Achieving Lean Data Science Agility Via Data Driven Scrum

Jeffrey S. Saltz
Syracuse University
jsaltz@syr.edu

Nicholas Hotz
Indiana University
hotz@iu.edu

Alex Sutherland
Scrum Inc
alex.sutherland@scruminc.com

Abstract

This paper first reviews the concept of a lean data science project and defines four principles that a team should follow to achieve lean data science. It then describes a new team process framework called Data Driven Scrum (DDS) which enables lean data science project agility and addresses the key challenges that have been identified when using Scrum in a data science context. As compared to Scrum, DDS increases the focus in observing and analyzing the output of each iteration (i.e., each experiment). DDS also defines capability-based iterations (as compared to Scrum's time-based sprints). The paper then reports on how to integrate DDS with data science workflow frameworks such as CRISP-DM as well as on a pilot study of an organization that adopted DDS.

1. Introduction

Data science develops actionable insight from data by encompassing the entire life cycle of requirements, data collection, preparation, analysis, visualization, management and the preservation of large datasets [1]. This broad view embraces the notion that data science is more than just analytics in that it integrates a range of other disciplines including computer science, statistics, information management and big data engineering.

Unfortunately, most data science research has focused on the technical capabilities required for data science and has overlooked the topic of managing data science projects [2]. In fact, it has been noted that little research is available on the effectiveness of the different possible methodologies that are used by data science teams [3]. It has also been noted that data science teams might need to adapt and modify agile processes that have been used in other domains [4].

Furthermore, while *The Lean Startup* [5] describes how to iteratively validate or reject a sequence of lean testable hypotheses, which could map very well to how a data science team explores their data, there is no fully defined framework that supports this paradigm within a data science context.

The lack of a well-accepted data science process was demonstrated in a survey which found that 82% of the data scientists did not follow an explicit process; yet, 85% of the data scientists thought that their results would improve with a more systematic process methodology [6]. Hence, not surprisingly, it has been reported that project management is a key challenge for successfully executing data science projects and that a key reason many data science projects fail is not technical in nature, but rather, process oriented [7].

In an example of the challenge of delivering data science projects, Venture Beat reported that 87% of data science projects never make it into production [8]. John Akred, former CTO of Silicon Valley Data Science, summarized the challenge: "We've met a lot of data science teams that understand how to do the data science, but they don't have any real method of managing the data science project" [9].

Researchers have also noted the need for an improved process. For example, Cao's discussion of data science challenges and future directions [10] finds that one of the key challenges in analyzing data includes developing methodologies for data science teams. Angée [11] summarized the challenge by noting that it is important to use an appropriate process methodology, but which, if any, process is the most appropriate is not easy to know.

The rest of this paper first provides a review of key frameworks that could be used for data science projects and then explores the challenges of using those frameworks within a data science context. The paper next defines four lean principles for data science projects. Then, a new process that supports the four lean principles is described. A pilot study, where an organization adopts the new process, is then discussed. Finally, the conclusion is presented.

2 Potential Frameworks

This section reviews current frameworks that a team could leverage to execute a data science project. This includes approaches based on lean principles (e.g., Kanban), agile frameworks (e.g., Scrum), as well as life cycle approaches (e.g., CRISP-DM).

2.1 Lean

Lean focuses on maximizing value and minimizing waste. Many, such as Ahmed [3], have noted the key

URI: https://hdl.handle.net/10125/80218
978-0-9981331-5-7
(CC BY-NC-ND 4.0)

benefits of lean, such as reducing uncertainty by deferring commitment for future work (learning from an iteration to prioritize future work), reducing waste by focusing on small iterations (and adjusting after each iteration), and the ability to quickly try things.

Three books that have helped shape the use of Lean are summarized below. However, it should be noted that while each of these books defines a set of principles (for a group to become lean), how a team actually implements the principles is not defined.

