The Emerging Role of Data Scientists on Software Development Teams

Miryung Kim
UCLA
Los Angeles, CA, USA
miryung@cs.ucla.edu

Thomas Zimmermann Robert DeLine Andrew Begel
Microsoft Research
Redmond, WA, USA
{tzimmer, rdeline, andrew.begel}@microsoft.com

ABSTRACT

Creating and running software produces large amounts of raw data about the development process and the customer usage, which can be turned into actionable insight with the help of skilled data scientists. Unfortunately, data scientists with the analytical and software engineering skills to analyze these large data sets have been hard to come by; only recently have software companies started to develop competencies in software-oriented data analytics. To understand this emerging role, we interviewed data scientists across several product groups at Microsoft. In this paper, we describe their education and training background, their missions in software engineering contexts, and the type of problems on which they work. We identify five distinct working styles of data scientists: (1) Insight Providers, who work with engineers to collect the data needed to inform decisions that managers make; (2) Modeling Specialists, who use their machine learning expertise to build predictive models; (3) Platform Builders, who create data platforms, balancing both engineering and data analysis concerns; (4) Polymaths, who do all data science activities themselves; and (5) Team Leaders, who run teams of data scientists and spread best practices. We further describe a set of strategies that they employ to increase the impact and actionability of their work.

Categories and Subject Descriptors:
D.2.9 [Management]

General Terms:
Management, Measurement, Human Factors.

1. INTRODUCTION

Software teams are increasingly using data analysis to inform their engineering and business decisions [1] and to build data solutions that utilize data in software products [2]. The people who do collection and analysis are called *data scientists*, a term coined by DJ Patil and Jeff Hammerbacher in 2008 to define their jobs at LinkedIn and Facebook [3]. The mission of a *data scientist* is to transform data into insight, providing guidance for leaders to take action [4]. One example is the use of user telemetry data to redesign Windows Explorer (a tool for file management) for Windows 8. Data scientists on the Windows team discovered that the top ten most frequent commands accounted for 81.2% of all of invoked commands, but only two of these were easily accessible from the command bar in the user interface 8 [5]. Based on this insight, the team redesigned the user experience to make these hidden commands more prominent.

Until recently, data scientists were found mostly on software teams whose products were data-intensive, like internet search and advertising. Today, we have reached an inflection point where many software teams are starting to adopt data-driven decision making. The role of data scientist is becoming standard on development teams, alongside existing roles like developers, testers, and program managers. Online service-oriented businesses such as Bing or Azure often require that software quality to be assessed in the field (testing in production); as a result, Microsoft changed the test discipline and hires data scientists to help with analyzing the large amount of usage data. With more rapid and continuous releases of software [6], software development teams also need effective ways to *operationalize* data analytics by iteratively updating the software to gather new data and automatically produce new analysis results.

So far, there have been only a few studies about data scientists, which focused on the limitations of big data cloud computing tools and the pain points that data scientists face, based on the experiences of participants from several types of businesses [7, 8]. However, these studies have not investigated the emerging roles that data scientists play within software development teams.

To investigate this emerging role, we interviewed 16 data scientists from eight different product organizations within Microsoft. During the period of our interviews, Microsoft was in the process of defining an official "career path" for employees in the role of data scientist, that is, defining the knowledge and skills expected of the role at different career stages. This process made Microsoft a particularly fruitful location to conduct our research, and several of our participants took part in this process. We investigated the following research questions:

Q1 Why are data scientists needed in software development teams and what competencies are important?

Q2 What are the educational and training backgrounds of data scientists in software development teams?

Q3 What kinds of problems and activities do data scientists work on in software development teams?

Q4 What are the working styles of data scientists in software development teams?

This paper makes the following contributions:

- We characterize the roles of data scientists in a large software company. (Section 4)
- We explore various working styles of data scientists. (Section 5)

The paper concludes with a discussion of implications (Section 6).

ICSE'16, May 14–22, 2016, Austin, TX, USA.
Copyright is held by the owner/author(s). Publication rights licensed to ACM. ACM 978-1-4503-3900-1/16/05 …$15.00.
DOI: http://dx.doi.org/10.1145/2884781.2884783

2. RELATED WORK

The work related to this paper falls into general data science and software analytics.

Data Science has become popular over the past few years as companies have recognized the value of data, for example, as data products, to optimize operations, and to support decision making. Not only did Davenport and Patil [9] proclaim that data scientist would be "the sexiest job of the 21st century," many authors have published data science books based on their own experiences, e.g., O'Neill and Schutt [10], Foreman [11], or May [12]. Patil summarized strategies to hire and build effective data science teams based on his experience in building the data science team at LinkedIn [3].

We found a small number of studies which systematically focused on how data scientists work inside a company. Fisher et al. interviewed sixteen Microsoft data analysts working with large datasets, with the goal of identifying pain points from a tooling perspective [7]. They uncovered tooling challenges in big data computing platforms such as data integration, cloud computing cost estimation, difficulties shaping data to the computing platform, and the need for fast iteration on the analysis results. However, they did not describe the roles that data scientists play within software development teams.

In a survey, Harris et al. asked 250+ data science practitioners how they viewed their skills, careers, and experiences with prospective employers [13]. Then, they clustered the survey respondents into four roles: Data Businesspeople, Data Creatives, Data Developers, and Data Researchers. They also observed evidence for so-called "T-shaped" data scientists, who have a wide breadth of skills with depth in a single skill area. Harris et al. focus on general business intelligence analysts rather than data scientists in a software development organization. Due to the nature of a survey research method, Harris et al. also do not provide contextual, deeper findings on what types of problems that data scientists work on, and the strategies that they use to increase the impact of their work.

Kandel et al. conducted interviews with 35 enterprise analysts in healthcare, retail, marketing, and finance [8]. Companies of all kinds have long employed business intelligence analysts to improve sales and marketing strategies. However, the data scientists we study are different in that they are an integral part of the software engineering team and focus their attention on software-oriented data and applications. Unlike our work, the Kandel et al. study does not investigate how data scientists contribute to software debugging, defect prediction, and software usage data (telemetry) collection in software development contexts.

Software Analytics is a subfield of *analytics* with the focus on *software data*. Software data can take many forms such as source code, changes, bug reports, code reviews, execution data, user feedback, and telemetry information. Davenport, Harris, and Morison [4] define analytics "as the use of analysis, data, and systematic reasoning to make decisions." According to an Accenture survey of 254 US managers in industry, however, up to 40 percent of major decisions are based on gut feel rather than facts [14]. Due to the recent boom in big data, several research groups have pushed for greater use of data for decision making [15, 16, 17] and have shared their experiences collaborating with industry on analytics projects [18, 16, 19].

Analysis of software data has a long tradition in the research communities of empirical software engineering, software reliability, and mining software repositories [1]. Software analytics has been the dedicated topic of tutorials and panels at the ICSE conference [20, 21], as well as special issues of IEEE Software (July 2013 and September 2013). Zhang et al. [22] emphasized the trinity of software analytics in the form of three research topics (development process, system, users) as well as three technology pillars (information visualization, analysis algorithms, large-scale computing). Buse and Zimmermann argued for a dedicated data science role in software projects [17] and presented an empirical survey with software professionals on guidelines for analytics in software development [23]. They identified typical scenarios and ranked popular indicators among software professionals. Begel and Zimmermann collected 145 questions that software engineers would like to ask data scientists to investigate [24]. None of this work has focused on the characterization of data scientists on software teams, which is one of the contributions of this paper.

