

Understanding the Role of Reporting in Work Item Tracking Systems for Software Development: An Industrial Case Study

Pavneet Singh Kochhar¹, Stanislaw Swierc², Trevor Carnahan²,
Hitesh Sajani², and Meiyappan Nagappan³

¹Singapore Management University, Singapore

²Microsoft Corporation, USA

³University of Waterloo, Canada

kochharps.2012@smu.edu.sg, stansw, trevorc, hitsaj@microsoft.com, mei.nagappan@uwaterloo.ca

Abstract—Work item tracking systems such as Visual Studio Team Services, JIRA, BugZilla and GitHub issue tracker are widely used by software engineers. These systems are used to track work items such as features, user stories, bugs, plan sprints, distribute tasks across the team and prioritize the team's work. Such systems can help teams track the progress and manage the shipping of software. While these tracking systems give data about different work items in tabular format, using a reporting tool on top of them can help teams visualize the data related to their projects such as how many bugs are open and closed and which work items are assigned to a team member. While tools like Visual Studio and JIRA provide reporting services, it is important to understand how users leverage them in their projects to help improve the reporting services.

In this study, we conduct an empirical investigation on the usage of Analytics Service - a reporting service provided by Visual Studio Team Services (VSTS) to build dashboards and reports out of their work item tracking data. In particular, we want to understand why and how users interact with Analytics Service and what are the outcomes and business decisions taken by stakeholders from reports built using Analytics Service. We perform semi-structured interviews and survey with users of Analytics Service to understand usage and challenges. Our report on qualitative and quantitative analysis can help organizations and engineers building similar tools or services.

I. INTRODUCTION

Every enterprise-scale software goes through different phases in its lifetime: requirements gathering, analysis and design, development, testing, deployment, and maintenance. Some of these phases can be split and organized in an iterative process allowing customers to see and give feedback regarding partial deliverables, which effectively drives up the quality of the software to their satisfaction. These tasks and processes are together called Application Lifecycle Management (ALM) [1], [2].

Several specialized software suites such as VSTS, HelixALM etc. available in the market aim to support ALM. Work Item Tracking (WIT) system is one of them as every project requires some sort of coordination to make sure it is progressing towards its goal [3]. WIT keeps track of work items, which represent tasks that must be completed as part of the project. Work items may have additional information such as the date when it is due to be completed and information

about responsible individuals. A work item may represent a bug in a piece of software that needs to be fixed.

WIT system is useful, particularly from the perspective of individual contributors, that can help them prioritize and decide on what to work next. Upon completion, they can update the status in WIT that can be used to notify others who might be dependent on the deliverables as well as keep it for future use. Stakeholders, on the other hand, can access the system to create new work items or to observe the status, in which case they are typically interested in the summary status. To understand the overall health and progress of a project, WIT systems can be enhanced with reporting solutions that handle data aggregation and presentation. These reporting solutions can come in the form of extensions or in-product integrations.

There are many ways one can use a reporting solution for a WIT system. ALM product vendors publish lists of scenarios and questions that can be answered with the pre-defined charts and data models they provide. In this study, we would like to get a better understanding about which of these reporting capabilities are really used in practice and what are the actual benefits people get out of them as these insights will help to improve these tools and help developers better manage their projects. Given that such reporting tools are developed and deployed at Microsoft and that we can directly contact developers using them on a daily basis, we had the opportunity to understand why and how such reporting tools are used in practice as well as what are the benefits teams get out of them. Our study focuses on users of the Analytics Service, which provides data for all the reporting capabilities of the Visual Studio Team Services (VSTS) suite, a standard ALM solution used at Microsoft. We refer to Analytics Service as AX throughout the paper. AX is used by teams in Microsoft as well as by external users. In this study we aim to answer the following high-level research questions:

RQ1: Why are customers using Analytics Service?

RQ2: How are customers using Analytics Service?

RQ3: What are the learnings and outcomes of using Analytics Service?

Our study comprises a number of investigative techniques. We interviewed and surveyed developers from a diverse group

of products. We also examined activity logs for internal users to filter out respondents to be considered for the survey.

We expect the empirical result that we present here to highlight important aspects of reporting solution for WIT systems, specifically, what are the use cases of reports, who creates and consumes reports, what are the benefits that they provide already and what features should be added for even greater impact on the project. Finally, we hope that the results will be generalizable to ALM suites other than the one used in this study.

II. ANALYTICS SERVICE

A. Visual Studio Team Services (VSTS)

VSTS is a popular, extensible Application Lifecycle Management (ALM) product from Microsoft Corporation that makes it easier for software engineering teams to collaborate on code, build, test, work items, releases, packages and other team artifacts. The cohesive VSTS engineering experience and co-location of data makes it an ideal data source for comprehensive analytics that assist software teams to maximize velocity, quality, and predictability in their projects.

B. Visual Studio Team Services Analytics

The VSTS Analytics Extension, currently in preview, begins to deliver new data experiences backed by the recently released AX. The extension has been installed on over 3.5k accounts, has over 12k monthly active users, and has been experiencing a rapid 25% month-over-month growth since its release in May 2017. AX provides a curated OData v4 [4] model over the VSTS suite, beginning with WIT data, used to answer quantitative questions regarding past or present state and velocity of projects managed with VSTS. It is built upon a near real-time reporting warehouse hosted in the cloud and optimized for fast read-access and support of server-based aggregations. Figure 1 shows the overview of Analytics Service.

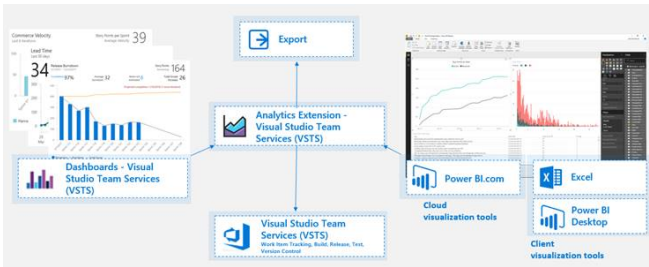


Fig. 1: Overview of Analytics Service

III. METHODOLOGY

In this section, we describe our methodology that is divided into two parts: interviews and survey.