2.2.1 Lean Thinking. Womack and Jones [12] describe a lean thinking approach that stresses continuous incremental improvement of a product (and/or process) as well as eliminating non-useful activities. Key concepts that define their view of lean include *Specify Value* (as defined by the ultimate customer), *Identify the Value Stream* (the actions needed to bring a "product" to the customer), *Define the Flow* (define value-creating small steps and eliminate non-essential steps), *Pull tasks* (get work from upstream steps, let the customer, or next step in the process, pull the product from you), and *Pursue Perfection* (continually improve the process - to reduce time, space, and cost).

2.2.2 Lean for Software Development. Poppendieck and Poppendieck [13] outline how lean concepts can be used for knowledge work, specifically software development. It provides seven key principles that teams should follow: *eliminate waste, build quality in, create knowledge*, *defer commitment, deliver as fast as possible, respect people*, and *optimize the whole*.

2.2.3 Lean Startup. Focusing on how to use lean concepts to help launch new companies, Ries [5] views lean as a culture / philosophy, with four key principles that teams should follow: *eliminate uncertainty* (iterate often, fail early), *develop an MVP* (Minimum Viable Product), *focus on "validated learning" (*via "Build-Measure-Learn"), and *work smarter* (not harder).

2.2 Kanban

One can implement a set of lean principles via Kanban. In fact, when using Kanban, each team is free to use any process framework that supports / encourages the key Kanban principles [14]. Kanban principles include *visualize the workflow*, *limit work-in-progress, measure and manage flow, make process policies explicit, improve collaboratively* and *implement feedback*.

One key strength of Kanban is that it visually represents work on a Kanban board, with work items flowing across the columns (or bins) of increasing work completion. This visualization allows all team members to see the state of every piece of work at any time [15]. A Kanban board typically starts with a 'to-do' column and ends with a 'done' column.

Another key strength is that Kanban aims to minimize work-in-progress (WIP), often with WIP limits that represent the maximum number of items that can be in each column at any given time. Minimizing WIP enables a lean approach (by focusing on reducing the time it takes to complete a task or user story) and also enables agility (since possible tasks are re-prioritized each time a new task starts). Kanban proponents note that Kanban improves project visibility, quality, team motivation, communication and collaboration [15].

2.3 Scrum

Scrum has become the most commonly-used agile approach, with over 12 million practitioners [16]. Software companies have most heavily adopted Scrum, but a wide variety of companies use it [17] for diverse purposes (e.g., National Public Radio uses it to create new programming, John Deere for new machinery development, and Saab for fighter jets).

In short, Scrum is an adaptive framework for "developing, delivering, and sustaining complex products" [18]. It divides a larger project into a series of mini-projects, called "sprints", each of a consistent and fixed length, typically one to four weeks long. Scrum teams have three roles: the product owner, the development team, and the scrum master.

Each sprint starts with a sprint planning meeting where the product owner explains the top items from the product backlog, which is an ordered list of product development ideas. The development team forecasts what items from the product backlog they can deliver by the end of the sprint and then makes a sprint plan to develop a product increment that includes the selected backlog items. During a sprint, the team coordinates closely and holds daily meetings. At the end of each sprint, the team demonstrates the new product increment to stakeholders and solicits feedback during the sprint review. This increment should be potentially releasable and meet the predefined definition of done. To close a sprint, the team inspects itself and plans for how it can improve in the next sprint (during the sprint retrospective). Throughout the process, the scrum master serves as a coach to help everyone effectively implement Scrum [18].

2.4 Data Science Life Cycle Frameworks

In addition to process collaboration frameworks like Scrum, there are data science life cycles that detail the steps of a data science project. Three common life cycle frameworks are reviewed below.

2.4.1 CRISP-DM. Since its introduction in the 1990s, CRISP-DM (Cross Industry Standard Process for Data Mining) [19] has been consistently the most frequently used life cycle framework for Knowledge Discovery in Database (KDD) projects, and more recently for data science projects [20]. CRISP-DM describes six major iterative phases (*business understanding, data understanding, data preparation, modeling, evaluation and deployment*). Typically, when using this framework, the team progresses through the different phases as they deem appropriate. As needed, the team can "loop back" to a previous phase (ex. more data preparation), and in general, can define milestones they think are useful.