Many software companies such as LinkedIn, Twitter, and Facebook employ data scientists to analyze user behavior and user-provided content data [25, 26, 27]. However, the authors of these published reports concentrate mainly on their "big data" pipeline architectures and implementations, and ignore the organizational architecture and work activities of the data scientists themselves. According to our study, data scientists in software teams have a unique focus in analyzing their own software teams' engineering processes to improve software correctness and developer productivity.

It is common to expect that action and insight should drive the collection of data. Goal-oriented approaches use goals, objectives, strategies, and other mechanisms to guide the choice of data to be collected and analyzed. For example, the Goal/Question/Metric (GQM) paradigm [28] proposes a top-down approach to define measurement; goals lead to questions, which are then answered with metrics. Other well-known approaches are GQM+ (which adds business alignment to GQM) [29], Balanced Scorecard (BSC) [30], and Practical Software Measurement [31].

Basili et al. [32] proposed the Experience Factory, which is an *independent* organization to support a software development organization in collecting experiences from their projects. The Experience Factory packages these experiences (for example, in models) and validates and reuses experiences in future projects. Some of the team structures that we observed in the interviews were similar to an Experience Factory in spirit; however, many data scientists were also directly embedded in the development organizations. While some experiences can be reused across different products, not all insight is valid and actionable in different contexts.

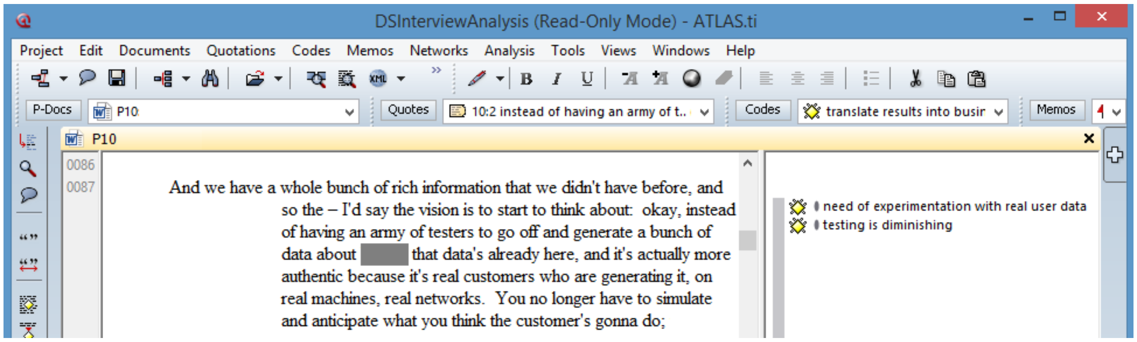
3. METHODOLOGY

We interviewed people who acted in the role of data scientists, then formed a theory of the roles that data scientists play in software development organizations.

Protocol. We conducted one-hour, semi-structured interviews, giving us the advantage of allowing unanticipated information to be mentioned [33]. All interviews were conducted by two people. Each was led by the first author, who was accompanied by one of the other three authors (as schedules permitted) who took notes and asked additional questions. Interviews were audio-taped and later transcribed for analysis. The interview format started with an introduction, a short explanation of the research being conducted, and demographic questions. Participants were then asked about the role they played on their team, their data science-related background, their current project(s), and their interactions with other employees. We also asked for stories about successes, pitfalls, and the changes that data is having on their team's practices. Our interview guide is in Appendix A.

TABLE 1. PARTICIPANT INFORMATION

	Title	Education
P1	Data Scientist II	BS in CS / Statistics, MS in SE, currently pursuing PhD in Informatics
P2	Director, App Statistics Engineer	MS in Physics
P3	Principal Data Scientist	MBA, BS in Physics / CS, currently pursuing PhD in Statistics
P4	Principal Quality Manager	BS in CS
P5	Partner Data Science Architect	PhD in Applied Mathematics
P6	Principal Data Scientist	PhD in Physics
P7	Research Software Design Engineer II	MS in Computer Science, MS in Statistics
P8	Program Manager	BS in Cognitive Science
P9	Senior Program Manager	BSE in CS and BAS in Economics/Finance
P10	Director of Test	BS in CS
P11	Principal Dev Manager	MS in CS
P12	Data Scientist	PhD in CS / Machine Learning
P13	Applied Scientist	PhD in CS / Machine Learning and Database
P14	Principal Group Program Manager	BS in business
P15	Director of Data Science	PhD in CS / Machine Learning
P16	Senior Data Scientist	PhD in CS / Machine Learning


DSInterviewAnalysis (Read-Only Mode) - ATLAS.ti
Project Edit Documents Quotations Codes Memos Networks Analysis Tools Views Windows Help
P-Docs
P10
Quotes
10:2 instead of having an army of t..
Codes
translate results into busir
Memos
P10
0086
0087

And we have a whole bunch of rich information that we didn't have before, and so the – I'd say the vision is to start to think about: okay, instead of having an army of testers to go off and generate a bunch of data about [redacted] that data's already here, and it's actually more authentic because it's real customers who are generating it, on real machines, real networks. You no longer have to simulate and anticipate what you think the customer's gonna do;

need of experimentation with real user data
testing is diminishing

Figure 1. An interview transcript excerpt in Atlas.TI. Using this tool, we added codes describing emerging themes.

Participants. In total, we interviewed 16 participants (5 women, 11 men) from eight different organizations at Microsoft: Advanced Technology Lab (1 participant), Advertisement and Monetization (1), Azure (2), Bing (1), Engineering Excellence (1), Exchange (1), Office (1), Skype (2), Windows (4), and Xbox (2).

We selected participants by *snowball sampling* [34]:

- First, we identified presenters at *data-driven engineering meetups* and *technical community* meetings, since these have been responsible internally for sharing best practices.
- Next, we selected additional data scientists by word-of-mouth, asking each participant to introduce us to other data scientists or other key stakeholders whom they knew.

At the time of this study in Summer 2014, there was no easy way to identify those who do data science work at Microsoft. In fact, Microsoft was in the process of creating a new job discipline called "data and applied science." Therefore, we used snowball sampling because it helped us locate hidden populations of data science practitioners, such as those employees working on data science tasks who do not have "data" or "data science" in their job title (see Table 1). As mentioned by P15, *"a lot of people kind of moonlighted as data scientists besides their normal day job."* Our sampling method may have caused us to miss some data scientists, however, to mitigate this threat, we seeded our sample with data science thought leaders from various product teams identified through company-wide engineering meetups and technical community talks.

Our findings reached saturation after interviewing 16 people. There was enough diversity in the participants' responses to enable us to find meaningful themes and draw useful interpretations. Stopping after saturation is standard procedure in qualitative studies.