A. Interviews

Protocol. For interviews, we follow an approach based on Grounded Theory [5] to explore how users use AX, the insights they are trying to get from the reports, challenges and the outcomes of using them. We interviewed developers who

are active users of AX. We emailed these users and invited them for the interview. Upon acceptance, we either scheduled meetings or video conferences.

We conducted semi-structured interviews with each user. The interviews were divided into two parts. In the first part, we asked a few questions related to demographics such as total work experience, which team and product they are working on and their area of discipline such as software engineering, program management, data science etc. In the second part, we asked questions related to AX usage. The high level questions that we asked included:

- 1) *Why do you use AX?*
- 2) *What is a typical scenario that you use AX for?*
- 3) *What do you get out of using AX?*
- 4) *Where and how often do you use AX?*
- 5) *How did you learn about AX?*
- 6) *What tools do you use to create and view reports?*

The questions were kept open to prevent us from having any bias on the response. The users were encouraged to give more details about their usage of AX or anything we did not ask them. Based on their initial responses, we sometimes asked them to elaborate on some of the topics they mentioned. We did not ask the questions that were answered directly by an interviewee as a byproduct of a previous question. Before concluding the interviews, we asked the users if they want to add anything else. Each interview lasted approximately 30 minutes and the audio was recorded. All the interviews were later transcribed.

Participants. In this study, we interviewed 11 AX users selected based on their usage patterns. We looked at how often they use AX to fetch data. To get a breadth of responses, we also targeted users who are working on different projects, teams and include both users working in Microsoft and non-Microsoft (external) users. Table I shows the participants we interviewed and their corresponding role.

TABLE I: Interview participants along with their current position

Interviewee ID	Role
P1	Program Manager
P2	Senior Security Analyst
P3	Software Engineer
P4	Senior Software Engineer
P5	Software Engineer
P6	Data Engineer
P7	Data Scientist
P8	Consultant
P9	Applications Development Manager
P10	Business Intelligence Manager
P11	Software Engineer

Data Analysis. After all the interviews were complete, we transcribed the recordings. For each transcribed interview, we generated 12 to 28 cards that contain the key points of the interview. Each card contained a quote from the interview mentioning a transcribed key point. In the end, we had a total of 206 cards. Next, we performed card sorting to categorize the cards based on related themes, similar to previous studies [6], [7]. We organized the card sort activity in two phases. In phase 1, we sorted the cards into high-level themes i.e., each card

falls into only one theme. In phase 2, we ensured that each card best fits the theme. In the end, we also had definitions for each theme and noted down example statements to represent each theme. These themes were not pre-defined but were chosen during the card sort.

B. Survey

Protocol. Based on the information from the card sort, we created a survey to validate our initial observations and further understand how users interact with AX, what insights they are trying to extract, the challenges they face and the outcomes they get. Our survey aimed to quantify the responses from the user interviews.

Kitchenham and Pfeelger discuss the process to design a survey, setting objectives, types of survey design, developing the survey instrument such as question types, answer types, questionnaire format, questionnaire length, evaluating and analyzing the survey data [8]. They suggest using answers such as numerical values (e.g., age), response categories (e.g., check boxes), yes/no answers, ordinal scales (e.g., Likert scale) and short free-form answer. Keeping these in mind, we design our survey that consists of questions with answers of types: numerical values, response categories, Likert scale and short free-form text. We keep the questions optional to make sure respondents are not compelled to fill the survey. We also asked several demographic questions such as total experience at Microsoft prior to diving deeper into more specific questions related to AX. The survey was anonymous to prevent any biases.

We piloted our survey on 5 of our colleagues to get inputs on the questions and their corresponding answers to determine if any parts of the survey were hard to understand. As these pilot respondents have good knowledge of the product, we also wanted to make sure we are asking the right questions as the quality of responses is based on the questions that are asked [9]. We did several iterations of the survey and rephrased the questions to make them easier to understand and removed questions that were considered too similar. An example of such changes would be replacing “When” by “How often” and providing examples for answers such as current state (e.g., no. of bugs as of today). We also updated to the survey to make it shorter as long surveys may deter participants from completing it [10]. Our goal was to ensure that the survey takes around 10-15 minutes. The responses from the pilot survey were used solely to improve the questions and these responses were not included in the final results. We kept the survey anonymous as this increases the response rate [11] and leads to respondents giving candid responses. Past study showed that giving incentives increases the chances of participation [12]. Therefore, we gave an option for respondents to enter a raffle for two \$50 gift cards.

Participants. To select respondents for our survey, we first sorted AX users by two criteria: the total number of queries and the number of distinct queries written during the month of July 2017 (the month previous to the survey date). The total number

of queries is the cumulative of all the queries associated with a particular user whereas the number of distinct queries represents only the unique queries for that user. We then found an intersection of these two lists and removed the participants whom we had interviewed in the previous step. In the end, we selected top 500 users as our survey respondents. We immediately received 43 out-of-office responses. Within two weeks, we received 99 completed responses with a response rate of 21.7%. The range of professional experience was from zero to 42 years with a median of 11 and mean of 12.3 years. The median and mean for years of experience working at Microsoft were 5 and 7.5 years respectively. These responders held positions from different disciplines: 41.8% software engineering, 36.3% program management, 9.9% service engineering, 7.7% data science and 4.4% others.

Data Analysis. We plot graphs to quantify various statements we got from the interviews.

IV. FINDINGS

In this section, we describe findings which answer each of our research questions.

A. RQ1: Why are customers using AX?

AX provides several capabilities for users to monitor the status and health of their project. During interviews, we find that people use it differently based on the needs and requirements of their projects. The main use cases of AX that we found in our study are listed below and summarized in Figure 2. The values do not add to 100% as users can specify multiple use cases.

