Plumbing for Philosophers: The Operations of a Data Science Team

Joseph W. Clark
joseph.w.clark@asu.edu
Arizona State University

Anqi Xu
anqixu789@ufl.edu

Abstract

The title of *data scientist* applies to individuals with a truly rare mix of business domain knowledge, software development skills, and mathematical and statistical acumen. So rare, in fact, those organizations may find it wiser to develop data science *teams* than to invest in individual data science superstars, even if such superstars are available.

Prior research informs us about the characteristics of data scientists, and the tools they use, but scarcely looks "inside the black box" at how the work of data science breaks down into roles, tasks, and processes. Therefore, managers who may wish to create their own data science teams have little guidance as to team composition and the design of a workflow.

To remedy this gap in our knowledge, we conducted a thorough survey of practitioner-oriented literature, and conducted several structured interviews with members and managers of data science and analytics teams. We develop a framework of transformations grouped into two workflow "pipelines": ad hoc analysis and productization. In addition, we distinguish the knowledge areas of different roles on data science teams, including data scientists, data engineers, data wranglers, and data communicators.

The overall contribution of this research is to introduce a *transformational process* perspective on the operations of a data science or analytics team. This perspective offers several avenues for new research.

Keywords: Data Science Team, Data Science Roles, Data Analysis Workflow

Plumbing for Philosophers: The Operations of a Data Science Team

Completed Research Paper

Epigraph

> "The society which scorns excellence in plumbing as a humble activity and tolerates shoddiness in philosophy as an exalted activity will have neither good plumbing nor good philosophy: neither its pipes nor its theories will hold water." John W. Gardner

Introduction

There has been a huge amount of interest in the work of a *data scientist* since the term was coined in 2008 by D.J. Patil and Jeff Hammerbacher of LinkedIn and Facebook respectively, and particularly since a well-read *Harvard Business Review* article dubbed it "The Sexiest Job of the 21st Century" (Davenport & Patil 2012). Data scientists are said to require a combination of skills: mathematical and statistical skills to develop the right analyses and visualizations, computer programming skills to acquire and process unstructured data sets, and knowledge and interest in the business context, strategies and decisions. Davenport and Patil suggests we consider a data scientist "as a hybrid of data hacker, analyst, communicator, and trusted advisor." *InformationWeek*'s Mark Bregman writes that "A good data scientist is someone who has the right tools (math, programming, critical thinking), is self-sufficient (doesn't need someone else to implement his or her ideas) and has an interest in understanding the *context* in which the skills can be applied." (Bregman, 2013).

Unfortunately for managers, "[t]he combination is extremely powerful – and rare" (Davenport & Patil, 2012). Bregman (2013) dubs it "The Sexiest Job No One Has" and suggests looking for potential data scientists where you might not expect them – a marine biology PhD, or a baseball statistician, he suggests, might have that mix of skills. Gartner's Peter Sondergaard projected in 2012 that 4.4 million IT jobs globally, 1.9 million of them in the US, would be created to support big data by 2015. In the Gartner press release, Sondergaard minces no words – "There is not enough talent in the industry. Our public and private education systems are failing us. Therefore, only one-third of the IT jobs will be filled. Data experts will be a scarce, valuable commodity" (Gartner, 2012). *McKinsey Quarterly* has projected a shortfall of 140,000 to 190,000 data scientists, and 1.5 million data-savvy managers, by 2018 (Brown, Chui, & Manyika, 2011).

So, what are managers to do? Since the data scientist's mix of skills is so rare, it is likely to be hard to find and recognize a potential data scientist. A data scientist with a proven track record is going to be difficult and expensive to hire (Davenport & Patil, 2012), and by hiring such a superstar, the business exposes itself to the risk that its superstar will leave for another company "and all of a sudden, all that good work is lost" (Bertolucci, 2013). Davenport and Patil ponder whether firms should wait until universities catch up to the emerging job description and a "second generation of data scientists" arrives on the scene, but they do not venture an answer. By contrast, Bertolucci's interviewee, Lattice CEO Shashi Upadhyay, asserts that only two strategies make business sense: to outsource data science by consuming "big data apps", or to commit to building *data science teams*.

Our research begins with this insight: it is more practical for businesses to develop *teams* to do data science work, than to rely on individual star data scientists, *even if such superstars are available*. Unfortunately, almost nothing has been written down about how to form, manage, or empower data science teams to do this work. Therefore, small and medium-sized businesses without existing data science teams have no guidance in trying to create them. Much of the extant literature has focused on the outside of the black box, i.e. the sorts of skills and backgrounds data scientists have, rather than on the activities and operations that they perform. The goal of the present research is to address this gap by

making an initial survey of the field and by building up theoretical and practical frameworks for a team-based data science discipline.

A number of high-level research questions may be addressed by this work:

- How should data science teams be composed? How many people? What roles?
- What workflow should a data science team follow? In other words, what operational processes (or "data plumbing") should be set up?

Theory Development

Doing Data Science Work in Teams

Before we can understand how teams can do the work of a data scientist, we need to understand what sort of work, in fact, data scientists do. Tamm et al. (2013) articulate three "pathways to value from business analytics" based on the different roles played by *analytics professionals* (which they explicitly equate to Davenport & Patil's data scientists) and *analytics end-users*. The three pathways are:

- *Advisory services*, in which data scientists perform ad hoc analyses of structured or unstructured data, using any of the tools in their rich toolkit, prompted by business questions from decision makers.
- *Tool creation*, which includes the development of high quality business intelligence (BI) systems for decision makers, as well as the embedding of analytic capabilities into operational systems.
- *End-user analytics*, that is, the use of analytic tools by end users without the direct assistance of data scientists. This encompasses power users generating reports from the BI system as well as front-line employees using systems with embedded analytics.

Tamm also provides a table that specifically enumerates the roles played by a data scientist as (1) providing advice on unstructured problems, (2) providing advice on semi-structured problems, (3) supervising development, and (4) BI platform building. Taken together, their research suggests that data science teams need to do two types of work: first, to develop data sources and ad hoc analyses, and second, to work with the IT function to "productize" their analyses as analytical or operational information systems.