Data Analysis. The authors individually used the Atlas.TI qualitative coding tool (http://atlasti.com/) to code emerging themes from the transcripts; together, we discussed the taxonomies derived from the interview transcripts. In concert, we employed affinity diagramming [35] and card sorting [36] to make sense of our data. Figure 1 shows a screen snapshot of Atlas.TI with an interview transcript excerpt and corresponding code describing emerging themes. In order to further help with traceability and to provide the details of our data analysis process, our technical report lists the codes we derived, along with supporting quotes [37].

To infer the working styles of data scientists (Q4), we performed two card sorts based on the roles data scientists played. One was done by the first author, another by the second and third authors. When participants shared experiences from multiple roles, we put each role on a separate card. This happened when participants shared experiences from previous jobs on different teams (P2 and P12) or had multiple responsibilities (P15 manages one team of engineers building a machine learning platform and another team of data scientists using the platform to create models). Both card sorts led to similar groupings of the participants, which are discussed later in the paper.

We categorized our results in terms of how and why data scientists are employed in a large software company (Section 4), the working styles that data scientists use with software development teams and their strategies to increase the impact of their work (Section 5).

Limitations. This is a qualitative study based on 16 interviews. In this paper, we share the observations and themes that emerged in the interviews. Because experts in qualitative studies specifically warn the danger of quantifying inherently qualitative data [38, 39], we do not make any quantitative statements about how frequently the themes occur in broader populations. We follow this guideline to focus on providing insights that contextualize our empirical findings, especially when they can serve as the basis for further studies, such as surveys.

Although this study was conducted only in one company, the participants came from eight different organizations, each working on different kinds of products. Several of our participants spoke of data science experiences they had prior to joining Microsoft. We believe that the nature of data science work in this context is meaningful, given that very few companies of a similar scale exist in the software industry. In addition, the software engineering research community also uses large-scale analysis of various types of software artifacts.

4. DATA SCIENTISTS IN SOFTWARE DEVELOPMENT TEAMS

Data science is not a new field, but the prevalence of interest in it at Microsoft has grown rapidly in the last few years. In 2015, one year after this study, over six hundred people are now in the new “data and applied science” discipline and an additional 1600+ employees are interested in data science work and signed up to mailing lists related to data science topics.

We observed an evolution of data science in the company, both in terms of technology and people. Product leaders across the company are eager to be empowered to make data-driven engineering decisions, rather than relying on gut feel. Some study participants who initially started as vendors or individual contributors have moved into management roles. Participants also reported that infrastructure that was initially developed to support a single data-driven task, for example, the Windows Error Reporting tool used for collecting crash data from in-field software deployments [40], was later extended to support multiple tasks and multiple stakeholders. Separate organizations within the company merged their data engineering efforts to build common infrastructure. The term, “data scientist” in this paper is a logical role, rather than the title of a position. Our goal is to understand and define this data scientist role by studying the participants in terms of their training and education background, what they work on, and how they fit in their organization, rather than restricting our study to those with the title “data scientists.”

4.1 Why are Data Scientists Needed in Software Development Teams?

Data-driven decision making has increased the demand for data scientists with statistical knowledge and skills. Specifically, participants described the increasing need for knowledge about experimental design, statistical reasoning, and data collection.

Demand for Experimentation. As the test-in-production paradigm for on-line services has taken off, our participants recognized the opportunity and need for designing experiments with real user data [41]. Real customer usage data is easier to obtain and more authentic than the simulated data test engineers create for anticipated usage scenarios.

> *Instead of having an army of testers to go off and generate a bunch of data, that data's already here. It's more authentic because it’s real customers on real machines, real networks. You no longer have to simulate and anticipate what the customer’s going to do.* [P10]

Participants mentioned an increase in the demand for experimenting with alternative software implementations, in order to assess the requirements and utility of new software features. Over the last decade, randomized two-variant experiments (called *A/B testing*) have been used to assess the utility of software prototypes and features, particularly for online services like web search. Because there are endless possibilities for alternative software designs, data scientists and engineering teams build software systems with an inherent capability to inject changes, called *flighting*.

> *You create an environment where, for example, in search, where I can actually experiment based on a mockup, if you will, of the idea. I can actually come up with a set of ideas, broad ideas about my work, and I can actually deploy them in some easy way.* [P5]

> *Do I change the size? Do I change the font? There are so many things you could do… We’re trying to flight things. It has capability to inject changes.* [P11]

Several participants took it upon themselves both to design incentive systems that get users to adopt a product feature and to create user telemetry and surveys that measure whether the systems worked.

> *So we create a game that gets people to repetitively use the feature. And then we watch what happens when we take the game away. Did it stick or did it not stick?* [P13]

Demand for Statistical Rigor. In the analysis of data, participants told us that there is an increasing demand for statistical rigor. Data scientists and their teams conduct formal hypothesis testing, report confidence intervals, and determine baselines through normalization.

For example, when participant P2 (who worked on estimating future failures) reported her estimate to her manager, the manager asked how confident she was. She gave him a hard number, surprising him because whenever he had asked the question to previous employees, he had just been told *highly* confident or *not very* confident.

> *He was like “So, you are giving me some predictions. How confident are you that this is what we get?” And I’m looking and go, “What do you mean? It’s 95 percent! It’s implied in all the testing. This is how we define this whole stuff.” And he goes, “Wow, this is the first time I’m getting this answer.”* [P2]

There has been a similar increase in the demand for conducting formal hypothesis testing. For example, instead of reasoning about averages or means, engineering teams want to see how different their observation is from random chance:

> *When I do my analyses, I always have a null hypothesis and an alternative hypothesis.* [P3]

Data scientists also have to determine a baseline of usual behavior so they can normalize incoming data about system behavior and telemetry data that are collected from a large set of machines under many different conditions.

> *I've got all of these different clients out in the wild running on all these different servers. I want to get a general sense of what things feel like on a normal Monday.* [P8]

Demand for Data Collection Rigor. When it comes to collecting data, data scientists discussed how much data quality matters and how many data cleaning issues they have to manage. Many participants mentioned that a large portion of their work required the cleaning and shaping of data just to enable data analysis. This aligns with a recent article on New York Times that said that 80% of data science work requires "janitor work" [42].

> *We need to cleanse the data, because there are all sorts of data quality issues [often, due to] imperfect instrumentation.* [P11]

Furthermore, data collection itself requires a sophisticated engineering system that tries to satisfy many engineering, organizational, and legal requirements.

> *What about storage, what about speed? What about legal, what about privacy? There is an entire gamut of things that you need to jump through hoops to collect the instrumentation.* [P1]

4.2 Background of Data Scientists

One column in Table 1 shows the educational background of the study participants. Data scientists often do not have a typical four-year degree in Computer Science [12]. In our study, 11 of 16 participants have degrees in Computer Science; however, many also have joint degrees from other fields such as statistics, physics, math, bio-informatics, applied math, business, economics, and finance. Their interdisciplinary backgrounds contribute their strong numerical reasoning skills for data analysis. 11 participants have higher education degrees (PhD or MS), and many have prior job experiences with dealing with big data.

Several non-CS participants expressed a strong passion for data.