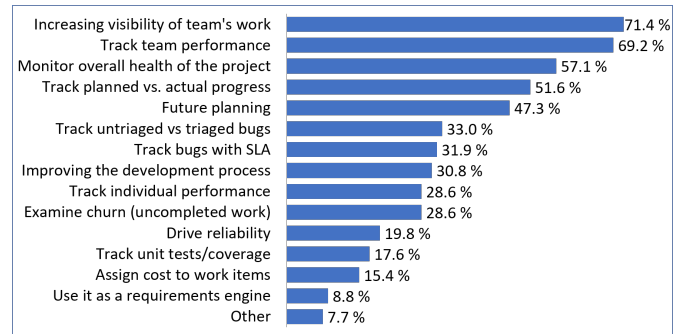


Fig. 2: Use cases of reports created using AX

1) **Increasing visibility of team's work::** AX provides features that make it easier for teams to track their progress and increase the visibility of what they are working on. **71.4%** of the responders express using AX to increase the visibility of their team's work. As mentioned in one of the surveys: “We didn't have a great visibility into all the work we were doing so the Team Services as a whole really helped with it” (P7).

Teams can use reports generated with AX for “exposing things to the management teams, leadership teams show them your people are doing this much work for driving your priorities.... This gives managers the ability to reprioritize their resources in a meaningful way... There is no guesswork... we have the data that logs the work we were doing” (P8).

2) **Track team performance::** Tracking performance of a team is one of the most important parts of any software project as it helps with delivering the product on time by showing what is the workload of different team members etc. P9 expresses, *"we use the VSTS metrics to record data, to generate some metrics around team performance, project performance...to know how the project is going to know how the team is doing on the project or series of projects."* **69.2%** of the respondents mention using AX for tracking team performance.

3) **Monitor overall health::** Teams use AX to monitor the overall health of the project i.e., to ensure items are tagged correctly, assigned to the right people, put under correct iteration path, closed on time etc. With many features provided by AX, it becomes easier for managers to track the health - *"I use this to gather hygiene for our teams so that I can fix it when things are not docked correctly or items are not closed even though they appear in the past iterations. As a manager, you want to keep your team's hygiene. With the right reports, you can go here and find it"* (P8). **57.1%** of the respondents mention using AX to monitor the overall health of the project.

4) **Track planned vs. actual progress (e.g., accountability)::** AX provides users with the possibility to track the amount of work planned against actual progress, thus, making team members accountable for their work. P7 expressed: *"My boss wants to know how many did we commit to originally and are we delivering against those commitments, how many did we add and are we delivering against those commitments and what if we deprioritize or remove."* **51.6%** of the respondents use AX for accountability reasons.

5) **Future planning::** An important part of any project is future planning where key stakeholders decide what they want to achieve against their current status. Features provided by AX helps teams visualize where they stand right now and effectively communicate that to the business stakeholders, who can then take crucial decisions. This planning can be both at the project and organization level. *"...planning and performance management within the system and then when we take the data out and look at it in Tableau, we are really looking not from a team's perspective, we are looking from overall organization...are we closing bugs in a timely manner and risk and requirement mitigation so we look at how many stories are getting moved after they are supposed to start..."* (P10). **47.3%** of the survey respondents mention using AX for future planning.

6) **Track untriaged vs triaged bugs::** AX is used by teams to track bugs that have been triaged against ones that are yet to be assigned. This lets teams visualize which bugs are assigned to whom and accordingly assign new bugs to team members on their current workload. One interviewee pointed out that AX helps answer the question of *"how many bugs are untriaged vs. triaged. We have been doing that since we switched over to VSTS."* (P1). **33%** of the respondents use AX for this reason.

7) **Track bugs with SLA::** A service-level agreement (SLA) is an official contract between service provider and service user. SLA documents the services that will be provided and aspects of service such as quality, availability etc. that a

provider is obligated to meet. Organizations offering Software as a Service (SaaS) products have service levels associated with managing defects that contain conditions by severity level such as response time, resolution time, time to triage etc. Some teams use AX to manage bugs that with SLAs and it helps them prioritize their work. *"We use this to identify what the priorities are. With those priorities, we have SLAs with that we can identify how many bugs are approaching their SLAs and that gives us an idea of how backlogged we are in terms of our team's resourcing. When something gets highlighted in our reports, then we question whether we have room to launch new features. Do we have the capacity to take on more work?"* (P1). **31.9%** of the users use AX to track bugs out of SLA.

8) **Improve the development process::** Development process defines how the teams are developing software such as agile, waterfall, spiral etc. Several users explained that with AX they can learn and improve their development process such as clearing up the backlog, tagging items correctly etc. A user mentions, *"We saw a high time taken to close user stories which led to a decision to have more granular stories so that we start closing work in a more granular and accountable manner. Our lead time was cut by a half."* **30.8%** mention using AX to improve the development process.

9) **Track individual performance::** Team members work on different things such as user stories, bugs, features etc. and it is important for managers to track who is working on what as well as developers to know which work items are assigned to them so that they can track their own progress. **28.6%** of respondents use AX to track their own performance. As put by a user, *"I learn how my topics are performing every day."*

10) **Examine churn (uncompleted work)::** AX is used by customers to generate reports to examine the churn of the team, i.e., amount of uncompleted work. This can help teams understand how much is the backlog, what resources and how much time would be required to accomplish those tasks. As opined by (P6), *"How much is the backlog, what is our churn rate all those calculations won't be possible without these services. That makes life easy for us. Why is this field (metric) going high? What are we going to do to solve this?"* **28.6%** of the survey respondents use AX to examine churn.

11) **Drive reliability::** AX is used to track particular features and ensure that they are reliable. Through continuous monitoring, reports created with AX can help users drive up the reliability by taking actions on time. As expressed by (P4), *"My primary use case is that I am trying to monitor the health of a particular Windows feature and drive up its reliability. We have telemetry that tells us this feature started and then 5 or 10 indicators have to go right for us to call it a success."*

12) **Track unit tests/coverage::** While many teams use AX to track features, user stories, bugs, etc. AX also provides a way to track unit test cases and associated code coverage results. This can help them understand and focus their testing efforts on the uncovered parts. *"We also want to know how terms of automation ... unit testing things like that ... how much coverage we have, how that initiative is going"* (P9).