2.4.2 OSEMN. A simpler and more recent data science life cycle, which was described by Mason and Wiggins in 2010, is OSEMN [21]. This workflow consists of five phases (*Obtain, Scrub / clean**,** Explore / visualize, Model and iNterpret*). OSEMN is similar to CRISP-DM, in that the focus is on the phases of work to be done, not on how the team should coordinate that work. Furthermore, while OSEMN focuses on the key tasks of doing data science, OSEMN skips the business and data understanding initial phases of CRISP-DM as well as the deployment of the analytics.

2.4.3 Team Data Science Process (TDSP). Microsoft launched TDSP as "an agile, iterative data science methodology to deliver predictive analytics solutions and intelligent applications efficiently" [22]. Its core project life cycle is like CRISP-DM and includes five iterative stages (*Business Understanding, Data Acquisition and Understanding, Modeling, Deployment,* and *Customer Acceptance*).

As one can see, compared to CRISP-DM, TDSP defines a 'customer acceptance' phase, which acknowledges the project is not done until there is an analysis of that deployment. The framework further expands CRISP-DM by defining four roles (*group manager, team lead, project lead* and *individual contributor*) [22].

In addition, TDSP can optionally be integrated with Scrum. In this use case, the TDSP life cycle is combined with Scrum sprints, with each sprint designed to follow a scrum-like framework.

3 Evaluating the Frameworks

This section briefly highlights some of the potential challenges a team might encounter when using these frameworks within a data science context.

3.1 Lean Challenges

Despite the benefits of using a lean approach, the key challenge in a team trying to be lean is that there is no defined set of processes to follow, but rather, just key principles to follow. While this lack of process structure can be a strength (since the lack of a specified process definition allows teams to implement lean concepts within existing organizational practices), the lack of process definition also means that every team needs to determine how to achieve the lean principles. In other words, a team that wants to be lean needs to figure out its own processes and artifacts. This makes adoption more difficult, and it also means that best practices are not shared, and that common pitfalls can easily be repeated.

Only one paper was identified that focused on lean data science. That research explored combining lean concepts with CRISP-DM [3]. Their framework defined seven steps across three main phases, specifically, the *business phase* (work discovery, analytical approach), the *data phase* (data resources, data preparation), and the *product phase* (build MVP, measure value, learn & update). After the seventh step, the product gets deployed, or the team loops back to the data phase. While this approach did introduce the concept of an MVP as well as lean's build-measure-learn concept, it did not address how the team should actually collaborate together (in other words, it was focused on the steps of the process).

3.2 Kanban Challenges

Kanban has had some use within a data science context. For example, to address the challenges of using Scrum sprints (discussed in the next section), one data science team migrated from Scrum to Kanban [23]. However, similar to the challenges with using lean, Kanban's lack of process definition means that teams need to determine how to define roles, meetings and artifacts to best implement Kanban. In other words, the fact that Kanban does not explicitly specify a process framework suggests that Kanban needs to be supported by additional practices [24].

This lack of process definition also explains why teams that use Kanban note that "Kanban requires integration with existing agile techniques, which can be complicated, expensive, and time-consuming" [3]. Hence, despite the benefits of using Kanban, there are also challenges to using Kanban. In general, these challenges include the lack of organizational support, culture conflicts, the lack of training, and the misunderstanding of key concepts.

3.3 Scrum Challenges

Scrum has also started to be used for data science projects, such as the work described by Lawler & Joseph [25], who note the use of Scrum in a financial analytics context. However, while that research noted the use of Scrum, it did not explore Scrum's strengths or weaknesses within a data science context.