> *I love data, looking and making sense of the data.* [P2]

> *I've always been a data kind of guy. I love playing with data. I'm very focused on how you can organize and make sense of data and being able to find patterns. I love patterns.* [P14]

When data scientists hire other data scientists, they sometimes look for skill sets that mirror how they were themselves trained. When one team manager with a PhD in machine learning spoke about hiring new employees for his data science tools team, he said that he looks for "machine learning hackers."

> *So, the typical guys on my team have some PhD in a quantitative field with machine learning background and the ability to code. They have to manipulate data. The other very essential skill is [that] we want programming. It's almost like ... a hacker-type skill set.* [P15]

Another data science team manager with strong statistics background demanded the same from everyone on his team:

> *My people have to know statistics. They need to be able to answer sample size questions, design experiment questions, know standard deviations, p-value, confidence intervals, etc.* [P2]

Our participants' background in higher education also contributes to how they view the work of data science. Usually, the problems and questions are not given in advance. A large portion of their responsibility is to identify important questions that could lead to impact. Then they iteratively refine questions and approaches to the analyses. Participants from a variety of product teams discussed how their training in a PhD program contributed to the working style they use to identify important questions and iteratively refine questions and approaches.

> *It has never been, in my four years, that somebody came and said, "Can you answer this question?" I mostly sit around thinking, "How can I be helpful?" Probably that part of your PhD is you are figuring out what is the most important questions.* [P13]

> *I have a PhD in experimental physics, so pretty much, I am used to designing experiments.* [P6]

> *Doing data science is kind of like doing research. It looks like a good problem and looks like a good idea. You think you may have an approach, but then maybe you end up with a dead end.* [P5]

4.3 Problems that Data Scientists Work on

Our participants said they worked on many kinds of problems ranging from performance and quality regression, user engagement and feature assessment, debugging and root cause analysis, bug reproduction, server log anomaly detection, failure rate estimation and failure planning. They also worked on business-specific problems, such as detecting fraud in e-commerce, identifying a mode of transportation for mobile users, and assessing advertisement ranking and news recommendations. Here are just a few of the example tasks that participants told us they worked on.

Performance Regression. *Are we getting better in terms of crashes or worse?* [P3] *How long did it take to detect when a new feature has blown up your system?* [P1]

Requirements Identification. *If you see the repetitive pattern where people don't recognize, the feature is there.* [P3]

Fault Localization and Root Cause Analysis. *What areas of the product are failing and why?* [P3] *How many failures are there per day?* [P11]

Bug Prioritization. *Oh, cool. Now we know which bugs we should fix first. Then how can we reproduce this error?* [P5]

Server Anomaly Detection. *We are interested in anomaly detection on real time servers in general.* [P7] *Is this application log abnormal w.r.t. the rest of the data?* [P12]

Failure Rate Estimation. *Is the beta ready to ship?* [P8]

Customer Understanding. *How long do our users use the app?* [P1] *What are the most popular features?* [P4] *Is my feature used in a way that improves the customer's productivity?* [P6]

Cost Benefit Analysis. *How much money can we save if we improve the AUC (i.e., area under the curve) for this machine learning classifier?* [P15] *How many customer service calls can we prevent if we detect this type of anomaly?* [P9]

4.4 Activities of Data Scientists

We found that data scientists worked on a variety of activities, which we organize into three categories: data collection, data analysis, and data use and dissemination. Please note that this list is not meant to be exhaustive. It is simply an overview of the activities we learned about from our study. (The mapping of activities to individual participants is shown in Table 2.)

Collection

TABLE 2. ACTIVITIES THAT PARTICIPANTS STATED THEY DID THEMSELVES (■) OR MANAGED (□)

		P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16
Collecting	Building the data collection platform	■			■				■			■		■	□		
	Injecting telemetry	■	□		■				■		□	■		■	□		
	Building the experimentation platform	■													□		
Analyzing	Data merging and cleaning	■	■	■	■	■	■	■	■	■	□	■		■	□		
	Sampling	■	■	■	■	■	■	■		■	□	■	■	■	□	■	■
	Shaping, feature selection		■	■	■	■	■	■			□		■	■	□	■	■
	Defining sensible metrics	■			■	■	■	■			□	■		■	□		■
	Building predictive models		■	■		■	■	■		■	□		■	■	□	■	■
	Defining ground truth							■		■			■	■	□	■	■
	Hypothesis testing		■	■		■	■				□				□		■
Using and Disseminating	Operationalizing models						■	■		■	□		■	■	□	■	
	Defining actions and triggers									■	■	■		■	□		
	Applying insights/models to business	■	■	■	■	■				■	■	■	■	■	□	■	■

- *Data engineering platform:* building a system for collecting data from multiple sources continuously
- *Telemetry injection:* inserting instrumentation code to gather software execution and usage profiles
- *Experimentation platform:* building inherent capability for experimentation with alternative software designs

Analysis

- *Data merging and cleaning:* joining data from multiple sources and dealing with missing values and imperfect instrumentation
- *Sampling:* selecting a subset set of behavior and weigh profiles to approximate normal behavior
- *Data shaping including selecting and creating features:* transforming data into a new format and creating new attributes in a feature vector
- *Defining sensible metrics:* defining metrics that are sensible to data consumers
- *Building predictive models:* building predictive models by applying machine learning, data mining, and statistics.
- *Defining ground truths:* defining class labels and scenarios of anomalies
- *Hypothesis testing:* setting a null hypothesis and an alternative hypothesis and estimating the confidence level of rejecting the null hypothesis using various statistical methods.

Use and Dissemination

- *Operationalizing predictive models:* integrating predictive models into software products and systems by invoking right models at a right point
- *Defining actions and triggers:* defining automated actions and triggers for different labels of predictions.
- *Translating insights and models to business values:* explaining the value of insights and predictive models using domain-specific terms.

4.5 Impact

When we asked the participants about their experiences in data science work that had impact and/or led to action, we heard several success stories. For example, several participants mentioned that their work on user engagement analysis led to new features which emerged from repetitive sequences of user actions that did not map to existing features. In some cases, their work also led the team to deprecate unused features. For example, participant P3 said that there was a feature that required a large amount of code, but nobody used it. His data science work led to identifying and deprecating the unused feature. Participant P2's work on failure rate estimation led to releasing a product two weeks earlier than the expected schedule. Another project on defect prediction enabled the team to rebalance resources to focus on bug fixing rather than adding new features. Root cause analysis of crash data led to automated bug filing and monitoring to reduce crash rates. Server log anomaly detection work led to reducing development operation cost.

> *Actionability is actually a big thing. If it's not actionable, the engineers then look at you, say, "I don't know what to do with this, so don't even bother me."* [P11]

4.6 Organization of Data Science Teams

Among the 16 interviewees, we observed five different ways of organizing a data science team or employing data scientists within an existing organization.