13) **Assign cost to work items::** Organizations providing services to clients charge based on the number of services that a client subscribes to. Several customers using AX assign cost to the work items to understand how much each feature costs and increases awareness to clients on how much they are paying for each feature. As pointed out during the interviews, *“The second report that we have generated is what we call feature story point distribution. For each high-level feature-set that we have put in the requirements of that project - we group these and we say ‘This feature-set now we know it costs us this much’ so that users know what they are paying for each feature they are getting. We do that based on the dollar value of the team. Basically, we assign a dollar value to a story point and then we know how many story points fall within a feature. Using the task hierarchy within VSTS, we can group these and see relatively how much these things are costing us and how long these things are going to take us, when we are going to be able to deliver to the business”* (P9). 15.4% of the respondents mention using AX to assign cost to work items.

14) **Use it as a requirements engine::** Among other uses of AX, teams adopt it to track requirements or questions asked by the customers, thus, helping them understand and deliver on those requirements. (P2) mentioned, *“...we have access to large amounts of data across the company. A lot of different sources and datasets. So, we can answer a lot of questions for people. We needed a service to track and record to use that as a requirements engine for development work for our development team. Internal consumers will reach out to us with a question or a different request and then we search through the data to find the answer.”*

B. RQ2: How are customers using AX?

AX provides several ways in which users can interact and build reports. The most popular options are using VSTS dashboard, connecting with Power BI, Tableau and Microsoft Excel. Moreover, even for connecting with tools like Power BI there are multiple options such as direct OData feed, VSTS Data Connector, custom connectors etc. Reports can be created for personal or collective team usage and their consumption varies on the case by case basis. Here, we present the results of who is creating and viewing reports as well as how often they interact with them.

1) **Creating reports using AX::** During interviews, we find that reports for a team can be created by different users such as team member, manager, vendor etc. This depends on what is the intended usage of the reports and who is the target audience. We find that a large number of users create reports for themselves (Figure 3): *“When I create Power BI reports, I pull data from VSTS to show the bug status, the area path who the bug is assigned to.”* (P3) or is created by their team member or manager.

2) **Consuming reports from AX::** Similarly to above, reports might be consumed by the users for their own purpose, team usage, or for a client or partner. Over 80% of the survey respondents mention that the reports are used by their team (Figure 4): *“We look at these in the meeting and talk about*

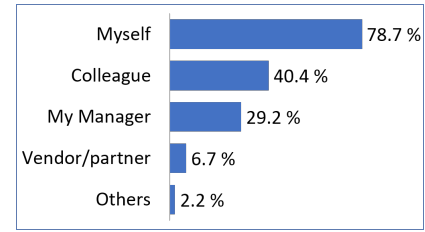


Fig. 3: Reports creation by users

them.” (P7), and 68.5% use it for their own use to track the work assigned to them: *“I will look at the report myself.”* (P3) or detect issues: *“I can look at it on my own to say if there is a gap that has to be addressed”* (P4).

Over 66% and 49% of the survey respondents mention that reports are consumed by their manager, and higher management or other stakeholders: *“Reporting is for the partners and the internal higher management of the customers. When they come, they should know how this build (product version) is performing for all the partners.”* (P5), respectively.

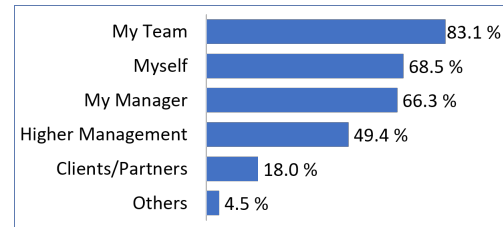


Fig. 4: Reports consumption by users

3) **Frequency of usage of reports in AX::** Similar to different usage patterns, the frequency also varies for different teams and users. While some teams look at reports daily, others might review them on a weekly or monthly basis depending on their needs. The breakdown of how frequently reports are viewed is shown in Figure 5. Over 60% of people use it *“on daily basis”* (P2), over 31% use weekly: *“Me and another PM look at this every week”* (P6), and over 7% use them once in two weeks. More than 6% of the users fall in the others category such as several times a week.

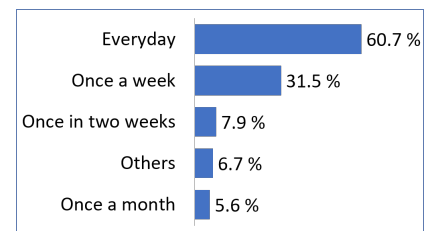


Fig. 5: Frequency of reports usage

4) **Location of usage of reporting from AX::** Due to a wide variety of usage of AX, reports can be displayed on different devices and locations based on the situation (Figure 6). It is unsurprising that a large number of users (>97%) view data on their desktop or laptop as over 67% of the users create

reports for themselves. As expressed during interviews, a large number of teams use reports regularly as part of their team meetings in a conference room: *“It’s a sync meeting. There are like 20 of us in the meeting and we look at it together.”* 8% of the respondents expressed that they use a common team room TV or mobile to view reports.

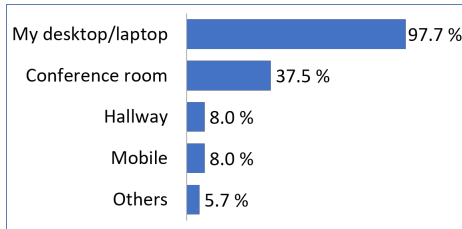


Fig. 6: Location of reports usage

5) Tools for creating reports:: AX provides several capabilities for users to create reports such as VSTS dashboard, Power BI, Microsoft Excel, Tableau or using custom tools. From the Figure 7, we observe that 69.7% of responders use VSTS dashboard. It is probably due to the ease of use, minimal setup requirements and the fact that widgets provide drag and drop capabilities as aptly put by P10, *“it was the dashboard and you can easily add these widgets.”* While VSTS dashboard is a popular choice, it does not cover all the possible reporting needs. Power BI is natural step forward as it provides more advanced capabilities to customize the dashboard by adding various kinds of charts that are not available as widgets. Moreover, AX has first-class integration with Power BI via dedicated data connector, which makes it simple to connect and import data. This ease of use attracts users to use Power BI. As such, we see that 58.4% of the users express using Power BI: *“Maybe a year ago or so Power BI started looking more widely used. We started to build a lot of our newer reports in Power BI. Now we have a bunch of dashboards in Power BI (service)”* (P1).