Data Science is Complex, Creative Work

Data science work is akin to other IS development work, particularly software development, but is also unique in that it includes the scientific, mathematical, and algorithmic work of cultivating data, building models, and generating analytical findings. Data scientists need to get their hands dirty with data, to "identify rich data sources, join them with other, potentially incomplete data sources, and clean the resulting set" (Davenport & Patil, 2012). They need to be "scientists" who are able to systematically formulate meaningful questions that can be answered with the available data (Leek, 2013). They apply "an integrated skill set spanning mathematics, machine learning, artificial intelligence, statistics, databases, and optimization" (Dhar, 2013). In addition, data scientists need to be hackers, able to build their own software for ad hoc processing of unique datasets: "it often falls to data scientists to fashion their own tools and even conduct academic-style research" (Davenport & Patil, 2012).

We know a number of things about complex, creative work like software development. Because the work is done under conditions of uncertainty and bounded rationality, the organization must be able to sense and adapt to exogenous change as well as an endogenously changing understanding of the problem (Overby et al, 2006). The problem solving organization requires some balance of *dynamic capabilities* for planned change, and *improvisational capabilities* for unplanned change, depending on the level and type of turbulence in the problem space (Pavlou & El Sawy, 2010). *Semi-structured* organizations which allow flexibility in problem-solving but are governed by simple rules outperform both unstructured and rigidly structured organizations (Brown & Eisenhardt, 1997).

We also know from that there is an operational discipline to supporting software development (Kim et al, 2013; Kniberg, 2011). Development work is creative but it is still possible to visualize the flow of work

from requirements gathering to task definition to coding, compiling, testing, and deployment. Bottlenecks and work pileups can be identified, and DevOps ("development operations") practices like automated regression testing and continuous deployment can be used to empower software development teams. (chromatic, 2003; Swartout, 2012)

Information Systems Development（ISD）Agility as Guiding Concept

Similar insights about information systems development (ISD) generally have led to the emergence of Agile methodologies such as Scrum and Extreme Programming (XP). Agile approaches vary, but common aspects are cross-functional teams, frequent iteration, team-level knowledge creation, and constant seeking of feedback from product owners, clients, and end users. Following practice, research has recently begun to catch up with a theoretical understanding of *why* these approaches work.

Conboy (2009) traces ISD agility back to two root concepts: flexibility and leanness. Although Lean aspect of agility is less understood and less frequently touted than the flexibility aspect, Austin and Devin (2009) make a powerful case for understanding agile ISD in relation to Lean manufacturing, as something analogous to *post-industrial production*. As rapid prototyping, just-in-time supply chains, reconfigurable assembly lines and other advances have enabled efficient, mass customization of physical goods, the Lean ISD practices encapsulated in methods like Scrum and XP have married responsiveness and adaptability to discipline and quality in software production. This marriage is possible because agile engineering and project management practices reduce two types of novelty costs: *exploration costs* (the cost of trying out design hypotheses) and *reconfiguration costs* (the cost of altering products and releasing new versions).

Austin and Devin give us the outlines of a contingency theory. Agile approaches are likely to be valuable when the value of customization (*novelty benefit*) is high, and when the approaches enable a reduction in novelty costs (exploration and reconfiguration costs). Thus it is that agile ISD has supplanted plan-driven ISD methods only in the past few years; before the development of modern development tools and DevOps, Agile approaches were less effective in reducing costs of exploration and reconfiguration in ISD.

Data Science is Not Exactly like Software Development

Undoubtedly much of what we have learned about ISD, particularly software development, over the past few decades is transferable to the data science domain, but there are important differences.

In software development, requirements can usually be documented as "user stories" which describe "features" to be added on to an evolving and growing "product", but these terms do not seem to fit the typical requirements structure for a data scientist, whose deliverables may be one-off reports and visualizations. Perhaps more akin to basic research, "analytical requirements by their very nature are accretive... users find it difficult to articulate future information needs beyond the 'next reports', because these needs are dependent upon the answer the 'next reports' will provide..." (Corr & Stagnitto, p.12).

An additional challenge for traditional IT managers may be a "project" mentality that celebrates the successful implementation of analytical technologies but does not follow up on the opportunity it might provide for rethinking the business: "Once the system goes live, no one pays any attention to figuring out how to use the information it generates to make better decisions or gain deeper – and perhaps unanticipated – insights into key aspects of the business." (Marchand & Peppard, 2013, pp. 105-106)

Finally, if we try to visualize the data science workflow from an operational perspective, it is clear that the "pipeline" is different. Instead of analyze-code-compile-test-deploy, there is implied in Tamm's (2013) classification a progression from ad hoc analysis to "productization" in which control is passed from the data scientists, to IT developers, and then to end users. Quality assurance and testing have different meanings, as two of the subjects we interviewed for this research remarked: "Business problems do not have a right answer, so they're difficult to test" (B). "For a regular piece of software, non-statistical software can be tested against a 'specification', but in this [data science] type of work you can't enumerate the conditions, and instead you have to make a statistical argument about the outcomes" (F). Furthermore, the feedback loops and decision points in data science work must differ from those in software development, which has no corresponding decision about when to "productize" a discovery.

In order to understand how data science teams will be similar to, and different from, software development teams, we need to investigate three additional research questions:

- How do (or should) data science teams discover, define, and work on requirements?
- What goals, values or principles are useful in understanding and evaluating a data science process?
- What operational workflow(s) or processes do (or should) data science teams follow?

We want to understand both the inside and outside of the "black box" of data science team work, so that we can not only describe data science teams but also inform managers and businesses who want to empower their teams to achieve maximum performance.

Initial Theoretical Framework

Based on the research literature reviewed above, we can begin to outline a framework for theorizing about how to create and support data science teams to create the maximum possible business value. Taking "business value from data science" as the outcome, then, we sketched out Figure 2, below.