- *The "Triangle" model.* In a triangle team structure, a third of the team are data scientists who perform analysis work and who have a strong statistics background; another third are called data stewards who perform data shaping and cleaning tasks; and the rest collects customer usage data (telemetry) through instrumentation of software and hardware. [P2, P14]
- *The "Hub and Spoke" model.* In this model, a centralized team builds a common platform for data collection and analysis, which is used by spoke teams with product-specific knowledge to build product-specific models. [P1, P4]
- *The "Consulting" model.* An organization consults both internal and external customers by creating custom models and solving data problems of other teams within Microsoft. [P12]
- *The "Individual Contributor".* A software development team has a data scientist as an individual contributor. [P13]
- *The "Virtual Team" model.* The individual contributors from different teams form a virtual team and share common data collection and analysis tools for data science work. [P3]

5. DATA SCIENTIST WORKING STYLES

Though the role of data scientist is relatively new in software development, the interviews reveal commonalities in how the participants function on their teams. Nonetheless, each of our participants followed a unique path to their current role.

Based on two independent card sorts (described in Section 3), we grouped the participants into five distinct styles of data scientists. The first author initially grouped participants by their primary activities. For example, the first author noticed that P2, P3, and P9 participate in a similar set of activities with the goal of communicating their insights to managers (see columns in Table 1), while P1, P4, P8, P11, and P14 focus on building a data engineering pipeline. The second and third authors performed another separate card sort. All authors then collaboratively refined the groups to the ones listed in this section.

These working style groups are not mutually exclusive because some participants [P2, P12, P15] discussed their work on several different product teams.

In the next subsections, we characterize the nature of each style and include a participant success story to exemplify it.

5.1 Insight Providers

This working style characterizes data scientists who play an interstitial role between managers and engineers within a product group [P2, P3, P9]. The managers want to take actions to achieve business goals such as increased customer adoption, improved product quality, or shipping products. With a strong background in statistics, Insight Providers' main task is to generate insights and to support and guide their managers in decision making. These data scientists guide the managers' actions by analyzing product and customer data collected by the teams' engineers. Their communication and coordination skills are key—they negotiate with engineers to get the data they need, iterate with managers to understand and refine their goals, and communicate their findings clearly to the team.

Example. P2 worked on a product line in which the managers needed to know whether an upgrade was of sufficient quality to push to all products in the family. At first, she struggled to get quality crash data from the engineers:

> *I basically tried to eliminate from the vocabulary the notion of "You can just throw the data over the wall ... She'll figure it out." There's no such thing. I'm like, "Why did you collect this data? Why did you measure it like that? Why did you measure this many samples, not this many? Where did this all come from?"*

She worked with management to get a clear goal:

> *It should be as good as before. It should not deteriorate any performance, customer user experience that they have. Basically people shouldn't know that we've even changed [it].*

Her analysis was able to determine a confidence interval on the probability of field failures, allowing the team to know when they reached the quality bar.

As part of the strategy to increase the impact of their work, Insight Providers go a step further by defining actions and triggers associated with the resulting insight. Participant P9 provided an example in the context of server log anomaly detection, where each anomaly should trigger some action:

> *You need to think about, "If you find this anomaly, then what?" Just finding an anomaly is not very actionable. What I do also involves thinking, "These are the anomalies I want them to detect. Based on these anomalies, I'm going to stop the build. I'm going to communicate to the customer and ask them to fix something on their side."* [P9]

As Insights Providers need to communicate their results to managers and engineers, it is important for them to translate analysis results to concepts familiar to stakeholder's decisions. Occasionally, information is "lost in translation," i.e., when findings are simplified for people with no statistical training.

> *So I think part of the problem is that a lot of the people aren't given training in statistics... So you got some p-value of .01. "Oh, gee, should I be happy or sad? What does that mean?" And they don't necessarily know that. So I have to explain things in terms that might not be forceful enough.* [P3]

Another strategy for getting their insights heard is to interact closely and engage with the stakeholders who plan to consume the results from the data analysis. They often set up channels such as weekly data meet-ups [P3].

5.2 Modeling Specialists

This working style is practiced by data scientists who act as expert consultants and build predictive models [P7, P12]. With a strong background in machine learning, their main task is to build predictive models that can be instantiated as new software features (e.g., server telemetry anomaly detection) or to support other team's data-driven decision making. In this case, both P7 and P12 are experts in machine learning, though conceptually other forms of expertise (statistics, survey design) would fit here as well.

Example. P7 is an expert in time series analysis and works with P9 to help her team automatically detect anomalies in their telemetry data.

> *The PMs [Program Managers] and the Dev Ops from that team...through what they daily observe, come up with a new set of time series data that they think has the most value and then they will point us to that, and we will try to come up with an algorithm or with a methodology to find the anomalies for that set of time series.* [P7]

Modeling Specialists sometimes partner with Insight Providers to define ground truths to assess the quality of their predictive models. As an example, participant P7, a Modeling Specialist, and participant P9, an Insight Provider, iteratively defined which events should be considered as server operation anomalies because the ground truth required for each analysis was often not known in advance.

> *You have communication going back and forth where you will find what you're actually looking for, what is anomalous and what is not anomalous in the set of data that they looked at.* [P7]

> *When you're seeing this part of the data, this one's good versus here's setting that ground truth. Here's where you should have alerted. Here's where you shouldn't have done anything. That's something that we are continuing to iterate on, but that's something that was fairly labor-intensive.* [P9]

Several interviewees reported that "*operationalization*" of their predictive models—building new software features based on the predictive models is extremely important for demonstrating the value of their work. However, this step of going the last mile is often difficult for Modeling Specialists, such as Participants P7 and P12. With each product team they were assigned to help, they had to get their algorithms running on a new infrastructure, and too often, had to make code changes to the infrastructure, itself.

> *Getting your algorithm at the right point to make sure right models are loaded. That's a big issue we face.* [P7]

> *They accepted [the model] and they understood all the results and they were very excited about it. Then, there's a phase that comes in where the actual model has to go into production. ... You really need to have somebody who is confident enough to take this from a dev side of things.* [P12]

Modelling Specialists also said it was important to “translate” findings into business values, such as dollars saved, customer calls prevented, or the number of days early that a product can be shipped. Precision, recall, and ROC curves, while popular with data scientists and academics, are less useful when presenting findings to analytics consumers.

> *In terms of convincing, if you just present all these numbers like precision and recall factors... that is important from the knowledge sharing model transfer perspective. But if you are out there to sell your model or ideas, this will not work because the people who will be in the decision-making seat will not be the ones doing the model transfer. So, for those people, what we did is cost benefit analysis where we showed how our model was adding the new revenue on top of what they already had.* [P12]

5.3 Platform Builders

This working style is demonstrated by seven data scientists who build shared data platforms used across several product teams [P1, P4, P6, P8, P11, P14, P15]. Of these seven, six work on data pipelines for data collection, storage, and querying, while P15 works on a service for building and deploying machine learning models. With a strong background in big data systems, their main task is to build a data engineering platform.

A defining characteristic of this working style is that they produce software systems designed to be reusable across many different product and business goals. The platform builders’ work balances both engineering and scientific concerns. For example, data collection software must be reliable, performant, low-impact, and widely deployable. On the other hand, the software should provide data that are sufficiently precise, accurate, well-sampled, and meaningful enough to support statistical analysis. Their expertise in both software engineering and data analysis enables them to make tradeoffs between these concerns.

