Sprint Performance Forecasts in Agile Software Development - The Effect of Futurespectives on Team-Driven Dynamics

Conference Paper · July 2019

DOI: 10.18293/SEKE2019-224

Fabian Kortum
Leibniz Universität Hannover

Jil Klünder
Leibniz Universität Hannover

Sprint Performance Forecasts in Agile Software Development

The Effect of Futurespectives on Team-Driven Dynamics

Fabian Kortum*, Jil Klünder*, Wasja Brunotte*, and Kurt Schneider*
*Software Engineering Group
Leibniz University Hannover, Germany
Email:{fabian.kortum, jil.kluender, wasja.brunotte, kurt.schneider}@inf.uni-hannover.de

***Abstract*—In agile software development, the sprint performances and dynamics of teams often imply tendencies for the success of a project. Post mortem strategies, e.g., retrospectives help the team to report and share individually gained experiences (positives and negatives) from previous sprints, and enable them to use these experiences for future sprint planning. The interpretation of effects on sprint performance is often subjective, especially with concern to social-driven factors in teams. Involving strategies from predictive analytics in sprint retrospectives could reduce potential interpretation gaps of dynamics, and enhance the pre-knowledge, also awareness situation when preparing for the next sprint. In a case study involving 15 software projects with a total of 130 involved undergraduate students, we investigated the post-effects on team performances and behavioral-driven factors when providing predictive analytics in retrospectives. Besides measures for productivity, we consider human factors, e.g., team structures, communication, meetings and mood affects in teams as well as project success metrics. We developed a unique JIRA plugin called *ProDynamics* that collects performance information from projects and derives trend-insights for next sprints. The *ProDynamics* plugin enables the use of a times series and neural network model within a JIRA system to interpret factorial dependencies and behavioral pattern, thus to show the next sprint course of a team.**

***Index Terms*—team dynamics, human factors, data analytics, futurespectives, sprint performances, agile**

I. Introduction

In agile software development, accurate sprint estimations and development performances by teams are essential and can cause the difference whether project goals will be sufficiently completed in time, or neglected [1] [2]. However, studies with the focus on software process improvement are often motivated by increasing the overall success of projects, with the help of using modern methods or technologies that enable additional feedback [3] [4]. For organizations, sprint feedback is important and valuable because it allows the teams to receive both subjective and objective information from different perspectives, e.g., productivity monitoring systems like JIRA, customer responses, and team perceptions also experiences. The feedback mentioned above strongly relies on human factors [5]. While tool-oriented technologies and development frameworks are continuously improved, do the social-driven factors of the team present particular difficulty in the development process chain. Accordingly, the need for understanding the effects of human factors has continuously grown in relevance for the software engineering discipline, in particular for the improvement of the development process [6] [7]. Towards this, team feedback with focus on past sprints conditions can provide a substantial insight towards earlier experiences, dysfunctional habits or positive performance influences. For example, feedback through retrospective sprint characterization and visualizations benefits of post-mortem summaries, similarly to sprint reports [8]. However, dependency implications or trend highlighting concerning team behavior or performances changes over time are barely considered or hard to grasp. This leads to possible knowledge and interpretation gaps when preparing follow-up sprints because previous effects may not be fully identified or adequately encountered. It is highly desirable to give teams a more sustainable feedback opportunity that involves sprint feedback involving both, retrospective and future characterization. Subsequently, we derive the following two research questions from the above-reported context.

> *RQ_1: Do agile teams show affects for their self-organization based on previous sprint highlighting and customer satisfaction feedback?*

> *RQ_2: Can sprint performance forecasts based on retrospective data help teams to improve development performances in follow-up sprints?*

We addressed and investigated *RQ1* and *RQ2* within a case study involving 130 undergraduate students working in 15 software projects, all founded from industrial, government or public institutional partners. The teams followed a Scrum-oriented development process with support through the project management software JIRA. The case study involves weekly self-assessments resolved in JIRA. The question set covers team behavioral driven features, to gain the situational dynamics in a project with effect for the performances in sprints — satisfactory reflections of customer became additionally elicited at the end of every sprint. Productivity information, e.g., velocity during sprints, estimation gaps or sprint interventions become tracked and accessed directly by

DOI reference number: 10.18293/SEKE2019-224

JIRA. Half of the projects (seven) were granted to obtain a JIRA plugin called *ProDynamics* [8]. The plugin is based on earlier studies, enabling teams to give and receive additional retrospective feedback for a better knowledge transfer at the end of sprints [8] [9]. In this study, we extended the primarily retrospective feedback in *ProDynamics* with two predictive analytics features. Teams with access to *ProDynamics* can additionally perform time series analyzes for short-term forecasts, also sprint performance tendencies with the help of a neural network encoder-decoder model [10].

This paper is structured as follows. In Section II, we discuss related work on team behavioral effects, feedback and data analytics adaptions. Section III provides a brief overview of the study context. In Section IV, we address the methodology about self-assessments in sprints, also time series and neural network forecasts. In Section V, we describe and interpret results. Section VI concludes our research and future work.

II. Related Work

This study builds on previous results and related work with the focus on human-centered software engineering, and the relevance of fast feedback in an agile context.