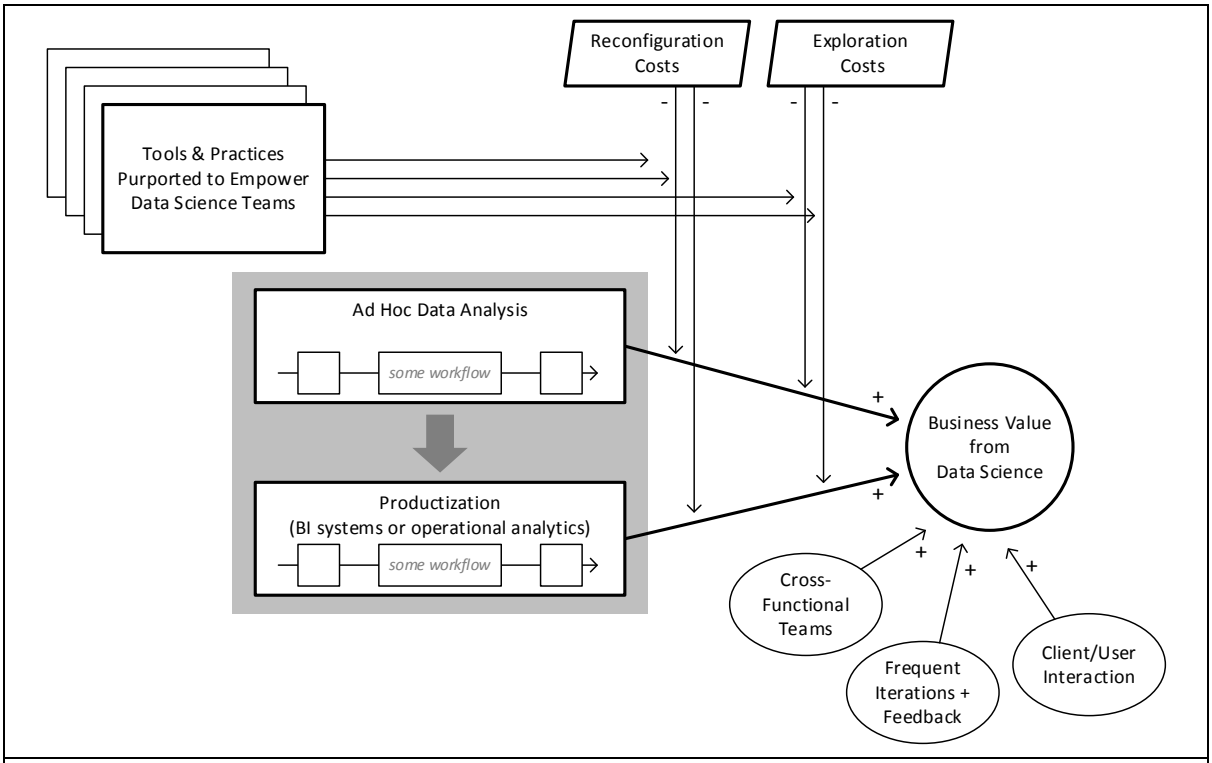

Figure 1. Initial Theoretical Framework

We posit that there are two general ways that data science teams produce value – first, by conducting unique and ad hoc analyses, and second, by developing analytic "products" such as BI applications and operational analytic functionality for end users. Ad hoc analyses *may* lead to later productization, or may stand alone as one-off chunks of value. Within each of these pathways to value there must be some operational workflow; identifying these flows is one goal of this research. Certain common features of agile ISD methods are likely to increase the business value produced, among them: use of cross-functional teams, frequent iteration or opportunities for feedback, and rich interaction with customers or end users.

In addition, we propose that each process is characterized by certain sources of reconfiguration costs and exploration costs. Each practice or tool that is purported to support data science teams can be evaluated, and its impact quantified, in terms of its potential to reduce these novelty costs. This is, then, a contingency model. Where the costs are not present, the practices that mitigate them are less relevant.

Where the costs are present, the relevance of a practice or tool can be determined by which cost(s) it reduces and by how much. Some practices and tools may conceivably be interchangeable.

Research Methods

As this study is the first major iteration in what will likely be a long-term project to understand the team-based discipline of data science, our goals were (1) to identify points of agreement and disagreement among practitioners and thought leaders about the operations of a data science team, and (2) to identify and develop important questions for future research. Our methods therefore have been a thorough survey of the leading publications from the field (books, articles, blogs, and webinars) and structured interviews with data science and analytics professionals. At the time of this writing, we had interviewed eleven experts in the USA, UK, India, and China, asking each one a battery of questions about activities, roles, processes, and challenges in their work. We analyzed our notes on the published literature and the interview data to identify meaningful themes and narrow our focus onto a few key questions.

We concluded the present round of research by iteratively developing the diagrams in Figs. 4-5 and a framework of transformations and roles, presented under "Synthesis", below. This has brought us to a meaningful stopping point at which we can identify targeted questions for further research.

Findings

We organize our findings in this section according to the major themes that frequently arose in the literature and were discussed by interviewees: team composition, principles, iteration, project types, and transformations. Interviewees are cited by letters (A) to (K) to preserve confidentiality. Table 1 presents non-identifying data about the interviewees:

Table 1. Interviewee Profiles

Code	Position	Industry	Company Size Range	Company Age (years)	Ownership	Country
A	Data Scientist	Public utility	1000-9999 employees	>100	Government & Private	USA
B	Sr. BI Manager	Online retailing	>10000	20	Public	USA
C	Data Dept. Leader	Financial services	> 10000	26	Public	China
D	Software Developer, Dept. of Data & Info.	Financial services	1000-9999	17	Public	China
E	Director of Data Analytics Center	Telecom	> 10000	20	Public	China
F	Test Manager	Automotive	> 10000	>100	Public	UK*
G	Modeling Analyst	Financial services, real estate, etc.	> 10000	26	Public	USA
H	Vice Director, Research Center	Information, Construction	10-99	2	Public	China
I	Data Scientist	Computer software & Consulting	1000-9999	2	Private	USA
J	Software Engineer, Machine Learning	Information Technology and Services: SaaS	10-99	5	Private	USA
K	Chief Data Scientist	Information Technology and Services: Consulting	10-99	1	Private	India

* Interviewee works in UK for a Germany-based company

Team Composition

Two issues are tied up in the question of how a data science team should be composed: the *roles* that must be filled, and the *people* who perform those roles. Jurney (2014), for example, identifies twelve roles on an "Agile Big Data team", but argues that the roles must be taken on by far fewer members. Generalists, such as data scientists who also do engineering, bring breadth of skills that makes agility possible.

We discovered a number of opinions on how a team should be composed. Interviewer (J), at a startup company, envisions that a mature data science team would include someone focused on acquiring data (a "data engineer"), a person focused on data visualization and dashboards, a team member to develop analytical models, and a product developer. Subject (C), a project manager at a startup, identified software development, data analysis, and data mining as the three key roles that every analytics team should cover. Others, such as (I) and (K), discussed frequent interaction with IS development or "tech" teams which were not part of the data science team, but fell into a different part of the organization chart.