We found two kinds of data platform builders. Participants P1, P4, P8 and P11 work with on systems that involve platforms to collect in-field software failure data, including Windows Error Reporting [40] and Reliability Analysis Component [43]. Their work unites these data sources into a common platform to fit current business goals. Participants P6, P14, and P15 work on new data collection and measurement platforms. For example, P14 works on a new common logging platform, and has the freedom to design new data schemas.

Example. P4 worked on a data platform that collects crash data and worked on making it actionable to developers.

> *You come up with something called a bucket feed. It is a name of a function most likely responsible for the crash in the small bucket. We found in the source code who touch last time this function. He gets the bug. And we filed [large] numbers a year with [a high] percent fix rate.*

As Platform Builders construct data engineering pipelines to collect measurements from various sources, data quality and cleansing is very important. They often triangulate multiple data sources to increase the confidence in the analysis results. They validate quantitative data through qualitative channels to ensure that measurements are meaningful and lead to correct actions. For example, Participant P4 discussed the importance of validating his product’s telemetry data through subjective channels:

> *If you could survey everybody every ten minutes, you don’t need telemetry. The most accurate is to ask everybody all the time. The only reason we do telemetry is that [asking people all the time] is slow and by the time you got it, you’re too late. So you can consider telemetry and data an optimization. So what we do typically is 10% are surveyed and we get telemetry. And then we calibrate and infer what the other 90% have said.* [P4]

Talking to non-experts also required the development of intuitive measurements. Participant P4 measured the impact of a software crash by the associating it with how many minutes his customers wasted because of it — a number that is easy to understand and assess over time.

5.4 Polymaths

This working style describes data scientists who “do it all,” e.g., forming a business goal, instrumenting a system to collect the required data, doing necessary analyses or experiments, and communicating the results to business leaders [P12, P13, P16]. In this working style, the boundary between the data scientist role and a software engineer role is not strict. They are naturally intertwined because they undertake activities common to both roles.

Example. P13 works on a product that serves advertisements and explores her own ideas for new advertisement data models.

> *So I am the only scientist on this team. I'm the only scientist on sort of sibling teams and everybody else around me are like just straight-up engineers.*

She expressed enthusiasm for her ability to operationalize her own models.

> *For months at a time I'll wear a dev hat and I actually really enjoy that, too. ... I spend maybe three months doing some analysis and maybe three months doing some coding that is to integrate whatever I did into the product. ... I do really, really like my role. I love the flexibility that I can go from being developer to being an analyst and kind of go back and forth.*

Polymaths embedded in some product teams often switched modes between modelling and deployment.

> *I kind of flip back and forth. I say I spend maybe three months doing some analysis, and maybe three months doing some coding that is to integrate whatever I did into the product.* [P13]

Polymaths set up regular channels such as “brown bag lunches” to deliver their project outcomes to their team.

5.5 Team Leaders

The last working style describes senior data scientists who run their own data science teams [P2, P5, P10, P15]. In addition to managing their teams, they also act as data science “evangelists,” pushing for the adoption of data-driven decision making within their business organization or the company as a whole. Data team leaders work with senior company leaders to inform broad business decisions.

Example. P10 and his team of data scientists estimated the number of bugs that would remain open when a product was scheduled to ship.

> *When the leadership saw this gap [between the estimated bug count and the goal], the allocation of developers towards new features versus stabilization shifted away from features toward stabilization to get this number back.*

P10 emphasized his role as intermediary between his data scientist and his management:

> *Sometimes people who are real good with numbers are not as good with words (laughs), and so having an intermediary to sort of handle the human interfaces between the data sources and the*

> *data scientists, I think, is a way to have a stronger influence. [Acting] an intermediary so that the scientists can kind of stay focused on the data.*

To increase the impact of their work, Team Leaders emphasized the importance of choosing the right questions for the right team. Participant P5 described three conditions that must be met before his data science team engages in a project*: priority, actionability,* and *commitment*:

> *(a) Is it a priority for the organization (b) is it actionable, if I get an answer to this, is this something someone can do something with? and, (c), are you as the feature team — if you're coming to me or if I'm going to you, telling you this is a good opportunity — are you committing resources to deliver a change? If those things are not true, then it's not worth us talking anymore.* [P5]

Team Leaders also mentioned the importance of working closely with consumers from day one. Their team must communicate with stakeholders early and often to define the questions and scenarios. They also emphasized the need to explain findings in simple terms to non-experts, especially to management.

> *You begin to find out, you begin to ask questions, you being to see things. And so you need that interaction with the people that own the code, if you will, or the feature, to be able to learn together as you go and refine your questions and refine your answers to get to the ultimate insights that you need.* [P5]

> *A super smart data scientist, their understanding and presentation of their findings is usually way over the head of the managers…so my guidance to [data scientists], is dumb everything down to seventh-grade level, right? And whether you're writing or you're presenting charts, you know, keep it simple.* [P10]

6. IMPLICATIONS

The findings in this paper have several implications for research, practice, and education.

6.1 Research

Many development teams now include data scientists as a standard role, alongside developers, testers, and program managers. For researchers, this new team composition changes the context in which problems are pursued. Many development teams are already collecting and monitoring data about user behavior, software execution, and team activities, as well as contextual data sources like social media. This means that researchers can assume the availability of such data, as well as an expert team member to handle them, as an ingredient for solving problems. Conversely, new technology that ignores the availability of such data could be less attractive for adoption in industry. Given the novelty of the role, emerging data scientist will also experience frustrations and inefficiencies, which are another target for research. While some of frustrations have been explored in some related work [7] [13], we expect distinct challenges and opportunities for software-oriented data scientists.

We observed a strong influence of higher education on data science (11 of the participants had PhD or MS degrees). To an extent, this is a testament to the transfer of many years of software engineering research to practice. The problems that data scientists work on — bug prediction, debugging, release planning, and anomaly detection in server telemetry — and the analysis methods that they use for solving these problems are similar to those employed in the software engineering research field for the past decade. As techniques that were once novel in the research literature become standard among data scientists, researchers will need to refocus to stay ahead. As one example, researchers could invent new analysis techniques to allow data scientists to analyze new kinds of data. Or they could focus on better tool support to automate the collection and analysis of data. Validating operationalized data solutions is a challenging task that requires *"careful inspection to understand the provenance and distribution of each piece of data, of the problem domain, and of the practices used"* [2], including assessing the quality of data. We expect that debugging for software-oriented data scientists is an important research topic to be addressed by the software engineering research community.

We believe that the strategies that data scientists use to ensure the impact and actionability of their work can also be used to increase the impact of software engineering research. The data scientists shared the importance of going the last mile to operationalize the predictive models and tailor the value of their insights for each beneficiary. Our participants reported that while precision, recall, and ROC curves are commonly used to evaluate (predictive) research, they are not effective in gaining buy-in from engineers or prompting management to take action. To increase the impact of research, we, software engineering researchers, must also operationalize the results of data analytics by defining actions and triggers than simply reporting increases in precision and recall.

6.2 Practice

The software world has changed over the past years. With cloud-based systems, the availability of operational data has significantly increased. Monetization of software now more heavily relies on a good understanding of how customers use software. There are also new opportunities for more efficient software development such as testing in production [41] [44] and the ability to flight changes for a short time before making them final [45]. We believe that these changes lead to an increased demand for data scientists in the software industry similar to what we see in other industries. By 2018, the U.S. may face a shortage of as many as 190,000 people with analytical expertise and of 1.5 million managers and analysts with the skills to make data-driven decisions, according to a report by the McKinsey Global Institute [46].