Human-centered software engineering takes an import role in modern software development [5]. The focus on team communication, self-organization, and well-working relationships within teams reflect the need for better understanding of behavior-driven factors [9]. When planning sprints in agile software development, pre-knowledge on people having different personalities, skills, and ambitions within the organizational structure is crucial [7]. By means, understanding such human interdependencies enrich project improvements throughout better team performances due to a reduction of estimation gaps [11]. Besides, e.g., the meeting and communication manner or atmosphere changes over time can grant valuable project directions towards potential follow-up conflicts or misleading habits that endanger the sprint performances [12] [13]. Nevertheless, human-centered software engineering strongly depends on the contribution and self-reflection of teams, in sharing experiences from previous sprints [14].

Whenever targeting sprint performance improvements in agile development, team feedback, and change adaptions should be sincerely considered [15] [4]. In the early 2000s, research results by Rising et al. [16] subsequently revealed that team feedback during iterative development phases helps the teams most when also involving customer feedback and reflections. The customer perception forms a substantial base in the improvement process by associating group subjective sprint perceptions with the team and development expectation of stakeholder. Vetrò et al. [3] also focused on the effect of fast feedback cycles in software development. The authors observed the impact of different feedback mechanism when gathering information from software development teams directed affects for the quality and transferability of experiences and knowledge with changes in following sprints.

Retrospectives are commonly applied in Scrum and most other agile processes to share and interpret team experiences on performance effectiveness collected during the last sprint [5] [11]. Knowledge gained this way is then taken into account to estimate the next sprints more effectively according to previous outcomes. However, the interpretation of team-behavioral pattern often remains subjectively. However, computer-supported interpretation of sprint performance becomes increasingly important and visible - study results, e.g., reported by Vetrò et al. [15] show the improvement potentials in combining data analytics with traditional team feedback appraisals to improve future sprint estimations [17] [15]. The authors applied multiple-regression analyzes to characterize behavioral pattern on, e.g., meeting and development manner, to grant teams better insights on factorial influences over time, especially for efficiency hazards occurred during sprints. Our case study combines feedback mechanisms and predictive analytics about human-centered sprint performance factors.

III. Context of Comparative Case Study

This approach focuses on the effects of feedback on development performance by teams when providing additional feedback that involves forecasts and sprint tendencies. The futurespectives considered in this study base on computer-supported analyzes that relates to team and customer reflections. Predictive analytics is also integrated to characterize interdependencies of behavior pattern. We analyzed *RQ1* and *RQ2* using a comparative study design [18] by observing 130 undergraduate students working in fifteen project teams (eight to nine student per team). Each project was pre-estimated with an approximated development complexity of 2,000 working hours, equally distributed over 15 weeks. Seven of the fifteen teams used the *ProDynamics* JIRA plugin, enabling them to characterize the previous team and development performances, and to esteem next sprints with the support of times series forecasts and neural network prediction models.

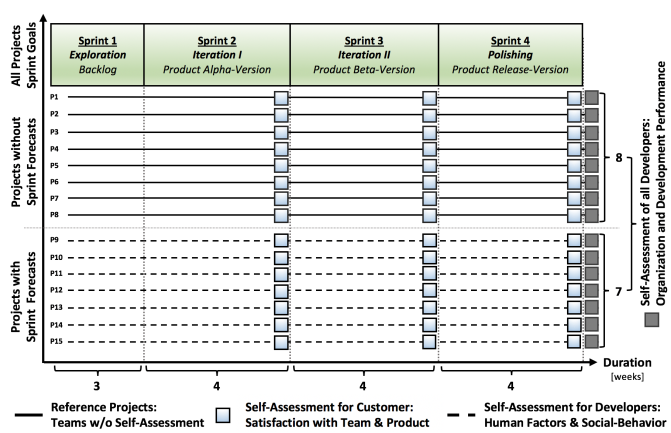

Fig. 1. Comparative Study: Projects, Sprints and Self-Assessments

Figure 1 shows a process overview of the comparative case study design. Participating in the software projects is eligible for students in the fifth semester of their computer science undergraduate studies. The main focus is to collect practical team experience following an iterative development process including agile methods and practices, such as weekly scrums and storyboards. Students do not receive gradings for participating or performances during the projects. Nevertheless, the course is compulsory within the syllabus. As mentioned before, Scrum was mandatory in each project. All teams had to self-organize inner structures, meetings, and communication, and to manage sprint tasks on a Scrum-board using the project management software JIRA.

For a primary information flow and status exchange, each team was requested to meet face-to-face for at least once a week. For product progress updates, a subsequent meeting with the customer was regularly scheduled once a week. During the sprints, change request could occur, which led to adjusted or discarded issues. At the end of each sprint, teams used retrospectives to highlight and reflect on positive and negative sprint situations. This has mostly been experiences that help the team to better estimate the next sprint tasks and organizational structures. During the projects, additional customer-reflections and satisfaction feedback were assessed to monitor group and development performances also from customer perspectives.

IV. Methodology

This methodology section starts with *(A)* the ideology of our JIRA plugin for advanced sprint retrospective and future trend support. The chapter covers the role and realization of *(B)* the self-assessments on teams and customer satisfaction that were collected as part of the case study. Furthermore, we show how the elicited team and sprint information is later used within *(C)* the time series and neural network prediction models. In the last subsection, we describe *(D)* the monitoring process for the sprint performances with a focus on *RQ1* and *RQ2*.