Other interviews stressed the need for cooperation between business-savvy and technology-savvy teammates. For example, (D) argued that the team should combine business analysts with "technique support" personnel with statistics and computer science knowledge. From (E)'s point of view, as a service provider that does business analytics for other companies, the team should include as many people from the client company as from his own firm. The client would manage the infrastructure for data storage and processing while (E) focused on analytic development.

Finally, (J) believed that outsourcing could be an ideal way to carry out data science work; he would outsource data engineering tasks to India, modeling to experts in Silicon Valley, and do solution development and team management in-house. Larger companies, he suggests, could do all of this work themselves with an internal consulting model.

Principles

Unlike traditional IT projects that reduce the data and information processing burden from humans by automating structured tasks, analytics projects aim to do the opposite: get data *out* of the technological domain and into the human domain so it can be turned into actionable information (Marchand & Peppard, 2013). We sought themes in the data that would distinguish data science work from other problem solving work (e.g. software development) in terms of values and overarching objectives.

Managers must understand that in analytics and data science work, a greater degree of exploration and research is needed than in a traditional technology project (K). The solution that will result from an analytics project cannot always be envisioned at the beginning. Analytics projects should be conducted "like experiments: framing questions to which the data might provide answers, developing hypotheses, and iteratively experimenting to build knowledge and understanding" (Marchand & Peppard, 2013).

Analytics professionals must be allowed to re-frame requirements to tackle higher-order questions (i.e., going from "What should we decide?" to "How can we change the way that we make this decision?"). But some clarity of project scope is necessary: "Every engagement has a clear business goal... Deadlines can't be too long, because we don't want these turning into research projects" (I). When (I) deals with a client who has data but doesn't know what to do with it, his team first conducts a day-long work session to brainstorm possibilities, evaluate data sources, and identify prioritized use cases.

Iteration

Several of our interviewees (at least A, F, I, and J) use Agile development methodologies similar to Scrum, iterative methods that impose a rhythm of work with regular feedback loops. The rationale for iteration may be related to the fact that data science projects are more exploratory than regular software development. (F) notes that human intuition is limited in its ability to foresee all the factors that will affect analytical algorithms when they meet real-world data, and therefore iterative testing with real data is essential: "You only really understand the challenges once you're able to put data through the pipeline and see how it performs. You can't predict all the things that may affect your algorithms until you try it. Agile is a risk management approach. If you knew everything in advance, there'd be no case for Agile. These projects can only be done in an Agile way."

Similarly, (G) described macros that his team uses to automatically analyze the predictive models that they build. The work is a cycle of preparing data, building models, and running the macros to evaluate them. Aspirationally, (F) suggested that such a process could have dual purpose: "If you've got systems that can evaluate the performance of your algorithms, you also have a mechanism for reporting that. Agile is a communication method."

We found little discussion of any notion of "progress" other than additive: each iteration or release has more features than the last. That is the typical assumption in Agile software development methods: requirements are broken down into pieces that are each valuable on their own. These may be "user stories", i.e., features or use cases in a software project (Sims & Johnson, 2011). In an agile DW/BI project, according to Corr and Stagnitto (2013), the atomic unit of business value that should be delivered in each iteration is an individual star schema, with the ETL that populates it and a BI application.

There is little written about any qualitative evolution as projects mature. In one noteworthy exception, Jurney (2014) recommends that data scientists work deliberately upward, while iterating, on a "data-value pyramid" modeled on Maslow's famous hierarchy of needs. This model reflects a progression of discovery, learning, and ultimately application (Figure 2).

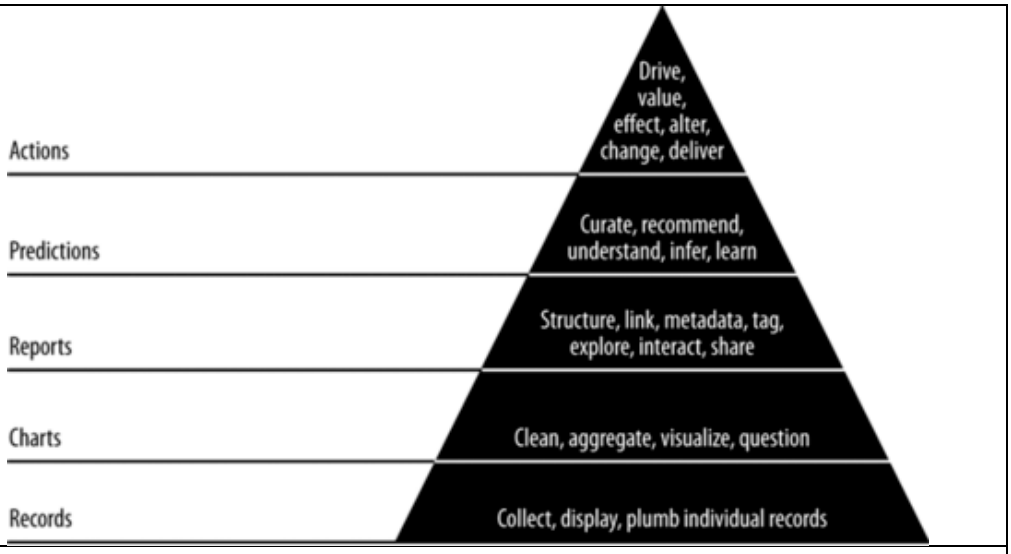

Figure 2. Data-Value Pyramid (Jurney, 2014)

Project Types

Many interviewees (including B, E, and K) talked about the distinction between projects where the output is analysis, for example, a one-time report or data set, and those where "the analysis is encased in technology and delivered as a product" or service (K). The need for "productization" may come from the desire to repeat a report on a weekly basis (B). The two types of projects may differ in time frame: (E) reports that analytic report development projects usually take 2-4 weeks while "system construction" projects can take 4-6 months. Some interviewees worked on only one of the two types, but others were involved in both.

The distinction between ad hoc analysis and productization is sometimes reflected in organizational structure. At a startup (K), data scientists are managed by a Chief Data Scientist and system builders are under a CTO. At a consulting firm (I), data scientists are grouped into teams according to the vertical (industry) that they serve, but engineers are in a different organization that does not have vertical silos.