One challenge of using Scrum sprints within a data science context is that task estimation is unreliable [23]. In other words, if the team cannot accurately estimate task duration (e.g., how long a specific exploratory analysis will take), the concept of a sprint, and what can get done within a sprint is problematic.

Another key challenge is that Scrum's fixed-length sprints can be problematic in that even if a team could estimate how long a specific analysis might take, having a fixed length sprint does not allow smaller (or longer) logical chunks of work to be completed and analyzed coherently. Sometimes, it might lead the team to define an iteration to include unrelated work items to "fill up" the sprint. Moreover, feedback could be unnecessarily delayed if the team waits until the end of the sprint to demo work even if it is ready for review before the sprint's end. On the other hand, if the sprint is too short, the team might compromise quality to get something 'done' in time. Moreover, many data science tasks such as data collection or allowing a model to run long enough so that its performance can be measured might require time that extends beyond a sprint.

These two challenges were implicitly noted in an effort to integrate Scrum with CRISP-DM [26]. The proposed framework stated that sprints should only be used in the modelling phase, since in the other project phases it would be more difficult to use Scrum's fixed length sprints. In short, the paper reinforced the challenge of using sprints for data science efforts. However, the research did note that feedback after each iteration would "allow the team to adjust the work in progress dynamically because it enables frequent interactions among team members and provides regular feedback loops [with stakeholders]".

3.4 Workflow Challenges

CRISP-DM and OSEMN both provide an overall set of guidelines of what should be done during a data science project. These guidelines, while helpful, need to be integrated within a framework that can structure *how* the team explores and iterates. In short, these frameworks ignore the broader implications of team coordination, communication and prioritization. Hence, they are helpful to describe *what* to do, but not *how* to do it.

For example, while CRISP-DM has the concept of "looping back" to iterate through one or more of the different CRISP-DM phases, there is no defined process of how the team should know if/when to loop back. Moreover, since TDSP can leverage and integrate the concept of Scrum sprints, Scrum's sprint challenges previously mentioned are applicable to TDSP.

Hence, it is not surprising that while CRISP-DM is the defacto process standard [20], it has been noted that CRISP-DM is limited when working with Machine Learning / Data Science efforts [3], and that an increasing number of teams create their own methods. Thus, while useful in some contexts, these workflow approaches do not fully address how a team should iterate through a series of analyses (experiments) to better understand the data and provide actionable insight.

To address these challenges, in the next section, we explore four lean data science principles. Then, in the following section, we describe Data Driven Scrum, a new framework that can help data science teams implement these principles.

4 Principles for Lean Data Science

Table 1 shows the four key principles to achieve lean data science. The lean data science principles are analogous to the principles in the prior lean efforts. However, the principles are more focused, in that they are more specific in how they can be used within a data science context. Below we describe each of the lean data science principles.

Lean Data Science	Lean Startup	Lean Software Development	Lean Thinking
Understand the Situation	Eliminate Uncertainty (Iterate often, fail early)	Eliminate waste	Understand Customer Needs
Iterate Often (minimum viable increments)	Develop an MVP (Minimum Viable Product)	Defer Commitment, Deliver as Fast as Possible	Limit Work in Process
Validate ("Create-Observe-Analyze")	Validate ("Build-Measure-Learn")	Optimize the Whole	Continue to Streamline Activities
Continuously Improve	Work Smarter Not Harder	Empower the Team	Continue to Streamline Activities

Table 1: Principles for Lean Data Science

4.1 Understand the Situation

Teams should work to ensure that the organizational challenge or opportunity is fully understood. This includes the goal of the project, how a predictive model would be used, how the project will measure success, and what data might be available. Note that some of the understanding can come before the start of doing iterations, but additional insight would be incrementally added during project iterations.

4.2 Iterate Often

Teams can achieve lean agility via executing a sequence of iterative experimentation and adaptation cycles**.** Each iteration should be a Minimum Viable Product or Minimum Viable Increment (MVP/MVI) to an existing product / analysis.