A. ProDynamics –Retrospectives & Futurespective Support:

The project management system JIRA is worldwide known for its team-oriented sprint planning and issue tracking support, commonly used in agile software development. Its usage is scalable from large industrial projects to small entrepreneur solutions. The standard features of JIRA provide substantial help for teams to monitor not yet completed sprints and to derive performance reports for past sprints (development velocity). Lessons learned or experiences gained during a sprint are often reflected by the teams through post-mortem retrospectives [19]. This way, dysfunctional or beneficial sprint characteristics, as well as other individually gained insights, become shared and discussed within the team, enabling them to plan the next sprint in addition to pre-experienced situations or manners. However, reasons for performance affects are not always easy to explain, in particular since the standard JIRA system only characterizes productivity statistics without further implication. We addressed this problem in enabling teams to access fast feedback through a JIRA plugin called *ProDynamics* [8]. In Fig. 2, we give an overview on the sprint analysis features currently provided in *ProDynamics*.

The red box highlights the two newly integrated sprint forecasting features of this study. The plugin addresses the ideology to support and increase the team's understanding and awareness for factorial effects on team-driven sprint behavior and development performance. Previous study results have shown that after suitable preparation of retrospective data, teams can be supported by retrospective computer-aided feedback [8] [15]. However, we believe that the support for such teams in their sprint estimations can further enrich the organizational and development performances with the help of integrated data analytics solutions.

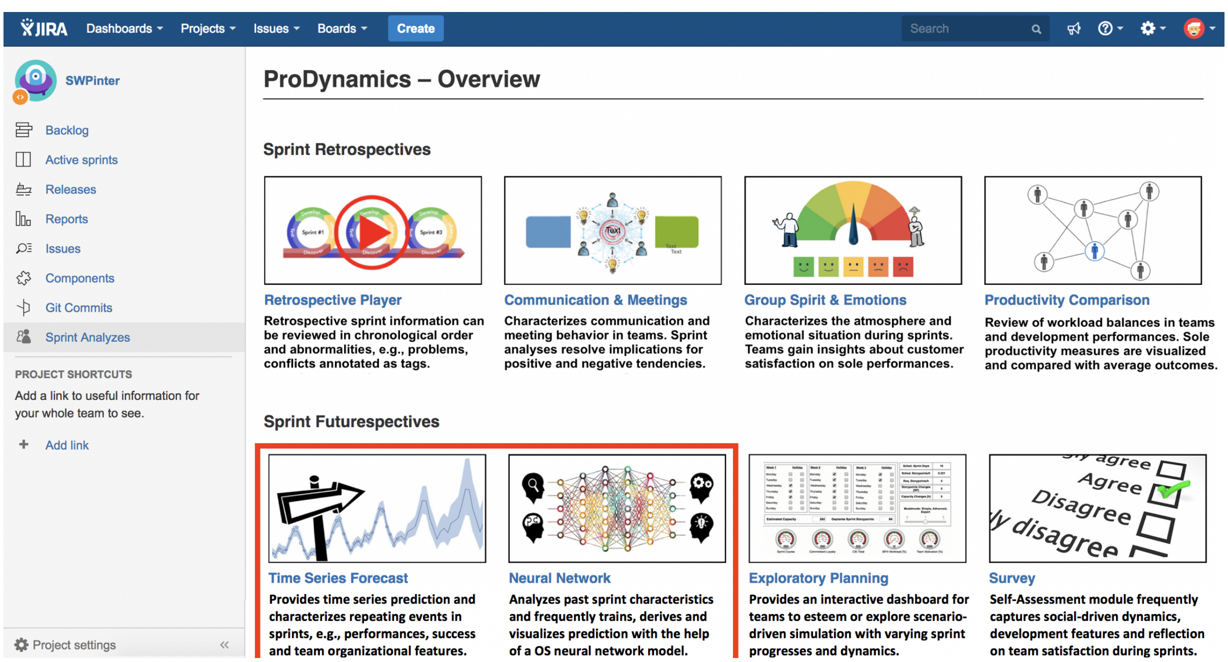

Fig. 2. Behavior-Driven Feedback Support on Sprint Futurespectives in JIRA

B. Self-Assessments for Teams, and Customer-Satisfaction:

Interpretations of previous sprint dynamics or estimations of follow-up sprints concerning the social nature of agile teams can be challenging [7]. Various human factors, such as mood, communication or meeting manner have a direct effect for the ongoing project, while dysfunctional behavior often remains undetected or hard to grasp until problems enlarge [9] [20].

Our approach is designed for teams with an open mentality for self-reflection in exchange for sustainable feedback that enables opportunities for change-driven improvements of organizational and development structures [14] [4]. Using the *ProDynamics* plugin, we enable an integrated self-assessment solution for a systematic elicitation of team dynamics in ongoing sprints. During our comparative case study with n = 15 projects, three different assessments were applied to grasp the maximal descriptive team characteristics over time. The assessment designs and question features are based on previous studies, also related work [21] [9], and continuously refined to reach the currently applied versions. The question set is self-adapting, e.g., the information elicitation about communication or meeting behavior only occur for members with active information exchange. With this, the interviewees' effort to complete a survey could be minimized to a length of 1-2 minutes. All assessments are realized through 5-points Likert scales, determining his or her level of agreement on a symmetric agree-disagree scale with predefined sprint or team behavioral statements. In the following, we explain the three assessment types in detail. A summary of the variants, in particular, the intervals, self-assessment sprint information by categories, and interviewees are shown in Fig. 3.

