing of information and aid in identifying patterns and changes over time, and may thus stimulate memory and aid in creating a joint picture from many different perspectives [5]. All of these aspects are important objectives of retrospectives, thus making visualization of project history and evolution an interesting avenue for improving retrospective analysis and learning. Visualization of timelines has been suggested as a technique also in the field of computer forensic to enable analysis of large amounts of time-stamped data from confiscated computers [14]. In that context, the use of an interactive tool for visualizing timelines has been found to support criminal investigators in finding patterns and evidence, and to complete the task more efficiently and accurately [14]. In addition, visualization of the evolution of project data from multiple sources has been shown to be promising in understanding the relationship between multiple concerns or aspects [18], which is also part of the analysis performed at a retrospective. A different approach to visualizing the evolution of a project is investigated by Ripley et al. with the dual purpose of providing awareness of current and post-mortem events, as well as, the evolution of a project and, thus, allowing both steering a running project and learning from a completed one [16].

The purpose of our method is to stimulate a deep common understanding of issues and decisions including the underlying factors and motivations for a project. This is similar to the motivation for the project history day advocated by Collier et al. [3]. The timeline technique has been found to be beneficial in providing a joint common background and understanding of a whole project, and in supporting reflection on and observations of patterns at the project level [12]. The usage of experience-based timelines has been reported as supporting teams in reflecting on a project's process and in revealing discrepancies in interpretations of events [11]. In addition, Collier et al. found that simple timeline data gathered from three points in time supported reflecting and analysing issues concerning over- and underestimation of project cost [3]. Evidence-based timelines may act as *integrators* at the retrospective meetings and thereby, similarly to the usage of whiteboards and post-its, support creating an environment productive to constructive reflection and sharing [4].

Furthermore, using historical data has been found to support prompting memory and aiding in reflecting on project processes [10], as well as, motivating participation in deeper analysis also for team members without previous information about the full development cycle [17]. Timelines can also be useful for eliciting events with an objective approach, focusing on facts rather than opinions and have been reported to enable people to grasp different perspectives and resolve conflicts more easily [1].

However, using large amounts of data as input to retrospectives requires both filtering to avoid information overload [12], and structuring to provide focus [10]. By preparing the data beforehand saves time at the actual retrospective meeting [12], which is the case with the proposed use of evidence-based timelines.

6 Conclusions

We propose the usage of evidence-based timelines as input to agile project retrospectives. Visualization of time-stamped project data may enhance identification of patterns and problems and thereby support in-depth analysis of the project process. A deep and joint understanding of a full process can be stimulated by applying a timeline technique [5], and thus enable joint identification of problems and root causes [3, 12]. However, producing timelines requires time and effort of the participants [3]. This cost can be reduced for the participants, by, e.g. a process manager preparing the evidence-based timeline before the retrospective, and further reduced by tool support for extracting and displaying data. Examples of data that could be visualized in timelines include project schedules, problem reports, change requests, requirement and test case entities, frequency and size of source code changes. The amount of available and version controlled documentation and data limits the extent of what can be visualized in the timelines.

The project data prepared in a timeline before the meeting is complemented by gathering subjective data at the retrospective. This approach may thus enable providing a more complete and richer in-depth view of the project process by combining objective and subjective data. Furthermore, a structured collection of retrospective reports may enable organizations to more easily analyze and identify patterns between retrospectives and support improvements and learning within the whole organization [3, 4].

Finally, future work includes evaluating and further refining the proposed method in a pilot case study and investigating how to perform meta-analysis of multiple retrospectives. In addition, tool support and visualization techniques for time-stamped data are also interesting areas to pursue.

Acknowledgements. We would like to thank the practitioners involved in discussing and provided valuable input to the design of this method. The work is partially funded by the Swedish Foundation for Strategic Research.

References

1. Baird, L., Holland, P., Deacon, S.: Learning from Action: Imbedding More Learning into the Performance Fast Enough to Make a Difference. Organizational Dynamics 27(4), 19–32 (1999)
2. Beck, K.: Extreme Programming Explained. Addison-Wesley (2000)
3. Collier, B., DeMarco, T., Fearey, P.: A Defined Process for Project Postmortem Review. IEEE Software 13(4), 65–72 (1996)

4. Desouza, K.C., Dingsoyr, T., Awazu, Y.: Experiences with Conducting Project Postmortems: Reports versus Stories. Softw. Process Improve. and Pract. 10, 203–215 (2005)
5. Derby, E., Larsen, D.: Agile Retrospectives: Making Good Teams Great! Pragmatic Bookshelf (2006)
6. Drury, M., Conboy, K., Power, K.: Decision Making in Agile Development: A Focus Group Study of Decisions and Obstacles. In: Agile Conference 2011, pp. 39–47 (2011)
7. Glass, R.L.: Project Retrospectives, and Why They Never Happen. IEEE Software 19(5), 112–113 (2002)
8. Jorgensen, M., Sjoberg, D.: The Importance of NOT Learning from Experience. In: Proc. of European Software Process Improvement, Copenhagen, Denmark (2000)
9. Karlstrom, D., Runeson, P.: Combining Agile Methods with Stage-Gate Project Management. IEEE Software 22(3), 43–49 (2005)
10. Krogstie, B.: Using Project Wiki History to Reflect on the Project Process. In: Proc. of 42nd Hawaii International Conference on System Science (2009)
11. Krogstie, B., Divitini, M.: Shared Timeline and Individual Experience: Supporting Retrospective Reflection in Student Software Engineering Teams. In: 22nd Conf. on Softw. Engineering Education and Training (2009)
12. Maham, M.: Planning and Facilitating Release Retrospectives. In: Agile Conference 2008, pp. 176–180 (2008)
13. Nolan, A.J.: Learning from Success. IEEE Software 16(1), 97–105 (1999)
14. Olsson, J., Boldt, M.: Computer Forensic Timeline Visualization Tool. Digital Investigation 6, 78–87 (2009)
15. Pikkarainen, M., Haikara, J., Salo, et al.: The Impact of Agile Practices on Communication in Software Development. Empir. Softw. Eng. 13, 303–337 (2008)
16. Ripley, R.M., Sarma, A., van der Hoek, A.: A Visualization for Software Project Awareness and Evolution. In: VISSOFT 2007, pp. 137–144 (2007)
17. Sertic, H., Marzic, K., Kalafatic, Z.: A Project Retrospective Method in Telecom Software Development. In: ConTEL 2007, pp. 109–114 (2007)
18. Treude, C., Storey, M.: CONCERNLINE: A Timeline View of Co-Occurring Concerns. In: ICSE 2009, Vancouver, Canada, pp. 575–578 (2009)
19. Wohlin, C., Gustavsson, A., Höst, et al.: A Framework for Technology Introduction in Software Organizations. In: Proc. Softw. Process Improve. Conf., pp. 167–176 (1996)
20. von Zedtwitz, M.: Organizational Learning Through Post-Project Reviews in R&D. R&D Management, 21(3), 255–268 (2002)