The goal of such cycles should be to have an idea or experiment in mind, to build it, observe the outcome, and then analyze those observations to create the next idea or experiment. Going from an initial idea, through implementation, and the analysis of the results should be the basis for an iteration. The completion of the empirical process should mark the end of an iteration (not a predetermined number of elapsed hours). Note that the end of an iteration is *not* when the team completes their MVP/MVI, but after the iteration has been analyzed and then validated by the sponsor/partner.

Teams should focus on maximizing the number of iterations (experiments) that they can achieve, weighted by the value of each experiment / item to be explored. This encourages teams to naturally focus on their process efficiency.

4.3 Validate the Iteration

The team should understand the value (or lack of value) of each iteration - the MVP/MVI. In fact, each iteration should be viewed as validating or rejecting a specific lean hypothesis. Hence, an iteration is defined by the following steps:

1. ***Create***: A thing or set of things that will be created, put into use with a hypothesis about what will happen.
2. ***Observe***: A set of observable outcomes that will be measured (and any work that is needed to facilitate that measurement).
3. ***Analyze***: Analyze those observables and create a plan for the next iteration

The create, observe, analyze process is similar to "build, measure, learn" from *The Lean Startup* [5], but with an emphasis on ensuring that the work that is required for data collection and analysis is directly incorporated into the team's tasks for a given iteration. We use "analyze" rather than "learn" to indicate that the team should focus on the output of the iteration, which is different from other learning that might occur during an iteration, which is incorporated in the principle of continuous improvement.

4.4 Continuously Improve

The team should aim to improve both the analysis and the process the team uses to create current / future work. To improve the solution, the team should meet to review its iteration results. This review should foster conversation with respect to recently completed iterations and the observations and analysis that the team has generated regarding the performance of those completed iteration(s).

To help improve the team's process, the team should meet at regular intervals (ex. once a month), to inspect, reflect and adapt the team's process. In the spirit of continuous improvement, the team comes together to discuss what is and is not working with the current process and associated technical practices.

5 The DDS Framework

This section explores Data Driven Scrum (DDS), which is an adaptation of Structured Kanban Iterations [27]. DDS integrates many of Scrum's key concepts within a Kanban set of principles, but also addresses some of the key challenges data science teams encounter when using Scrum in a data science context.

First, the three key tenets of DDS are explored. Next, DDS's roles, artifacts and meetings are summarized. Finally, this section ends by reviewing how to integrate a data science life cycle with DDS.

5.1 Key DDS Tenets

To achieve the lean data science principles, DDS defines three key tenets:

5.1.1 Define capability-based iterations. It sometimes makes sense to have an iteration that lasts one day (e.g., for a specific exploratory analysis), and other times, it might make sense for an iteration to last three weeks (e.g., to acquire / clean data). Independent of the duration of an iteration, the goal of an iteration should be to allow logical chunks of work to be released in a coherent fashion.

5.1.2 Decouple meetings from iterations. Since an iteration could be short and of varying length, meetings (such as a retrospective to improve the team's process) should be based on a logical time-based window, not linked to each iteration.

5.1.3 Create high-level item estimations. In many situations, defining an explicit timeline for an analysis is difficult, so generating detailed task estimations will provide misleading (i.e., often inaccurate) information. Instead, high-level effort estimates (e.g., "T-Shirt" sizing - S, M, L, XL) should be generated and used to help prioritize future iterations (but not estimate how long an iteration will take).

5.2 DDS Roles, Artifacts and Events/Meetings

The overall flow of work is shown in Figure 1 and described in the rest of this section.

5.2.1 Roles. Each DDS team is a group of three to nine people, one of whom is the ***product owner***. Similar to Scrum, the product owner in DDS is the empowered central point of product leadership – the person who decides which features and functionality to build, the order in which to build them, and what aspects of them to observe and analyze. Meanwhile, the ***process expert*** is similar to the Scrum Master role and acts as a coach, facilitator, impediment remover as well as helping everyone involved understand and embrace DDS. Both the product owner and the process expert are part of the DDS Team and may contribute to creating, observing and analyzing throughout an iteration. Finally, the DDS team should be a diverse, cross-functional collection of people that have all the skills needed to create the desired product/analysis.