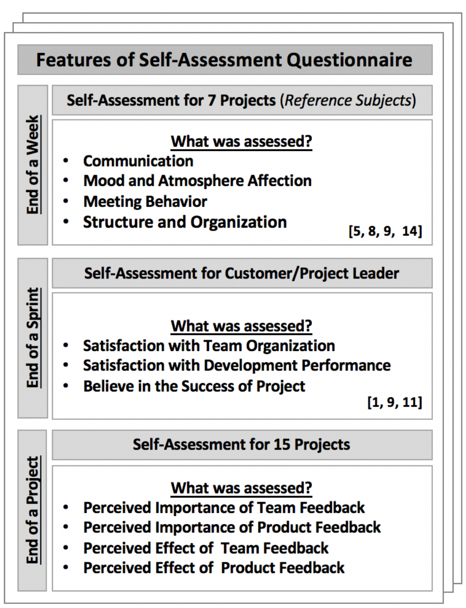

Fig. 3. Self-Assessment Intervals, Interviewees and Features

1) Self-Assessment at the End of each Week: During the 15 weeks of this case study, seven out of the fifteen student developer teams voluntarily participated in the study.

Besides the access to the *ProDynamics* retrospective and futurespective sprint features, the participation included a weekly self-assessment for each team member. The assessments capture social- and organization features, such as:

- who-to-whom communication and media channel use
- meeting quantity and average duration
- the atmosphere in the group, personal mood
- satisfaction about the last weeks' performances

We derived other team and sprint metrics, e.g., meeting participation, maverick trends, and centralized communication structures from each team member's response. A summary of all assessment information is shown in Fig. 3. Subsequently, all category features that are currently considered by the time series and neural network model, including productivity measures from JIRA, become listed in Fig. 5.

2) Self-Assessment at the End of each Sprint: The second self-assessment question set on the satisfaction with the team and development performances for all 15 software projects was also answered by the customer, and the scrum master at the end of each sprint (except the first). We activated the performance assessments with the second sprint, because the first three weeks of the project were mainly for exploration, to create and fill the backlog, form team structures, get to know the customer and reach a steady state before the next sprint.

However, the survey covers in total ten questions, four about the team organization, four on the development performance as well as two items focusing on the overall satisfaction with the team and product. The question structure became realized through 5-points Likert scales, determining the customers and scrum master's level of agreement on a symmetric agree-disagree scale, similar to the other two self-assessments. The question set was used as one reference indicator between team-driven dynamics during the sprints and potential effects for the customers' satisfaction. The scrum masters' responses become utilized to determine possible offsets between external views and the team inside knowledge. For the comparison of satisfaction changes between the teams with access to *ProDynamics* and those that did not, the customer and scrum master of all fifteen projects were invited to complete this self-assessment form.

3) Self-Assessment at the End of each Project: The third self-assessment took place only once at the end of each project and became applied to all 130-student developer. The survey includes questions on the personal perception of the developers, e.g., whether there were moments with a need for additional feedback during the sprints, and if there were recognizable effects (positive or negative) within the own team in case of provided feedback. The following four assessment features became only ones elicit at the end of each project for the validation of every student's perception of their team and development performances during the 15 weeks:

- importance of organizational feedback for the team
- importance of product feedback for the development
- perceived effect in the team because of team feedback
- perceived effect in the team because of product feedback