In this new world that is ruled by data, software companies have to figure out what data scientists really are, what skills they need, who to hire, and where to put them in their organization. More concretely, testers should be trained with a data science skill set, as the assessment of software quality and correctness increasingly depends on analysis of large-scale usage data. The success stories, activities, and working styles of data scientists, which we reported in this paper, can serve as guidelines for structuring software organizations to include data scientists and to improve data-driven engineering decision making. For example, organizations that would like to adopt and employ data scientists may structure their teams using the triangle model or the hub and spoke model described in Section 4.6. Data scientists who are hired into software development teams can also learn how to improve the impact and actionability of data science work from the strategies shared by other data scientists.

6.3 Education

As illustrated by the Polymath working style, data science is not always embodied as a distinct role on the team, but sometimes as a skillset that blends with other skills such as software development. Polymaths may become the prevalent work style, if data science follows the precedent of software testing. Historically, testing was the domain of a distinct role of testers. Later, however, with the rise of unit testing and test-driven development, testing became a skill the developer role practiced as well [47]. Similarly, over time, data science may become less of a distinct role and more a skillset that many team members employ. Indeed, every team role has distinct information needs that could be answered through data analysis. For instance, program managers have questions about feature use; testers have questions about hot paths; developers have questions about execution. This implies that there is increasing demand for an integrated software engineering and data science curriculum.

The characterization of data scientists in this paper can also be used in software engineering courses to illustrate real-life data science. The activities that data scientists participate in and the skill sets required can be useful to undergraduate and graduate educators who teach computer science, statistics, and data science courses. Data scientists need to combine a deep understanding of software engineering problems, strong numerical reasoning skills, strong programming skills, and the ability to communicate the value of models and insights in domain- and business-specific terms. Computer science students commonly hold wildly inaccurate preconceptions about the character of the work they will encounter in their future software engineering careers [48, 49]

7. CONCLUSIONS

In this paper we characterized the role of data scientists in a large software company. We observed a demand for designing experiments with real user data and reporting results with statistical rigor. We shared activities, several success stories, and five distinct styles of data scientists. We reported strategies that data scientists use to ensure that their results are relevant to the company. For future work, we plan to conduct a large scale survey of data scientists to quantify the working styles and tasks observed in this study and to shed light onto the challenges associated with data science work.

8. ACKNOWLEDGMENTS

We thank all interview participants for sharing their story, connecting us with other data scientists, and providing insights. We thank Alan Ho, Judith Bishop, Brendan Murphy, Kim Herzig, Michaela Greiler, Tom Ball, Ben Zorn, Sumit Gulwani, Todd Mytkowitz, Chris Bird, Nachi Nagappan, Mike Barnett, Ratul Mahajan, Jaeyeon Jung, and Madan Musuvathi for their curiosity. We thank the anonymous reviewers for their valuable feedback on earlier versions of this paper. Miryung Kim performed this work as a visiting researcher at Microsoft Research. Miryung Kim was in part supported by NSF grants 1238498, 1149391, and 1117902.

APPENDIX A: INTERVIEW GUIDE

INTRODUCTION

GOAL. The emerging role and impact of data scientists in software development teams.

- Our goal is to conduct a broad survey on how “big software data” impacts engineering teams across different organizations at Microsoft, and how data scientists and other team members coordinate, communicate, and make decisions based on data-driven insights.
- Spread best practices for making data-driven engineering decisions.

LOGISTICS

- Confidentiality. Anonymization. Participant sign off.
- Audio Recording.

DEMOGRAPHICS

- How long have you been at Microsoft?
- What do you do at Microsoft?

ROLE. What do you consider your role to be here?

BACKGROUND

- How did you get into data science? Background / training / education/ resources that you've taken to help with your job

CURRENT PROJECT. Tell me about your current project:

- Analysis / decision support questions?
- Kinds of data?
- How long?

SUCCESS STORY in the PAST: (specific / particular one)

- What kinds of information/ insights?
- Who/ How do you work with and share insights?
- What happened as a result of your analysis? Can you tell us about business impact?

PITFALL in the PAST: (specific / particular one)

- What kinds of information / insights?
- Who / How do you work with and share insights?
- Training / resources would you need?
- What would make your job more effective?

TREND & IMPACT. How data is changing your team?

TOOL/ ENVIRONMENT

- Would you show us your work environment? Tools, Data?

CONTACT

- Is there anybody else that we should talk to? Others who work with data—the ones are producing, storing, and using the information, etc.

REFERENCES

[1] T. Menzies and T. Zimmermann, "Software Analytics: So What?," *IEEE Software,* vol. 30, no. 4, pp. 31-37, July 2013.

[2] A. Mockus, "Engineering big data solutions.," in *Fose '14: Proceedings of the on Future of Software Engineering*, Hyderabad, India, 2014.

[3] D. Patil, Building Data Science Teams, O'Reilly, 2011.

[4] T. H. Davenport, J. G. Harris and R. Morison, Analytics at Work: Smarter Decisions, Better Results, Harvard Business Review Press, 2010.

[5] A. Simons, "Improvements in Windows Explorer," http://blogs.msdn.com/b/b8/archive/2011/08/29/improvements-in-windows-explorer.aspx, 2011.

[6] B. Adams, S. Bellomo, C. Bird, T. Marshall-Keim, F. Khomh and K. Moir, "The Practice and Future of Release Engineering: A Roundtable with Three Release Engineers," *IEEE Software,* vol. 32, no. 2, pp. 42-49, 2015.

[7] D. Fisher, R. DeLine, M. Czerwinski and S. M. Drucker, "Interactions with big data analytics," *Interactions,* vol. 19, no. 3, pp. 50-59, 2012.

[8] S. Kandel, A. Paepcke, J. Hellerstein and J. Heer, "Enterprise Data Analysis and Visualization: An Interview Study," in *IEEE Visual Analytics Science & Technology (VAST)*, 2012.

[9] T. H. Davenport and D. Patil, "Data Scientist: The Sexiest Job of the 21st Century," *Harvard Business Review,* pp. 70-76, OCtober 2012.

[10] C. O'Neil and R. Schutt, Doing Data Science: Straight Talk from the Frontline, O'Reilly Media, 2013.

[11] J. W. Foreman, Data Smart: Using Data Science to Transform Information into Insight, Wiley, 2013.

[12] T. May, The New Know: Innovation Powered by Analytics, Wiley, 2009.

[13] H. D. Harris, S. P. Murphy and M. Vaisman, Analyzing the Analyzers: An Introspective Survey of Data Scientists and Their Work, O'Reilly, 2013.

[14] Accenture, *Most U.S. Companies Say Business Analytics Still Future Goal, Not Present Reality,* http://newsroom.accenture.com/article_display.cfm?article_id=4777, 2008.

[15] A. E. Hassan and T. Xie, "Software intelligence: the future of mining software engineering data," in *FOSER '10: Proceedings of the Workshop on Future of Software Engineering Research*, 2010.