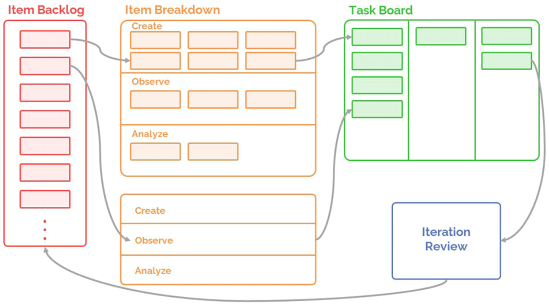

Figure 1: Flow of work in DDS

5.2.2 Artifacts. A ***Product Backlog Item (PBI)*** may take a variety of forms such as "testable hypotheses" as popularized by XP and Lean or "user stories" which is typically used within a Scrum team. Each PBI should include at least one thing to create, one thing to observe and one thing to analyze. The ***Product Backlog*** is a prioritized list of PBIs (i.e., work to be done). The product owner, with input from the stakeholders and the other team members, is responsible for maintaining the product backlog, which evolves and changes throughout the project. The ***Task Board*** is a visual representation of the work items currently in progress.

5.2.3 Meetings / Events. An ***Iteration*** is a collection of one or more product backlog items that are broken down into tasks, that together comprise a single testable experiment, which is then observed and analyzed. The team collectively strives to complete the iteration as soon as possible. If an iteration is waiting for observations (or analysis), then the team should start the next iteration. The ***Daily Meeting*** occurs each day, when the team meets for a 15-minute inspect-and-adapt activity. An important goal of this meeting is to help a self-organizing team better manage the flow of its work (ex. helping a team member get past an issue). Just as with Scrum Daily meetings, a common approach for conducting this meeting is for team members to share with each other what they did yesterday, what they are planning to do today, and identify any obstacles they are facing. The ***Iteration Review*** occurs on a regular and repeating calendar-based basis (e.g., weekly) because it is not practical to schedule these meetings after each variable-length iteration or on an ad hoc basis. The review could help identify potential features, metrics and experiments for future iterations. Furthermore, during this meeting, the product owner should reprioritize potential future iterations (since, for example, the insights gained might suggest a change in item priority or the creation of new items). Finally, the ***Retrospective*** also occurs at regular intervals (e.g., once a month) and provides time to inspect and refine the process, in that the team comes together to discuss what is and is not working with the current process. At the end of a retrospective, the team should have committed to a practical number of process improvement actions that it will implement.

5.2.4 Additional team activities. In addition to the DDS team working on one or more iterations, the team also spends time ***refining the PBI***. As part of the refinement process, the team provides an estimate of the effort for completing different items. This relative level of effort estimation could be a T-shirt sizing or be a number representing relative effort. The team uses this level of effort to help prioritize backlog items, but not to define what is part of an iteration.

5.2.5 Starting an iteration. When the team has capacity to start a new iteration (e.g., when an iteration has been completed, or when an iteration observation does not require full-time focus), the team reviews the prioritized backlog items and defines the next iteration by selecting the top item(s) that will now be the team's focus. Note that since the iteration is capability-based, and is the minimally viable set of items that can deliver value, the estimation is used to help prioritize items, not determine how many items should be included in an iteration (e.g., if two items deliver the same value but one is deemed a "small" effort and one is a "large" effort, the team might select the smaller level of effort item). Since shorter iterations tighten the feedback loop, combining multiple items into a single iteration is generally only desirable in the case that the associated hypothesis or observable data overlap.

5.3. Integrating a life cycle within DDS

This section explores how to integrate DDS with a life cycle framework (e.g., CRISP-DM).