Microsoft Excel offers many basic and advanced features such as line chart, PivotTable, Power View etc. It can be used to view work items and to create reports: *“Basically, we have generated some reports just out of Excel files where we pull the generated query in VSTS ... pull that data into an Excel file and generate reports”* (P9). Over 39% of survey responders use Microsoft Excel to create reports. Apart from the above tools, users also choose Tableau, SQL Server Reporting Services (SSRS) etc.

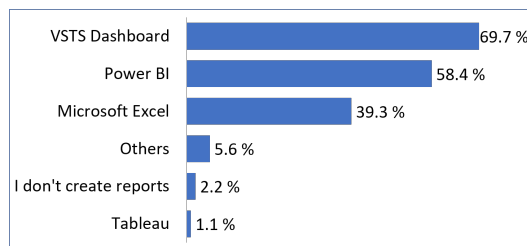


Fig. 7: Tools used for creating reports

6) Tools for viewing reports:: Similarly to the choice of tools for creating reports, there are many options for viewing and interacting with them. From Figure 8, we observe that over 74.2% of the respondents use VSTS dashboard for viewing reports. While a vast majority prefers using VSTS dashboard, some of them use *“Power BI for the visualization and VSTS for actual tracking”* (P2).

More than 55% of the users prefer to use Power BI website compared to 33.7% choosing Power BI Desktop. This could be due to Power BI website providing capabilities to both create dashboards and share them with others. Thanks to this feature, it becomes easier to share dashboards with all the team members and view them during meetings. Over 34% of the users prefer Microsoft Excel and around 4.5% select other tools.

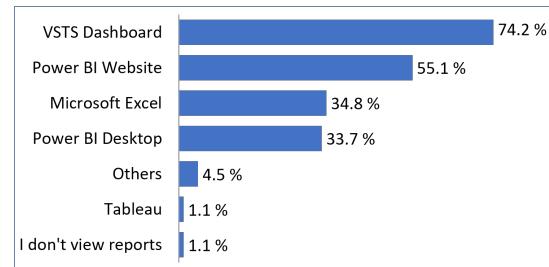


Fig. 8: Tools used for viewing reports

7) Date Range for reports used in AX:: AX provides users with the possibility of viewing different range of reports such as current state or historical trends. As presented in Figure 9, 81.8% of the respondents use AX reports to see the current state of the project such as the number of active bugs. This is important as teams would want to know their current status and then prioritize or assign more resources if needed. Over 78% of the users mention analyzing historical trends, e.g., progress since the beginning of the sprint. These trends help teams understand how they are performing over time and provide them evidence for future planning. 42% use historical point, i.e., snapshot as of first day of sprint. Over 11% of the users use different range of reports such as since the beginning of the year, product releases, i.e., every 6 months etc.

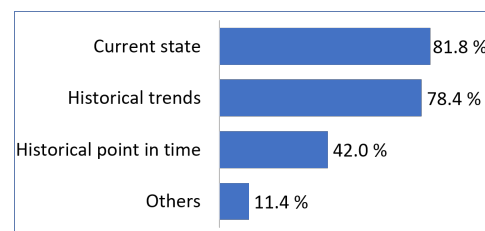


Fig. 9: Tools used for viewing reports

8) Exploratory analysis:: Users of a tool may perform exploratory analysis to learn more about the available functionalities, or they look for something specific. This can help understand how users can interact with the tool. For example, if users are doing more exploratory analysis,

it shows that they are curious to know more about what it has to offer. In our survey, over 65% of the users perform exploratory analysis and over 51% search for something specific.

Team Meetings

Team meetings are an important place of deliberation for teams in any organization. In our interviews and survey, we try to understand how users interact with AX during meetings. There are different types of meetings such as daily scrum, retrospective, sprint planning etc. that are part of a given software development process followed by a team or organization. Each of these meetings have different objectives. For example, daily scrum meetings are the time when team members explain what they did the previous day, what they plan to do and signal if there are any challenges. Sprint planning meetings, on the other hand, have a completely different format and audience. AX provides capabilities to build reports that can be used in different types of meetings. In our survey, we asked respondents on what are types of reports they use during meetings and what are the discussions that take place.

9) **Frequency of usage::** While most of the users look at the reports every day or once a week (see Section IV-B3), we wanted to know how often the teams look at the reports during meetings. From Figure 10, we observe that 58.6% of the teams use reports on a weekly basis and 25.3% on a daily basis. 9.2% of the teams expressed that they use reports on monthly basis, whereas other teams find the need to review them several times a week, every two weeks (sprint reviews).

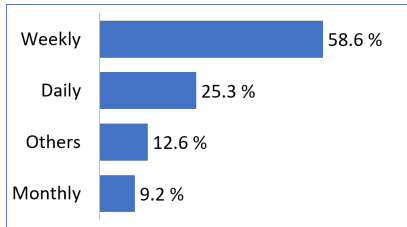


Fig. 10: Frequency of reports usage during team meetings

10) **Discussion during meetings::** Team meetings are an important place for a wide range of discussions such as current status of the project, work assigned to different team members, discuss roadblocks, or plan for upcoming sprints. In our survey, users describe how they spend time discussing reports (Figure 11) - 74.2% mention that everyone in the team discusses the work they are doing, 69.7% mention discussing the plan for the upcoming sprint and 64% specify that they only discuss open work items. Over 52% of the users mention that team meetings are focussed on discussing bugs/failures and their causes. Over 41% of the responders mention having discussions about long-term plans.

11) **Types of reports::** Currently, AX provides several widgets, i.e., cumulative flow diagram, velocity, lead and cycle time on VSTS dashboard. Apart from these, users reach out for Power BI, Excel, Tableau or custom apps and create customized charts to monitor different aspects of their projects.