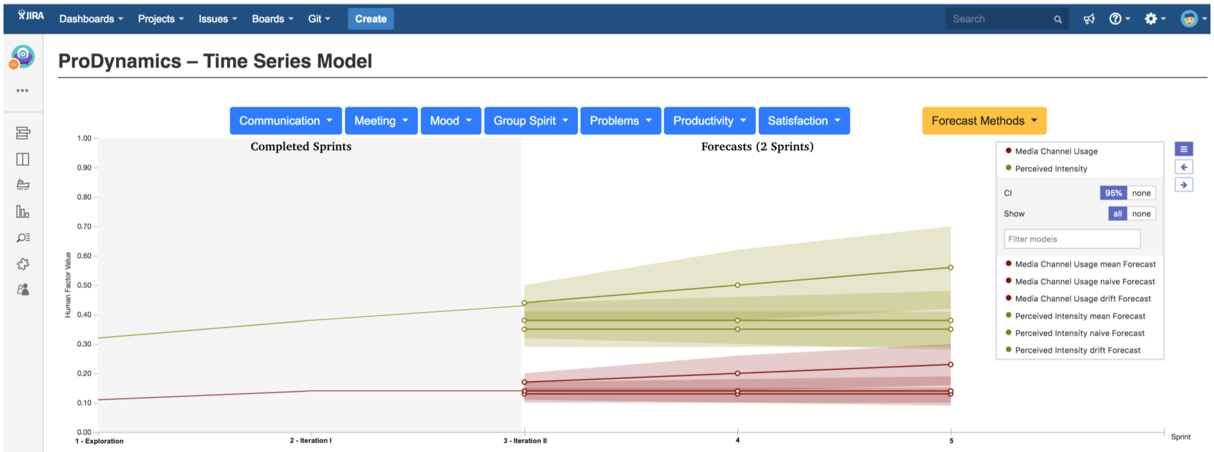

Fig. 4. Time Series Sprint Forecasts derived in *ProDynamics*

The responses were compared with the customer and scrum master satisfaction after each sprint. This survey also involves questions about the need and usefulness, e.g., of a centralized feedback solution in JIRA. Also, whether JIRA was an adequate solution to manage and organize sprints during the case study project. Those are by-product information, in particular with no relevance to answer *RQ1* and *RQ2*.

C. Sprint-Trends through Time Series and Neural Networks

With this case study, we investigated whether teams can gain a more sustainable use of feedback when considering both, past retrospective records and future sprint performance predictions. Besides this, we incorporated retrospective sprint characteristics (e.g., organization structures, communication- and meeting behavior, productivities, motivation) with two predictive models to grant teams an additional trend characterization on behavior-driven factors when esteeming the next sprints. We chose both predictive methods based on their functional properties in supporting inference-statistical analysis, e.g., sprint series. The analysis helps to estimate sprint and team measurements in the future based on past team-behavior in sprint sequences.

The times series forecast used for the sprint-trend esteems based on an open source java library published by Workday[1]. The library provides time series analyzes, involving ARIMA-, Mean-, Naïve- and Drift-forecasts. The forecasting model in this study become fitted by the measurements listed in Fig. 4.

The ARIMA-model characterizes seasonal inferences from past and forecasts future points in the series [10]. The Mean-method derives the arithmetic mean from the sequence of sprint data to esteem follow-up values within fitted parameter ranges. The seasonal Naïve-method uses sprint metrics from the second sprint week on to predict the third sprint parameters. The prediction of the fourth sprint metrics is derived with the values from the third sprint, and so on. The Drift-method obtains a straight line between the first and last data point to characterize the sprint metrics tendency drift. Seasonal patterns are not taken into account with this Drift-method. Figure 4 shows a time series forecast for the communication metrics Media Channel Usage and Perceived Intensity.

The time series viewer enables the teams to choose from one to four supported forecasting methods. The interactive chart allows a user to select different past sprints and the underlying metrics. With this, the chosen sprint metric(s) become analyzed and forecast with the help of the four forecasting methods. The prediction interval can reach a maximal length smaller than the number of yet completed sprints. The colored lines within the gray background area shown in Fig. 4 present the real data points for two selected communication parameters. The colored areas on the right half of the chart mark the 95% confidence interval of the prediction of each forecast. For example, the maximally available forecast horizon in Fig. 4 is two sprints, because the time series model derives its prediction based on three yet completed sprints. Sprint forecasts are labeled on the time axis through a counter and completed sprints name tags.

The *ProDynamics* – Neural Network viewer focuses on the second prediction approach for estimating social-driven team measurements based on retrospective sprint and team records. The neural network is implemented in using the open source library Deeplearning4j. The Deeplearning4j-library covers cross-platform algorithms on machine learning and artificial intelligence is implemented in Java and runs in a JVM such as used by the JIRA system. Similar to our time series forecast solution the neural network viewer provides past sprint metric plots based on the user selections.

Also, the neural network model allows the team to review past sprint conditions covering organizational structures, social-driven team behavior, customer satisfaction, productivity measures as well as problem and conflict appearances. An overview of all covered factors is listed in Fig. 5, which is a result of the assessed team and customer responses as well of development performances that is natively tracked in JIRA. The model is trained with the listed data features considering the availability of already completed sprint records.