[16] D. Zhang, Y. Dang, J.-G. Lou, S. Han, H. Zhang and T. Xie, "Software Analytics as a Learning Case in Practice: Approaches and Experiences," in *MALETS '11: Proceedings International Workshop on Machine Learning Technologies in Software Engineering*, 2011.

[17] R. P. L. Buse and T. Zimmermann, "Analytics for software development," in *FOSER '10: Proceedings of the Workshop on Future of Software Engineering Research*, 2010.

[18] J.-G. Lou, Q. W. Lin, R. Ding, Q. Fu, D. Zhang and T. Xie, "Software Analytics for Incident Management of Online Services: An Experience Report," in *ASE '13: Proceedings of the Internation Conference on Automated Software Engineering*, 2013.

[19] T. Menzies, C. Bird, T. Zimmermann, W. Schulte and E. Kocaganeli, "The Inductive Software Engineering Manifesto: Principles for Industrial Data Mining," in *MALETS '11: Proceedings International Workshop on Machine Learning Technologies in Software Engineering*, 2011.

[20] D. Zhang and T. Xie, "Software analytics: achievements and challenges," in *ICSE '13: Proceedings of the 2013 International Conference on Software Engineering*, 2013.

[21] D. Zhang and T. Xie, "Software Analytics in Practice," in *ICSE '12: Proceedings of the International Conference on Software Engineering.*, 2012.

[22] D. Zhang, S. Han, Y. Dang, J.-G. Lou, H. Zhang and T. Xie, "Software Analytics in Practice," *IEEE Software,* vol. 30, no. 5, pp. 30-37, September 2013.

[23] R. P. Buse and T. Zimmermann, "Information needs for software development analytics," in *ICSE '12: Proceedings of 34th International Conference on Software Engineering*, 2012.

[24] A. Begel and T. Zimmermann, "Analyze This! 145 Questions for Data Scientists in Software Engineering," in *ICSE'14: Proceedings of the 36th International Conference on Software Engineering*, Hyderabad, India, 2014.

[25] J. Lin and D. Ryaboy, "Scaling Big Data Mining Infrastructure: The Twitter Experience," *SIGKDD Explorations,* vol. 14, no. 2, pp. 6-19, April 2013.

[26] A. Thusoo, Z. Shao, S. Anthony, D. Borthakur, N. Jain, J. S. Sen, R. Murhty and H. Liu, "Data Warehousing and Analytics Infrastructure at Facebook," in *Proceedings of ACM SIGMOD International Conference on Management of Data*, New York, NY, 2010.

[27] R. Sumbaly, J. Kreps and S. Shah, "The Big Data Ecosystem at LinkedIn," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, New York, NY, 2013.

[28] V. R. Basili, "Software modeling and measurement: the Goal/Question/Metric paradigm," College Park, MD, USA , 1992.

[29] V. R. Basili, M. Lindvall, M. Regardie, C. Seaman, J. Heidrich, J. Münch, D. Rombach and A. Trendowicz, "Linking software development and business strategy through measurement.," *IEEE Computer,* vol. 43, p. 57–65, 2010.

[30] R. Kaplan and D. Norton, "The balanced scorecard—measures that drive performance," *Harvard Business Review,* pp. 71-80, January/February 1992.

[31] J. McGarry, D. Card, C. Jones, B. Layman, E. Clark, J. Dean and F. Hall, Practical Software Measurement: Objective Information for Decision Makers, Addison-Wesley Professional, 2001.

[32] V. R. Basili, "The experience factory and its relationship to other," in *ESEC'93: Proceedings of European Software Engineering Conference on Software Engineering* , 1993.

[33] C. B. Seaman, "Qualitative Methods," in *Guide to Advanced Empirical Software Engineering*, F. Shull, J. Singer and D. I. Sjøberg, Eds., Springer, 2008.

[34] L. Goodman, "Snowball sampling," *Annals of Mathematical Statistics,* vol. 32, no. 1, p. 148–170, 1961.

[35] S. J. Janis and J. E. Shade, Improving Performance Through Statistical Thinking, ASQ Quality Press, 2000.

[36] D. Spencer, Card Sorting: Designing Usable Categories, Rosenfeld Media, 2009.

[37] M. Kim, T. Zimmermann, R. DeLine and A. Begel, "Appendix to The Emerging Role of Data Scientists on Software Development Teams," Microsoft Research. Technical Report. MSR-TR-2016-4. http://research.microsoft.com/apps/pubs/?id=261085, 2016.

[38] R. K. Yin, Case Study Research: Design and Methods, SAGE Publications, Inc; 5 edition, 2013.

[39] N. K. Denzin and Y. S. Lincoln, The SAGE Handbook of Qualitative Research, SAGE Publications, Inc; 4 edition, 2011.

[40] K. Glerum, K. Kinshumann, S. Greenberg, G. Aul, V. Orgovan, G. Nichols, D. Grant, G. Loihle and G. Hunt, "Debugging in the (Very) Large: Ten Years of Implementation and Experience," in *SOSP '09: Proceedings of the 22nd ACM Symposium on Operating Systems Principles*, 2009.

[41] R. Musson and R. Smith, "Data Science in the Cloud: Analysis of Data from Testing in Production," in *TTC '13: Proceedings of the International Workshop on Testing the Cloud* , 2013.

[42] S. Lohr, *For Big Data Scientists, "Janitor Work" is Key Hurdle to Insights,* New York Times, Aug. 17, 2014.

http://www.nytimes.com/2014/08/18/technology/for-big-data-scientists-hurdle-to-insights-is-janitor-work.html?_r=1.

[43] P. L. Li, R. Kivett, Z. Zhan, S.-e. Jeon, N. Nagappan, B. Murphy and A. J. Ko, "Characterizing the differences between pre- and post- release versions of software," in *ICSE '11: Proceedings of the 33rd International Conference on Software Engineering* , 2011.

[44] R. Musson, J. Richards, D. Fisher, C. Bird, B. Bussone and S. Ganguly, "Leveraging the Crowd: How 48,000 Users Helped Improve Lync Performance," *IEEE Software,* vol. 30, no. 4, pp. 38-45, 2013.

[45] R. Kohavi, R. Longbotham, D. Sommerfield and R. M. Henne, "Controlled experiments on the web: survey and practical guide," *Data Mining and Knowledge Discovery,* vol. 18, no. 1, pp. 140-181, 2009.

[46] McKinsey Global Institute, *Big data: The next frontier for innovation, competition, and productivity,* http://www.mckinsey.com/insights/business_technology/big_data_the_next_frontier_for_innovation, 2011.

[47] T. Xie, N. Tillmann, J. d. Halleux and W. Schulte, "Future of developer testing: building quality in code," in *FoSER '10 Proceedings of the FSE/SDP workshop on Future of software engineering research*, 2010.

[48] M. Hewner, "Undergraduate conceptions of the field of computer science," in *ICER '13: Proceedings of the international ACM conference on International computing education research*, 2013.

[49] L. A. Sudol and C. Jaspan, "Analyzing the strength of undergraduate misconceptions about software engineering," in *ICER '10: Proceedings of the international workshop on Computing education research*, 2010.