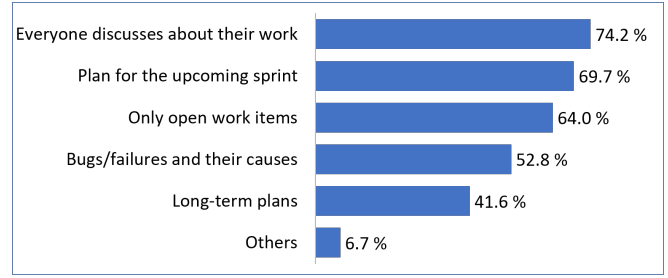
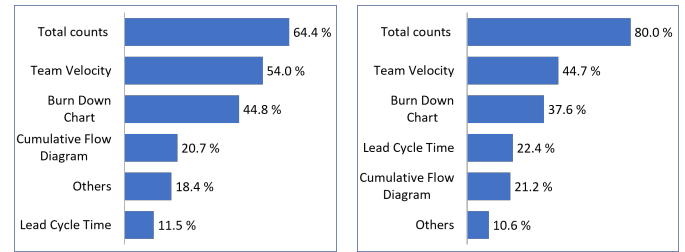


Fig. 11: Discussions during team meetings

From Figure 12a and 12b, we find that more than 64% and 80% of the users mention that they look at total counts (e.g., number of bugs) chart personally and during meetings, respectively. 44.7% and 37.6% mention using team velocity and burndown charts, respectively during team meetings. Almost 21% of the users admit to analyzing Cumulative Flow Diagram (CFD) during meetings.



(a) Personal Usage

(b) Meetings

Fig. 12: Types of reports used

C. RQ3: What are the learnings and outcomes of using AX?

With this question, we try to understand what are the learnings and outcomes of using AX, in particular, the decisions or actions taken by a team based on the reports. We asked our survey respondents “When was the last time you or anyone in your team took any decisions or actions based on the reports? Can you give an example?” Figure 13 shows that majority of the users or their teams have taken a decision in the past week. To further understand the decisions taken, we ask the respondents to specify the actions or decisions taken based on the data shown by the reports. We categorize these responses and present examples of decisions taken by the users below.

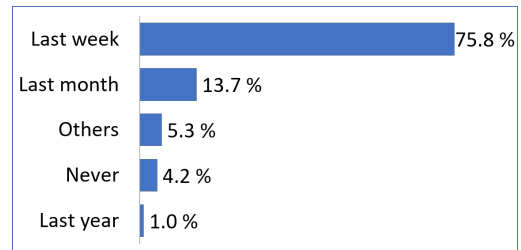


Fig. 13: Last time decisions taken based on the reports

1) **Better project tracking::** Users are building reports using AX that helps them to “*show how business progresses in certain areas*” to the relevant stakeholders. AX made it easier to track project progress and “*to see the completed work (user stories) and pending work*”. Based on the information available on the dashboard built by users, they can “*make recommendations based on it. For example, when do we need to of bug pushes, and how long do they need to last?*”

2) **Future planning::** Using AX, users were able to better plan their projects, as reports built using AX show the current status and the overall health of the project. This helps team gauge the work they can take up in the upcoming sprints and “*use the information to roadmap work for next months*”. As expressed by a user, “*I showed my manager the velocity chart. We’ve been better at planning for future sprints before the sprint starts now.*”

3) **Work prioritization::** Considering software engineers often face time constraints, it is important for teams to prioritize their work and deliver products to customers on time. Using AX, users were able to “*see that the amount of lower priority bugs were being solved quicker, therefore a concentrated push on high priority bugs were instigated.*” Some responders used AX reports to reprioritize “*work items, moving several items into backlog because they were repeatedly pushed down by the higher priority work.*” Furthermore, reports helped to highlight things that need immediate attention as mentioned by a respondent, “*this report, which is part of a dashboard, highlighted a problem that needed to be addressed very quickly to make sure we can deliver in September.*”

4) **Resource allocation::** Similar to work prioritization, teams do resource allocation based on the team’s capacity as well as skill sets of the team members. Allocation helps assign the right task and keep the workload balanced among the team members. Reports built using AX helped users to discover “*uneven work distribution among engineers*” and help to “*reassign some work items to employees or vendors.*”

5) **Detect issues::** As seen in Section IV-B11, users adopt various types of reports such as total counts, burn down chart, lead cycle time and team velocity. These reports can help users identify issues that might be otherwise unnoticed. For example, lead time measures the total time between the creation of work items to their completion and has helped users “*to spot issues in engineering lifecycle.*” As expressed in Section IV-B3, responders use reports on a daily basis for several reasons. One user mentioned, “*I look at my report daily to find any issues with the inventory system or the development process.*”

6) **Increasing visibility::** More than 71% of our survey respondents mentioned using AX reports to increase visibility of their team’s work to the stakeholders. AX provides capabilities to these teams to build reports that can be shown to higher management in the organization to give more exposure, “*to identify trends and take action either within the team or reach out to our engineering partners for them to take action accordingly.*”

7) **SLA Compliance::** Over 31% of the users mention having AX reports to track bugs with well defined SLAs. Reports helped them identify bugs with approaching SLA as well as ensure that the team satisfies the SLA requirements, as opined by a user, “*we have decided to check these reports daily and verify the resolved bugs as early as possible so that we know we meet the SLA.*”

8) **Rewards::** Following the crowdsourcing model, software organizations often have bug bounty programs to leverage strength of the engineers outside the organization to help find and fix bugs in the software [13]. Organizations often pay rewards in terms of financial incentives, hall of fame etc. All these motivate engineers to find more bugs. Our respondents expressed that AX reports were used to reward team members to encourage them to find bugs - “*We also use the bug dashboard to reward the best ‘bug hunters’.*”

V. LESSONS LEARNED

In this study, we present the reasons, usage patterns, outcomes and challenges that are associated with using a service in Microsoft that helps users build reports from the data in their work item tracking systems. AX has been able to help users better track their work items and manage their teams effectively. Its adoption has increased in recent months to several thousands of users both within and outside Microsoft. We present some of the key-takeaways for organizations and developers planning to build similar services or tools for users:

Good documentation can help increase the adoption of the tool. This is intuitive as good documentation can help novice users to get started as well as experienced users to know functionalities of the service. During interviews and survey, several users emphasized the importance of documentation - “*Improved documentation on how to do some basic, intermediate, and advanced things would help a great deal. Sometimes I just need a ‘starter’ to get me going and I can modify it from there.*” Apart from written documentation, having videos on how to use the tool goes a long way in making it easier for users to adopt. A user puts it succinctly, “*I would suggest that the team invest in how to videos much like Guy in a Cube (weekly video podcast) for solution selling with VSTS and Power BI for more adoption.*”

More education is required to spread awareness about the tool. With numerous tools available at the disposal of users, it has become increasingly important to educate users about various functionalities provided by the tool and at the same time spread awareness to increase the adoption by new users. As one user puts it, “*Come out with informative sessions to teach PMs about reports.*”

Customization diversifies the usage of the tool. Any service or tool should provide users with the flexibility to customize to suit their individual needs to increase the adoption. Currently, AX provides a set of charts and users can switch to Power BI to create more customized reports. As one user expresses, “*Can we write our own customize formula for creating graph similar like Power BI. It would be really helpful.*”

Increased **user awareness** of the processes and organization's work will help user appreciate the value of the tool. As one AX user mentions, *"It shows us the breadth of work across our org."* These users can then act as agents to spread information to their peers. Peer recommendation and observation are the two topmost reasons for adoption of AX (Figure 14). These findings corroborate previous research that peer influence has a significant adoption of new tools and services [14]. Furthermore, a clear recommendation from the engineering team (One Engineering System IES) also increases the adoption.

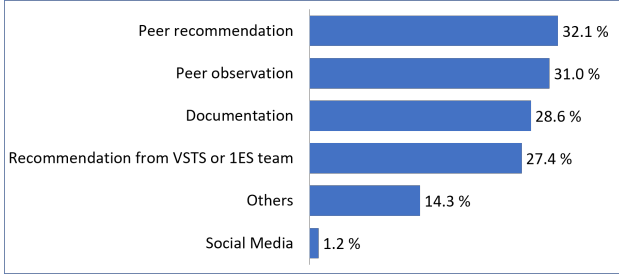


Fig. 14: Learning about AX

Better query support to help users only fetch the data they require and build reports from it. Currently, AX provides support for users to query and fetch the work items. There are some limitations imposed by the query language and protocol [4]. Therefore in some cases, it is necessary to fetch more data and interpret it on the client. As one user expresses, *"Would like to be able to query the work items... I want to create a chart in Power BI showing only the work items with resolved state equal to fixed and then a separate chart showing the work items with resolved state fixed, using the same data source."*

With the size of software crossing hundreds of thousands of lines of code, it has become increasingly important that the service or the tool is **scalable** to work with a large dataset. This can potentially be helpful for developers and organizations building various size of software. While AX is used by various large-scale products in Microsoft, we also see adoption by organizations outside Microsoft.

Tools or services that can help users **visualize their data** will have higher adoption as it can help generate meaning out of the data in an easy to digest format. Visualization can help reveal the data at several levels of granularity, from a broad overview to minute details. Furthermore, data visualization can help in effective storytelling with numbers to back it up and help in generating actionable insights. P6 expressed it during the interview, *"Visualization is always attractive and gives you a perspective than going through a big table. Obviously, it helps to save lot of time and help us prepare for reviews - we demonstrate everything we accomplish to the higher management every month."*

VI. THREATS TO VALIDITY

In this section, we describe several threats to validity for our study.

External Validity: As in any empirical study, our study too, has the threats to external validity. To reduce these threats, we conducted interviews with both internal as well as external (non-Microsoft) customers. Our survey, however, was conducted only on users of AX in Microsoft. Hence, we can not provide any guarantee of how the results would translate to another organization. However, we limit the threats to external validity by restricting the claims made in our paper to only the AX service at Microsoft and not to any other services that are available. We also tried to sample users based on frequency of usage and distinct queries written so that our results would generalize to other users both within and outside Microsoft.

Construct Validity: is the degree to which an experiment measures what it claims to be measuring. Such a threat could be possible when the questions we ask could potentially be misinterpreted by the participants. In order to minimize this threat, we carried out one interview and revised our questions based on the replies we got so that we were more precise. We also based our survey questions on the card sort analysis of the interviews that we carried out.

Internal Validity: In any qualitative study, there are the threats to internal biases impacting the coding process. We minimized this threat by involving both internal experts (developers who built the AX service) and outside observers (who are not Microsoft developers involved with the AX service). We also interview and survey active users of AX. While this may be a threat to this study, we believe that active users can give deeper insights about AX usage and challenges.

VII. RELATED WORK

Software development is a data-rich activity [15]. Software practitioners often gain knowledge by analyzing data from development team repositories or data collected specifically for studies. While the results of these data analyses are continually improving our understanding of software engineering, the actual work of data collection and analysis remains as clerical today as it was two decades ago. As a result, practitioners often struggle to satisfy their information needs, either because tools or services are unavailable, difficult to use, or simply not useful. Under such circumstances, software practitioners often resort to intuition-based decision making that may not always work out well [16]. Our goal with the Analytics service is to change this behavior and induce support more data-driven culture in software development activity.

Over the last two decades, empirical researchers have increasingly provided insights into software engineering, however, only a few studies have focused on understanding the information needs of software practitioners. Wallace et al. conducted a survey of 507 project managers and used cluster analysis to identify risk factors in projects [17]. They found that even low-risk projects have a high level of complexity. Begel and Zimmermann conducted a survey at Microsoft seeking questions that professional software engineers at Microsoft would like to ask data scientists to investigate about three topics: software, software processes and practices, and

software engineers [18]. **WorkItem measurements was one of the 12 categories** that they found in the study. Punter also conducted a survey results to identify the topics that interest practitioners, and what type of information is needed by practitioners [19]. Xia et al. surveyed 235 software practitioners to understand what developers search for on the web [20]. In particular they focussed on 34 search tasks that are divided into seven dimensions: general search, debugging and bug fixing, programming, third party code reuse, tools, database, and testing. Jedlitschka proposed a model to improve the delivery of relevant information from empirical software engineering research to software managers [15]. Using a controlled study, he showed the impact on costs, quality, schedule, and technology on a manager's decision. On one hand, our study is similar to prior work as it is also trying to understand the information needs of software practitioners using mixed investigative techniques. However, on the other hand, the study also adds a distinct value as it is the postmortem analysis of practitioners' experiences who are already using analytics service to track their WorkItems.