[1] https://github.com/Workday/timeseries-forecast

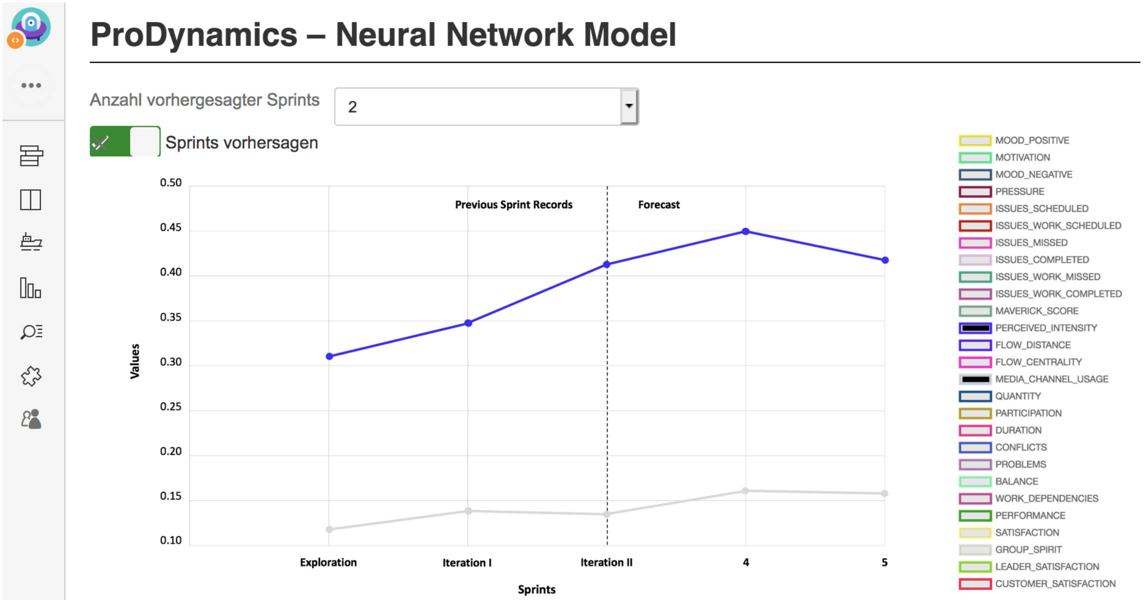

Fig. 5. Neural Network Sprint Predictions derived in *ProDynamics*

Training the model requires enlarged computation effort for the data encoding process. Therefore, model updates are performed automatically, but only once during the weekend. The neural network viewer enables the teams to choose individual real data plots of past sprints, or with an active future horizon. The maximal size of future-horizon is limited to the number of yet completed sprints minus one, similar to the time series model. The interactive chart allows a user to select between the sprint metrics and plot trends for the considered team-driven factors. The colored lines in front of the dash border line present the real data points for the two chosen communication parameter Media Channel Usage and Perceived Intensity. The colored lines on the left side of the dashed border present the computed prediction according to the feature records encoding of three yet completed sprints.

D. Sprint Performance Monitoring

In this study, the sprint performances of teams' base on the development (velocity of tasks) as well on the organizational performances during each sprint. The velocities in all teams are comparable productivity measures tracked within JIRA, thus did not require to be separately assessed. This enabled us a direct performance comparison was between a particular group and the average of all other teams during a sprint. Besides the sole productivity measures of teams, we used the customer and scrum master satisfaction feedback after each sprint to determine whether the development outcome also fulfilled product expectations according to quality and functional requirements.

However, the team performances over time with concern on social-driven changes due to futurespective feedback became solely traced through the customer, and scrum master feedback elicit at the end of every sprint. We are reasoning this processing with the fact, that only half of all teams could access the *ProDynamics* futurespectives, and also frequently completed the self-assessments on social-driven behaviors. In considering the customer and scrum master perceived team performances during each sprint, we could compare the organization performance changes of all teams. Subsequently, the effects and trends could become characterized due to the comparative study subjects with different sprint estimation and planning support. Of course, at this appraisal level, sole factorial influences become not closer taken into account. However, it allowed us first interpretations about whether teams adopt the *ProDynamics* usage, also whether groups with access get used to derive better sprint estimations with constant or even positively improving customer satisfaction outcome, on organizational and product aspects.

V. Interpretation and Validity of Results

In the following two subsections, we statistically interpret and discuss the effects of *ProDynamics* futurespectives on the sprint performances, also emphasize related threats to validity.

A. Interpretation of statistical measures

In Section IV, we described the sprint performances as a result of sprint productivity (velocity), also the customer and scrum master satisfaction responses on the team and product performances after every sprint. With the help of Pearson correlation analysis and 2-paired t-Tests, we determined sprint performance differences between the seven groups that actively used the *ProDynamics* prediction features and the other eight teams without access. We found out that the teams with access to the provided sprint forecasts showed fewer estimation errors, therefore with more optimal velocity distribution at 98% as the comparison of the orange boxes *(1)* in Fig. 6 and Fig. 7 reveal. The overall sprint estimation error for those teams was ±9 %. In particular, do the groups