A few other project classifications were mentioned by interviewees. (K) believes that as his startup grows, there may be a distinction between customer-facing work, done to satisfy customer requests, and exploratory research projects. Interviewee (D) distinguished between "data collection" projects and "data analytics" projects, the former more engineering focused and supportive of the latter.

Transformations

Holsapple et al (2014) identify several theoretical perspectives on business analytics: analytics as a movement, as a collection of practices and technologies, as a capability set, as a transformational process, as an activity type set, and as a decisional paradigm. Our interest in the "plumbing" of data science made us particularly open to transformational process perspectives, and several were evident.

Interviewee (F) synthesized his experience in multiple industries with a "conveyor belt" metaphor: "Innovation takes ideas and turns them into something usable. You get a really clever guy who can produce innovation, algorithms, then try to end up with some software either as a product or on a server somewhere that can be sold or used by the business." The links of the conveyor belt depend on the skills of the team members. One team may have mathematicians writing code to hand off to software engineers, and another may have downstream developers who receive academic papers as inputs and translate the ideas into code. "Regardless of the circumstances, your end goal is this 'conveyor belt'" (F).

Jurney (2014) describes a data processing flow based on a quick abstraction of the tools he uses in his own work; its generalizability is unclear, but it should certainly be valuable as a starting point for developing one's own workflow. See Figure 3.

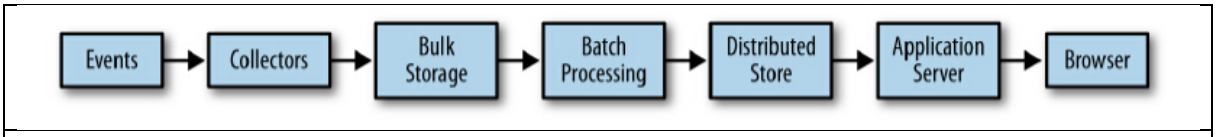

Figure 3. Flow of data processing in Agile Big Data (Jurney, 2014)

Kouzes et al (2009), coming from a scientific research point of view, conceive of the "data processing pipeline" as a three-stage progression from large datasets of low information density to small, information-rich result sets. In the first stage, data is collected or summarized in ways that reduce noise and prepare it for downstream analysts. In the second stage, analytical algorithms extract information, and in the third stage, researchers make the results presentable to users who can act on them.

Finally, a few interviewees described a loop back from the end of a productization process to the beginning: new software products themselves become data sources, and the developers and users of those systems have a duty to consider the importance of generating good, clean, accessible data (J).

Synthesis

To answer our research questions about roles, processes, and value production in data science work, we found it necessary to chart the transformational processes as a first step to synthesizing and integrating our findings. We identify two distinct but interconnected value chains: one describing ad hoc data analysis and the second describing data science productization.

The Analysis Pipeline

The Analysis Pipeline begins with data sources and outputs reports (or other types of "results") which are consumed by decision makers. Two intermediate steps are data preparation and model creation, but neither of these describes the whole process. The steps are discussed below Figure 4.

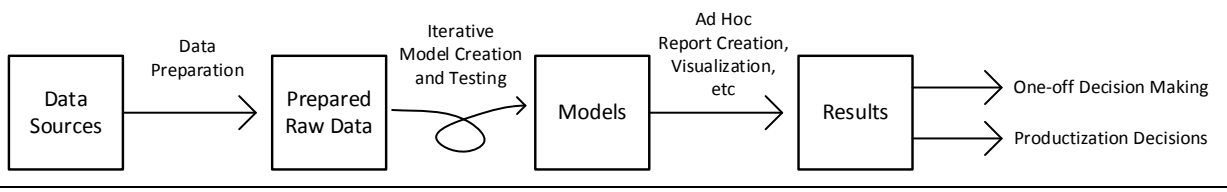

Figure 4. Analysis Pipeline.

Sources to Raw Data

Interviewee (D) states that "first, you need to translate the business problem into a data problem". Data preparation involves locating, evaluating, loading, cleansing, summarizing, filtering, and otherwise fashioning data "from the wild" into the raw material of data science. The preparation of data for analysis, known as "data wrangling", "data munging", or "data janitor work" is increasingly being recognized as a huge part of the work of data science teams. "Data scientists, according to interviews and expert estimates, spend from 50 percent to 80 percent of their time mired in this more mundane labor of collecting and preparing unruly digital data, before it can be explored for useful nuggets" (Lohr, 2014).

Often the Hadoop Distributed Filesystem (HDFS) serves the role of bulk data staging area where data from source systems can be gathered and made available to downstream transformations. "HDFS sets the standard for bulk storage, and without it, big data would not exist. There would be no cheap place to store vast amounts of data where it can be accessed with high I/O throughput for the kind of processing we do in Agile Big Data" (Jurney, 2014). Interviewee (K) confirmed that Hadoop is used in his workflow for data storage even when his team is not using MapReduce.

Raw Data to Models

Drawing on the prepared data, data science teams use algorithms and experiments to develop models. This is probably an iterative process following methodologies like CRISP (Provost & Fawcett, 2013), in which models are built and evaluated in search of continually better performance (G, F, I). There are many techniques for analytical model building (cf. Silver, 2012; Janert, 2011) and many specialties within this transformation (cf. discussion of "Data Scientist" role, below). A complete model is an artifact, possibly manifested as R, Python, or C++ code, and it may be used to create a report, or to engineer an analytical product, but model creation is not the whole picture of how data science produces value.

Models to Results (reports, visualizations, or experimental results)

In order to be of value to the business, models must be run and the results communicated, either to inform real business decisions or to get feedback from decision makers on the suitability of the results. These "results" may take many forms, depending on whether the consumer is a human or a machine (Li, 2014). Human managers may need data visualizations and explanatory reports, whereas machines may consume statistical information by running automated evaluations or tests. Jurney (2013) insists upon "publishing" of results, even intermediate results, in order to create business value and facilitate feedback loops. His process includes the use of a minimalistic web framework to quickly make results interactive and accessible to teammates and customers.

Results to One-off Decisions

These results may be used by decision makers in unique, non-repeating contexts. Typically in these cases the business is looking for a definitive answer to an important question, or is reacting to a breakthrough discovery by exploratory data scientists, and the situation will not repeat itself. The data science process has created business value if the decisions that result from this process are better on average than they would have been otherwise.