There are a number of existing tools designed to help practitioners with their information needs. For example, Hackstat is used for collection, analysis, visualization, interpretation, annotation, and dissemination of software development process and product data [21]. Similarly, PROM facilitates decision making by enabling practitioners with rich information on internal as well as external product and process attributes [22].

While the data-driven development culture is getting traction, and modern tools can present a large amount of data from varied sources, software managers are often too busy with their day-to-day duties to spend much time performing measurement activities [23]. Moreover, software practitioners also rely on colleagues for their information needs and do not value empirical evidence as much [24]. Due to such limitations and cultural barriers, it only becomes more important to get the tools/services providing such data right to not pose any more challenges in their adoption. Therefore, a retrospective analysis as in this study of how practitioners are using analysis tools/services would feed back into a better design of the tools or services to effectively fulfill their needs.

VIII. CONCLUSION

Previous research empirically shows that software engineers often face time constraints. Tools that help improve productivity will have more adoption. In our analysis, we find that AX has helped users improve their productivity by making it easier to visualize their data and exposing things to help users make data-driven decisions.

With more and more teams following the agile methodology and companies following the mantra of *fail fast, fail often*, services that help users **move fast** will provide an edge for them over their competitors. This will, in turn, lead to an increase in adoption of such services as they will have a positive impact on the customers' business. AX helps users in various ways and *"allows for a fail fast mentality because*

you can immediately get rid of something and plug something back in, the data will pick it up right away." (P8)

Thus, we hope that through our study we were able to highlight various reasons and the different ways in which stakeholders in an organization use reporting service for work item tracking systems. Any researcher or organization planning to build a similar service can use our study to understand product usage and customer challenges.

REFERENCES

- [1] J. Rossberg, *Pro Visual Studio Team System Application Lifecycle Management*. Apress, 2008.
- [2] S. Chanda and D. Foggon, *Beginning ASP.NET 4.5 Databases*, 3rd ed. Apress, 2013.
- [3] K. Kelly and S. Guckenheimer, "Work item tracking system for projects," 2012, uS Patent 8,126,760.
- [4] Edited by Michael Pizzo, Ralf Handl, and Martin Zurmuehl, "Odata version 4.0. part 1: Protocol plus errata 03," 2016-06-02. [Online]. Available: <http://docs.oasis-open.org/odata/odata/v4.0/odata-v4.0-part1-protocol.html>
- [5] A. Strauss and J. Corbin, *Grounded Theory in Practice*. SAGE Publications, 1997.
- [6] T. D. LaToza, G. Venolia, and R. DeLine, "Maintaining mental models: A study of developer work habits," in *ICSE*, 2006, pp. 492–501.
- [7] K. Muslu, C. Bird, N. Nagappan, and J. Czerwinka, "Transition from centralized to decentralized version control systems: A case study on reasons, barriers, and outcomes," in *ICSE*, 2014, pp. 334–344.
- [8] B. A. Kitchenham and S. L. Pfleeger, *Personal Opinion Surveys*. Springer London, 2008, pp. 63–92.
- [9] F. Fowler, *Improving Survey Questions: Design and Evaluation*. SAGE Publications, 1995.
- [10] F. J. Yammarino, S. J. Skinner, and T. L. Childers, "Understanding mail survey response behavior a meta-analysis," *Public Opinion Quarterly*, 1991.
- [11] P. K. Tyagi, "The effects of appeals, anonymity, and feedback on mail survey response patterns from salespeople," *Journal of the Academy of Marketing Science*, vol. 17, no. 3, pp. 235–241, 1989.
- [12] E. Smith, R. Loftin, E. Murphy-Hill, C. Bird, and T. Zimmermann, "Improving developer participation rates in surveys," in *CHASE*, 2013, pp. 89–92.
- [13] Bugcrowd. (2017) Bug bounty list. [Online]. Available: <https://www.bugcrowd.com/bug-bounty-list>
- [14] E. Murphy-Hill and G. C. Murphy, "Peer interaction effectively, yet infrequently, enables programmers to discover new tools," in *CSCW*, 2011, pp. 405–414.
- [15] A. Jedlitschka, "Evaluating a model of software managers' information needs: An experiment," in *ESEM*, 2010, pp. 19:1–19:10.
- [16] L. Strigini, "Limiting the dangers of intuitive decision making," *IEEE Software*, vol. 13, no. 1, pp. 101–103, 1996.
- [17] L. Wallace, M. Keil, and A. Rai, "Understanding software project risk: A cluster analysis," *Inf. Manage.*, vol. 42, no. 1, pp. 115–125, 2004.
- [18] A. Begel and T. Zimmermann, "Analyze this! 145 questions for data scientists in software engineering," in *ICSE*, 2014, pp. 12–23.
- [19] T. Punter, "What information do software engineering practitioners need?" pp. 85–95, 2003.
- [20] X. Xia, L. Bao, D. Lo, P. S. Kochhar, A. E. Hassan, and Z. Xing, "What do developers search for on the web?" *Empirical Software Engineering*, vol. 22, no. 6, pp. 3149–3185, 2017.
- [21] C. S. D. Laboratory. (2017-10-23) Hackstat. [Online]. Available: <http://csdl.ics.hawaii.edu/research/hackstat/>
- [22] P. M. Johnson, H. Kou, M. Paulding, Q. Zhang, A. Kagawa, and T. Yamashita, "Improving software development management through software project telemetry," *IEEE Software*, vol. 22, no. 4, pp. 76–85, 2005.
- [23] S. Komi-Sirvio, P. Parviainen, and J. Ronkainen, "Measurement automation: methodological background and practical solutions a multiple case study," in *METRIS*. IEEE, 2001, pp. 306–316.
- [24] A. Rainer, T. Hall, and N. Baddoo, "Persuading developers to 'buy into' software process improvement: a local opinion and empirical evidence," in *ISESE*, 2003, pp. 326–335.