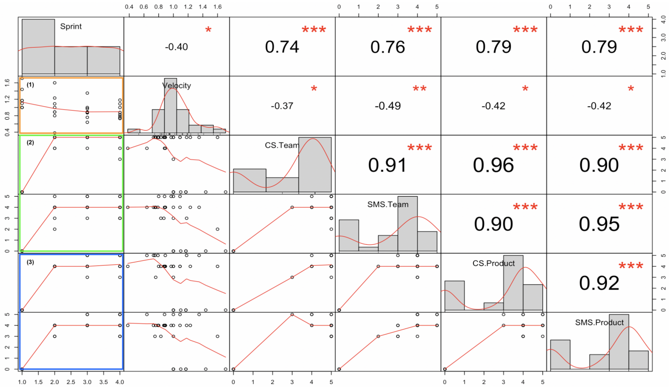

Fig. 6. Team Projects without *ProDynamics*-Futurespective access

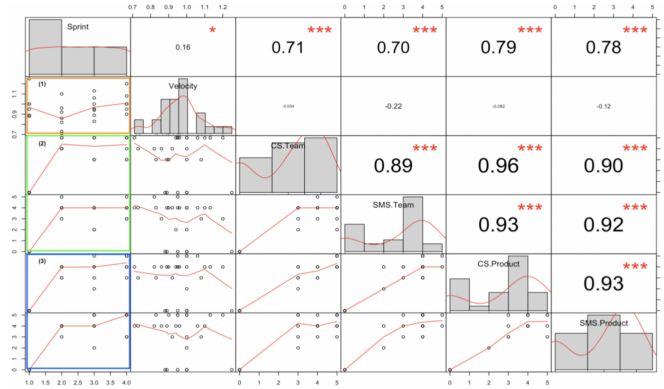

Fig. 7. Team Projects with *ProDynamics*-Futurespective access

without access to *ProDynamics* showed strong tendencies for over-estimating their sprint tasks during the first two sprints, followed by underestimations that caused strong deviations between the number of scheduled tasks and completed ones. Due to this, an overall sprint estimation error of ± 19 % was identified. However, in comparing the yellow markings *(2)* in both figures, the team's organizational performances revealed no benefits given due to the additionally provided sprint forecasts. Moreover, the outcome is on a constant level, while both, customer and scrum master reflected a sustainable organization and communication structure in most teams.

The third factor of the sprint performances involves the software product, in particular, the quality, and requirements fulfillment after each sprint. The futurespective in *ProDynamics* enabled a few teams to preview customer satisfaction according to previous performances. However, the forecasts only highlight chances for adjustment, while the teams decide whether to use the available information to improve the previous situation. By comparing the blue boxes *(3)* in Fig. 6 and Fig. 7, an affect for teams with *ProDynamics* usages becomes reflected throughout an increasing product satisfaction by the customer and scrum master. The satisfaction increase can be of course also because of excitement about the product majority.

Nevertheless, a significant rise in teams with customer feedback knowledge could be measured, while comparable projects remained on a constant level. For redundancy, we also considered the product satisfaction of scrum master, which significantly correlates with our interpretation. Beside results of this case study showed strong accordance between customer and scrum master perceived team and development performances. While the scrum master usually tends to have more team internal information and critical knowledge about accomplishments, does the deviation with external customer ratings present a low perception gap.

B. Threats to validity

Construct validity: We looked at the social-driven team affects only through statistical and artificial methods. The sprint information features obtained in *ProDynamics* are chosen based on previous studies [18] [13]. However, the accuracy of the predictions strongly depends on the quality and completeness of the self-assessed data. The effects on performances depend on whether the teams considered the sprint forecasts in follow-up esteems. Customer satisfaction responses as one success indicator might have involved offsets, because of unjustified expectation or lack of experience.

Internal validity: The interpretations and results of this study rely on the self-assessed team and customer feedback. Only voluntary groups completed surveys, in exchange for accessing additional prediction support in JIRA. Therefore, team responses can be assumed to be accurate and unbiased [16]. Participating teams accepted the privacy limitation, e.g., communications and performances were viewable, but exclusively by the assigned team.

External validity: Since we examine student teams, the results should not be overgeneralized. However, the software projects are founded by a real customer from industry, public institutions or governments. Hence, data collected from other company or university projects could lead to different results. Due to further limitations like the involvement of social-driven factors and unknown domain influences, our interpretations may not embrace all possibly existing project scenarios.

Conclusion validity: All interpretations are plausible and statistically valid. However, there may be different self-assessment responses when repeating the project with the same participants: team-behavioral factors, emotions, skills or unknown influences could have changed in the meantime. Subsequently, the accuracy of predictions could vary due to differently completed surveys, and project progresses. However, the methodology can be generalized and applied to various agile projects that allow assessments on team and sprint information.

VI. Conclusion

This study focuses on the effects of social-driven dynamics in agile software developments when providing teams additional feedback on sprint tendencies. To determine the impact of futurespective feedback, we realized a JIRA plugin named *ProDynamics* that simplifies the elicitation of team-driven factors as well as performance measures within JIRA.

The feedback becomes resolved through sprint series forecasts and neural network predictions in an integrated JIRA plugin solution. The computer-based sprint analyzes use team and customer reflections that were frequently assessed, analyzed and characterized for factorial interdependencies on development performances and team-driven behavioral pattern.

With the help of a comparative case study involving fifteen software projects with 130 students, throughout 15 weeks, we gathered weekly information about communication, meeting and emotional metrics from half of the projects. The elicit data became frequently used to train time series and neural network models, enabled the 7 out of 15 groups to gain additional insights about previous sprint and team performances, also derive trend-forecasts for follow-up sprints. Measuring the performance differences between the groups with pro-active feedback and those without involved customer and scrum master feedback from all 15 projects, that became repeatedly elicit at the end of every sprint. The feedback covered past sprint perceiving on both, team and development performances.

Pearson correlation statistics helped us to interpret the effects on sprint performance, in particular, the team and development performance in each project. We found statistical evidence towards that the groups with access to the additionally provided *ProDynamics* forecasts showed a definite decrease for sprint estimation gaps by 10%, while the groups without *ProDynamics* access tend to have more volatile velocity performances. We could show, that the additional use of forecasting methods supports the groups to interpret customer satisfaction better, thus improve the product outcome at the end of sprint. The study also revealed, that teams not necessarily adjust internal organization structures due to predictive information. Most of the groups showed an almost steady level in their weekly communication and meeting behavior, towards no significant affects could be determined.