Results to Productization Requirements

Provost & Fawcett (2013) write that data science is interested in two types of decisions: "(1) decisions for which 'discoveries' need to be made within the data, and (2) decisions that repeat, especially at massive scale, and so decision-making can benefit from even small increases at decision-making accuracy based on data analysis." For the second type of decision to be improved by data science, the team needs to be prepared to transform their insights into data products and data systems. This leads to the second value chain, which we call the Productization Pipeline.

The Productization Pipeline

Productization occurs when some stakeholder decides to commit resources to developing systems that incorporate the data scientist's data, models, or algorithms.

The Productization Pipeline, akin to Tamm et al's (2013) "tool creation" pathway to value, describes the creation of systems that incorporate data, algorithms, or models, for use by others who are not part of the

data science team. It appears to draw on a blend of data science techniques and traditional software development disciplines, and it results in at least two distinctly different types of products: *analytical systems* such as DW/BI applications used by decision-makers, and *operational analytics*, a term which refers to statistics and data science features built into systems used in everyday work processes.

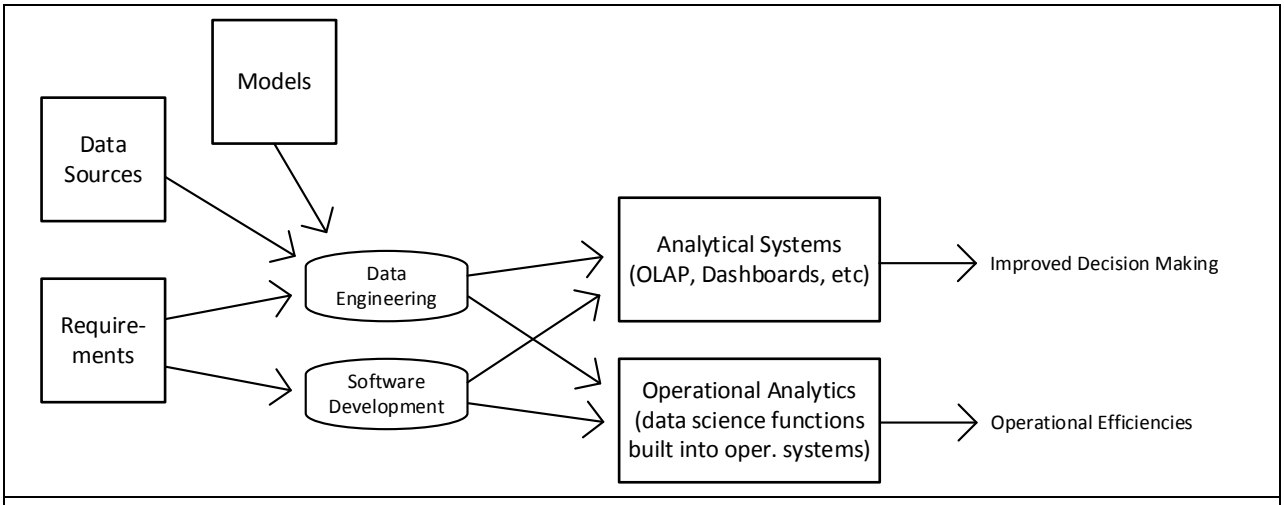

Figure 5. Productization Pipeline.

Analytical System Development

Analytical systems include data warehousing and business intelligence applications, OLAP tools, dashboards, automated reports, and other types of decision support systems (DSS) that empower managers and other decision makers to get the information they need, often in a self-service manner. Much has been written about how to develop analytical systems (Corr & Stagnitto, 2013; Kimball & Ross, 2013). What our findings show is that this development requires a combination of traditional software development capabilities, including project management, programming, and testing, on the one hand, and on the other hand "data engineering" expertise in the big data technologies such as Hadoop and MapReduce needed to scale up the data scientists' models to massive data sets and real time performance.

Operational System Development

Operational analytics is the embedding of analytical functionality into systems used by front line workers, customers, suppliers, and other nonexpert users. Common examples are recommendation engines such as those that might be seen on an e-commerce website, and fraud detection algorithms that exercise a constant vigilance over everyday operations in a bank or credit card system. These, too, require a combination of software development skills and data engineering technologies; perhaps weighted more toward the former than in analytical system development.

Systems to Business Value

Neither analytical systems nor operational analytics provide business value until they are used. Analytical systems lead to value creation when managers and others are able to use them in a self-service manner to get the information they need to make better decisions. This process is what Tamm et al (2013) call "end-user analytics". The value contribution of operational analytics, on the other hand, is in operational efficiency gains for front line workers, customers, suppliers, and others. If not explicitly measured, these gains may go unobserved, therefore, it may be wise to engineer these systems to produce high-quality data of their own (J).

Roles in Data Science Work

Our identification of the Analysis and Productization pipelines in data science work does not necessarily tell us about the number of people who should be on a data science team or what their responsibilities should be. At one extreme, one person could theoretically enact the entire pipeline. But even at the other

extreme, in which each type of task is assigned to one individual, some specialists may have multiple roles to play because of the interdependency between the two value chains.

In this section we strive to identify the *roles* (defined as distinct knowledge areas) on a team, but do not mean to imply that each role must be done by a different person, or that the team must have any particular structure. Teams may differ. As (F) commented, "The links of the conveyor belt depend on the skills of the guys in the team."

Data Scientist

The role that has received the most attention is that of the data scientist who creates models and algorithms and evaluates them to decide which ones work best. They are true "scientists" who propose hypotheses and design experiments to test them (Leek, 2013). Our interviewees identified specializations within this role.

> "[One specialization is] more 'applied', on the edge of machine learning, where the data scientist needs to know how to code things into a product. [Another type involves] trying to have inferences, with a lot of Bayesian estimation, analytics, statistics, that involve spending a lot of time developing the priors for the models... with models that are pretty complex sometimes. Then there's the type of data scientist who tends to focus more on communication, writing reports, showing data in ways that make sense to management. The fourth one is close to what we call a data engineer." (J)

Interviewee (K) also reported at least four specialties among data scientists. There are specialists in statistical modeling with stats backgrounds, who understand the biases in data, machine learning specialists who understand algorithms beyond the realm of statistical modeling, experts in the "big data" components such as Hadoop and MapReduce programming, and there may also be a place among data scientists for specialists in "optimization" from operations research and supply chain backgrounds.