We can conclude that the *ProDynamics* futurespectives enabled a sustainable team organizational and development performance improvement for the groups with access to the plugin. The team performances dynamics during the sprint sequences showed strong stabilizing characteristics, due to more accurate sprint esteems compared with the comparison groups that only used general sprint information in JIRA, e.g., burndown- and velocity charts. Besides, the *ProDynamics* plugin realized a simplified data elicitation for social-driven team factors, while some group had could reach a positive effect for follow-up sprint executions.

We are currently working on a newer version for the *ProDynamics* plugin, that does extend the retrospective, and futurespective sprint analyzes, by a planning-oriented simulation feature. With this, we believe that teams can potentially improve sprint estimations because of training effects. Various scenarios could be explored before an official sprint start, by incorporating past performances with a generalized simulation model for agile development processes, e.g., system dynamics. A simulation-based approach could grant the team a better insight about appearing behavioral dynamics over time, also help to discover new characteristics, that would remain undetermined otherwise. Generally spoken, simulations could gain further knowledge and train the sprint estimation skills of teams and project manager.

Acknowledgment

This work was funded by the German Research Society (DFG) under the project name Team Dynamics (2018-2020). Grant number 263807701.

References

[1] N. Agarwal and U. Rathod, "Defining 'success' for software projects: An exploratory revelation," *International journal of project management*, vol. 24, no. 4, pp. 358–370, 2006.

[2] P. Mohagheghi and M. Jørgensen, "What contributes to the success of it projects? an empirical study of it projects in the norwegian public sector." *JSW*, vol. 12, no. 9, pp. 751–758, 2017.

[3] A. Vetrò, S. Ognawala, D. M. Fernández, and S. Wagner, "Fast feedback cycles in empirical software engineering research," in *Proceedings of the 37th International Conference on Software Engineering-Volume 2*. IEEE Press, 2015, pp. 583–586.

[4] L. Williams and A. Cockburn, "Agile software development: it's about feedback and change," *IEEE Computer*, vol. 36, no. 6, pp. 39–43, 2003.

[5] A. Cockburn and J. Highsmith, "Agile software development: The people factor," *Computer*, no. 11, pp. 131–133, 2001.

[6] F. Kortum, J. Klünder, and K. Schneider, "Characterizing relationships for system dynamics models supported by exploratory data analysis," in *29th International Conference on Software Engineering and Knowledge Engineering. KSI Research Inc*, vol. 15, 2017, pp. 39–43.

[7] E. Whitworth and R. Biddle, "The social nature of agile teams," in *Agile 2007 (AGILE 2007)*. IEEE, 2007, pp. 26–36.

[8] F. Kortum, J. Klünder, and K. Schneider, "Behavior-driven dynamics in agile development: The effect of fast feedback on teams," in *Proceedings of the 2019 International Conference on Software and System Process*. ACM, 2019.

[9] K. Schneider, O. Liskin, H. Paulsen, and S. Kauffeld, "Media, mood, and meetings: Related to project success?" *ACM Transactions on Computing Education (TOCE)*, vol. 15, no. 4, p. 21, 2015.

[10] G. P. Zhang, "Time series forecasting using a hybrid arima and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, 2003.

[11] N. B. Moe and T. Dingsøyr, "Scrum and team effectiveness: Theory and practice," in *International Conference on Agile Processes and Extreme Programming in Software Engineering*. Springer, 2008, pp. 11–20.

[12] S. Kauffeld and N. Lehmann-Willenbrock, "Meetings matter: Effects of team meetings on team and organizational success," *Small Group Research*, vol. 43, no. 2, pp. 130–158, 2012.

[13] J. Klünder, K. Schneider, F. Kortum, J. Straube, L. Handke, and S. Kauffeld, "Communication in teams-an expression of social conflicts," in *Human-Centered and Error-Resilient Systems Development*. Springer, 2016, pp. 111–129.

[14] C. J. Stettina and W. Heijstek, "Five agile factors: Helping self-management to self-reflect," in *European Conference on Software Process Improvement*. Springer, 2011, pp. 84–96.

[15] A. Vetro, R. Dürre, M. Conoscenti, D. M. Fernández, and M. Jørgensen, "Combining data analytics with team feedback to improve the estimation process in agile software development," *Foundations of Computing and Decision Sciences*, vol. 43, no. 4, pp. 305–334, 2018.

[16] L. Rising and N. S. Janoff, "The scrum software development process for small teams," *IEEE software*, vol. 17, no. 4, pp. 26–32, 2000.

[17] F. A. Batarseh and A. J. Gonzalez, "Predicting failures in agile software development through data analytics," *Software Quality Journal*, vol. 26, no. 1, pp. 49–66, 2018.

[18] T. Dybå and T. Dingsøyr, "Empirical studies of agile software development: A systematic review," *Information and software technology*, vol. 50, no. 9-10, pp. 833–859, 2008.

[19] J. Highsmith and A. Cockburn, "Agile software development: The business of innovation," *Computer*, vol. 34, no. 9, pp. 120–127, 2001.

[20] K. Schneider, J. Klünder, F. Kortum, L. Handke, J. Straube, and S. Kauffeld, "Positive affect through interactions in meetings: The role of proactive and supportive statements," *Journal of Systems and Software*, vol. 143, pp. 59–70, 2018.

[21] J. A. Ross, "The reliability, validity, and utility of self-assessment," 2006.