The data scientist role has been much discussed, but not always distinguished from the other roles that we have identified, such as the data engineer and data wrangler. We aim to highlight this distinction by giving separate attention to the other roles, below.

Data Engineer

"There is a lot to data processing that is not data science" (Provost & Fawcett, 2013). Provost and Fawcett understand data engineering to include expertise at using data processing technologies, particularly "big data" technologies like Hadoop, HBase, and MongoDB, and recognize its importance in supporting data science. Data engineers are involved both in the ad hoc analytics pipeline and the productization pipeline. In the former, their expertise is needed in setting up and tearing down cloud servers, and preparing MapReduce jobs. In the latter, they may use these skills and more, possibly working with big data streams, event processing systems and other key parts of the back end of an analytical or operational system. Data warehousing experts may be considered a subset of data engineers.

Data Wrangler

If, as Lohr (2014) reports, data scientists spend upwards of 50% of their time on data preparation, it may be necessary to recognize this as a distinct role. Certainly it requires a different set of skills and routines from those used by the data scientist. "A good data munger excels at turning coffee into regular expressions and parsers, implemented in a high-level scripting language of choice (often Perl, Python, even Javascript). This is problem solving with programming, and quite different from statistics. An aspiration toward excellence... is rarely rewarded, and often punished." (Driscoll, 2009) A number of startups have been seen emerging in recent months that claim to automate or outsource some aspects of this data hygiene work, further reinforcing the view that this is a role distinct from data science. A good question for further research is how this part of the work differs from data engineering.

Data Communicator

Some of our findings indicate that there may be a need for professionals who specialize in data visualization and in communicating the results of analysis either within the team or between the team and business. Along these lines, Li (2014) draws a distinction between data scientists who produce "analytics for machines" and those who produce "analytics for humans". While the former are likely to be focused

on algorithms and productization, the latter need to be able "to tell a story from the data" and may need to choose simpler models even when more complex ones may be more accurate.

Subject (H) referred to this team member as a "model explorer" who would have primary responsibility for understanding the business's needs and exploring models *with* the business. Subjects (J) and (K) both felt the lack of a specialist in data visualization, both citing the inefficiency and fragmentation that occurs when each team member builds their own visualizations with different tools. It seems likely that a data science team would benefit from having someone specialized in, and focused on, visualizing and communicating data science results.

It appears that within this role there may be two parallel goals: first, to facilitate collaboration within the team, and second, to communicate the team's discoveries with non-specialists in the business. Driscoll (2009) reminds us that we can point to two broad types of data visualizations: exploratory data visualizations intended to facilitate a data analyst's own understanding of the data, and data visualization intended for communication, "visual narratives" that may require collaboration between data analysts and designers. (Driscoll, 2009). Further research may want to investigate how similar or different these two types of "data communication" are in practice.

Software Developer

The products of the Productization pipeline are "analysis delivered to technology" (K), so for this work the data science team must also include, or collaborate with, software developers who build systems, configure servers, develop graphical interfaces, and so on. Within this role are also included testers, project managers, scrum masters, and other roles already known to the ISD literature.

Product Owners

This term, common in the Agile field, refers to stakeholders who make decisions about what products or features to commit resources to. The importance of this role in our framework is that this role is responsible for deciding when a data science result is valuable enough to go into productization; these are the stakeholders who connect the two pipelines because they are responsible for ultimate business value. The product owner is the "single wringable neck" as far as higher-up stakeholders are concerned.

Decision Makers

By "decision makers" we refer those consumers of data science results, whether analytical/BI systems or one-off reports, who use them to make decisions. These are one class of "analytics end users" (Tamm et al, 2013). Decision makers can provide to data scientists the vital feedback they need during iterative model creation, and should also be consulted for feedback by the developers of analytical/BI systems.

Front Line Workers

Another class of end users are those workers who use operational systems. Indeed customers, suppliers, and others who interact with the organization may touch those systems as well.

Challenges for Data Plumbers

This research has resulted in a roadmap of the processes inside the black box of data science or big data analytics work, highlighting the workflows, roles, and transformations in ad hoc data science as well as analytics productization. This can serve as a starting point for analytics managers to attempt to create their own teams and processes, but as with software DevOps, every team and every job is different. This framework must be adapted to the unique contexts. The next goals for practice, and those researchers who are concerned with problems of practice, is to learn how to learn from this framework. The challenges are as follows:

1. How can we implement these processes and roles using existing employees in existing organizations?
2. Once implemented, how can we visualize the work and confirm what we think we know? Kanban and Kanban-like tools have been found to be very valuable in visualizing the flow of work in software development processes, and to identify efficiencies and operational bottlenecks (Kniberg, 2011). What would a Kanban system for data science look like?

3. How can this information, once made visible, be used to improve data science processes? In other words, what variations are possible, and which ones have an impact? Priorities might be investigating the opportunities for tighter or looser coupling between transformations, and the impact of Lean techniques like work in process (WIP) limits.

Managers may employ our initial theoretical framework to begin interpreting some of the prescriptions with which they are continually bombarded. Given a roadmap to the data science processes, and an operational mindset, they can identify the inefficiencies or bottlenecks that impede the smooth flow of work. These can be seen in terms of reconfiguration costs and exploration costs. Claims that are made by marketers or proponents of new tools and practices, now, may be evaluated in terms of *which costs* they enable the team to overcome. Different data science teams working on different problems will face different process challenges; managers who know which costs are currently limiting their teams (and which costs are not) are armed to decide which prescriptions are relevant to their unique contexts and which are not.

Discussion

In this research, with reference to Holsapple et al's (2014) "Unified Foundation for Business Analytics", we have risen to the challenge to take a "process of transformation" perspective that highlights the processes of how, when, and by who, data is transformed into business value. We have synthesized practical descriptions of workflows and practices in the field that may be used by managers as a starting point in designing their own data science teams. An important next step will be acquiring empirical feedback on how well they work in application. With the benefit of our findings, there are numerous questions we can now pose for research:

1. Which roles in the data science team are likely to be complementary? Does a data scientist become more productive, or less productive, if he is also a skilled data wrangler? Data communicator? Data engineer? Should data engineers also be software developers? And so on.
2. How does the performance of data science teams compare with that of superstar data scientists on a variety of measures? What happens when superstars are part of the team?
3. What is the nature of the data communicator role? Do good data communicators come from among data scientists, or do they have a different sort of background? Are business-facing data communicators very different from team-facing data visualization experts?
4. What is the typical range of data science team sizes? The typical mix of specialties?
5. What factors impact the size of a data science team and its composition? Some interviewees indicated that data science teams should differ by industry (B, I) or by company size (C, H, J).
6. What are the typical bottlenecks affecting analysis and productization pipelines? What remedies best fit each typical problem?
7. If there is a taxonomy of typical analytics or data science project types, how does it map to our operational roadmap? Which parts of the team, and which transformations, are engaged in each type of analytics project? How would the optimal team composition change?
8. What fraction of analytics managers take an operational ("plumbing") view of their data science related process? How many are aware of their bottlenecks? What effect would the introduction of workflow tools (e.g. Kanban) have on their ability to improve data science processes?
9. How can researchers better study the working processes of data science teams, at scale, without disrupting their work?

We hope that this work generates new research interest in the importance of effective processes and "plumbing" to make it possible for the much-lauded data science activity to generate value. By making these processes explicit, research can enable practicing managers to imitate and innovate upon them.

We are grateful to Michael Goul and Aaron Read for helpful feedback on working drafts of this paper.

References

Interviewees are referenced with letters (A) through (K) to preserve their anonymity. Quotations from (F) were paraphrased and checked with the interviewee due to a failure of recording. Quotations from (B), (C), (D), (E), and (H) were translated into English from Chinese by one of the authors.

Austin, R. D., & Devin, L. (2009). Research Commentary: Weighing the Benefits and Costs of Flexibility in Making Software: Toward a Contingency Theory of the Determinants of Development Process Design. *Information Systems Research, 20*(3), 462-477.

Bertolucci, J. (2013). Are You Recruiting a Data Scientist, or Unicorn? Retrieved from http://www.informationweek.com/big-data/big-data-analytics/are-you-recruiting-a-data-scientist-or-unicorn/d/d-id/899843

Bregman, M. (2013). Data Scientist: The Sexiest Job No One Has. Retrieved from http://www.informationweek.com/big-data/big-data-analytics/data-scientist-the-sexiest-job-no-one-has/d/d-id/1112832

Brown, B., Chui, M., & Manyika, J. (2011). Are you ready for the era of 'big data'? *McKinsey Quarterly, 2011*(4), 24-35.

Brown, S. L., & Eisenhardt, K. M. (1997). The Art of Continuous Change: Linking Complexity Theory and Time-Paced Evolution in Relentlessly Shifting Organizations. *Administrative Science Quarterly, 42*(1), 1-34.

chromatic. (2003). *Extreme Programming Pocket Guide*. Sebastopol, CA: O'Reilly Media.

Conboy, K. (2009). Agility from First Principles: Reconstructing the Concept of Agility in Information Systems Development. *Information Systems Research, 20*(3), 329-354.

Corr, L., & Stagnitto, J. (2013). *Agile Data Warehouse Design*. Leeds, UK: DecisionOne Press.

Davenport, T. H., & Patil, D. J. (2012). Data Scientist: The Sexiest Job Of the 21st Century. *Harvard Business Review, 90*(10), 70-76.

Dhar, V. (2013). Data Science and Prediction. *Communications of the ACM, 56*(12), 64-73.

Driscoll, M. E. (2009). the three sexy skills of data geeks. Retrieved from http://medriscoll.com/post/4740157098/the-three-sexy-skills-of-data-geeks

Gartner. (2012). Gartner Says Big Data Creates Big Jobs: 4.4. Million IT Jobs Globally to Support Big Data by 2015 [Press release]. Retrieved from http://www.gartner.com/newsroom/id/2207915

Holsapple, C., Lee-Post, A., & Pakath, R. (2014). A Unified Foundation for Business Analytics. *Decision Support Systems, 64*, 130-141.

Janert, P. K. (2011). *Data Analysis with Open Source Tools*. Sebastopol, CA: O'Reilly Media.

Jurney, R. (2014). *Agile Data Science*. Sebastopol, CA: O'Reilly Media.

Kim, G., Behr, K., & Spafford, G. (2013). *The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win*. Portland, OR: IT Revolution Press.

Kimball, R., & Ross, M. (2013). *The Data Warehouse Toolkit* (3rd ed.). Indianapolis, IN: John Wiley & Sons.

Kniberg, H. (2011). *Lean from the Trenches: Managing Large-Scale Projects with Kanban*: The Pragmatic Bookshelf.

Kouzes, R. T., Anderson, G. A., Elbert, S. T., Gorton, I., & Gracio, D. K. (2009). The Changing Paradigm of Data-Intensive Computing. *Computer, 42*(1), 26-34.

Leek, J. (2013). The key word in "Data Science" is not Data, it is Science. Retrieved from http://simplystatistics.org/2013/12/12/the-key-word-in-data-science-is-not-data-it-is-science/

Li, M. (2014). The Question to Ask Before Hiring a Data Scientist. Retrieved from http://blogs.hbr.org/2014/08/the-question-to-ask-before-hiring-a-data-scientist/

Lohr, S. (2014, August 17). For Big-Data Scientists, 'Janitor Work' is Key Hurdle to Insights. *The New York Times*.

Marchand, D. A., & Peppard, J. (2013). Why IT Fumbles Analytics. *Harvard Business Review, 91*(2), 104-112.

Overby, E., Bharadwaj, A., & Sambamurthy, V. (2006). Enterprise agility and the enabling role of information technology. *European Journal of Information Systems, 15*(2), 120-131.

Pavlou, P. A., & El Sawy, O. A. (2010). The "Third Hand": IT-Enabled Competitive Advantage in Turbulence Through Improvisational Capabilities. *Information Systems Research, 21*(3), 443-471.

Provost, F., & Fawcett, T. (2013). *Data Science for Business*. Sebastopol, CA: O'Reilly Media.

Silver, N. (2012). *The Signal and the Noise: Why So Many Predictions Fail - but Some Don't*. New York: The Penguin Press.

Sims, C., & Johnson, H. L. (2011). *The Elements of Scrum*. Foster City, CA: Dymaxicon.

Swartout, P. (2012). *Continuous Delivery and DevOps: A Quickstart Guide*. Birmingham, UK: Packt Publishing.

Tamm, T., Seddon, P., & Shanks, G. (2013). *Pathways to Value From Business Analytics*. Paper presented at ICIS 2013, Milan.