5.3.1 Vertical Slicing. As shown in Figure 2, which is an example of a project trying to reduce customer churn, when a team uses vertical slicing, the team focuses on quickly delivering one vertical slice, going through the entire life cycle for that feature.

In considering CRISP-DM's six phases, with vertical slicing, one could execute a project by:

1. Develop a *high-level project roadmap* for the entire system with detailed next steps for *only* the initial model for the first deliverable.
2. Collect and analyze *enough* data for *only* the first deliverable.
3. Clean, integrate, and format *the most promising* data for *only* that first deliverable.
4. Develop a *basic,* model for *only* the first deliverable.
5. Evaluate the results of *only* the first deliverable.

Then, based on stakeholder needs, the next step could be to develop an automated system for the first model, or improve the first model (possibly with new data sources), or develop a model for the second deliverable. In other words, the first iteration provides just enough value to proceed to a decision point that leads to multiple possible paths. Numerous, small, vertical slices align the data science team's work with business value.

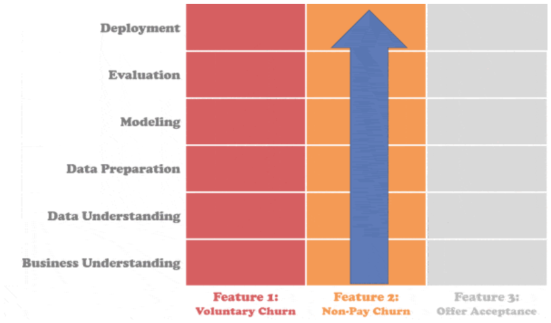

Figure 2: Vertical Slicing

By providing small increments, the stakeholders get value sooner and can provide feedback at multiple checkpoints. The team feeds this information back into the fungible project plan to possibly short-circuit non-value-add work or uncover other value streams that can be added into the future project phases.

5.3.2 Horizontal Slicing. As shown in Figure 3, if a team horizontally slices the work, it completes each major phase of the project in its entirety before moving onto the next phase. So, a team working horizontally would first complete the business understanding of all the use cases, then proceed to data understanding, and then so forth. The team goes back to a lower horizontal layer only if critically needed. Therefore, the team typically delivers the project as a "big bang" at the end of the project (i.e., at the end of the deployment phase).

In considering CRISP-DM's six phases, one could execute a project by:

1. Developing a *comprehensive* project plan based on an in-depth business understanding of *all three* deliverables
2. Understanding and collecting *(*nearly) all the relevant data
3. Cleaning, formatting, integrating (nearly) all the relevant data
4. Developing the *best possible* version(s) of *each of the three* models within generous time constraints
5. Evaluating the results of *all three* models
6. Deploying an application with an automated scoring pipeline for the *entire* system

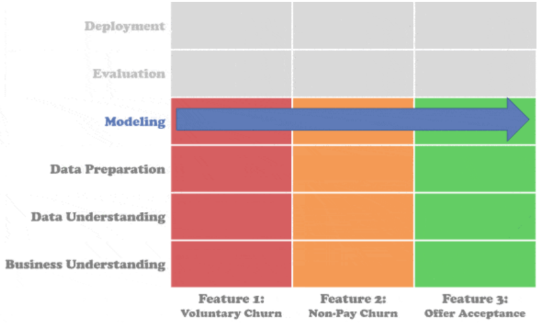

Figure 3: Horizontal Slicing

There are multiple problems with horizontally sliced projects. First, except for the last slice (deployment), each horizontal slice provides intermediate work that does not directly provide stakeholder value. Second, meaningful stakeholder feedback is difficult to obtain since deployment occurs once, at the end of the project. This increases the risk that the project might need significant rework after it is "delivered". Finally, model evaluation is deferred, which limits the data scientists' ability to understand whether they're on the right path.

5.3.3 Selecting and using a slicing approach. As implied by the strengths and weaknesses of each approach, vertical slicing is typically more appropriate. However, there are times when a hybrid approach is appropriate. For example, a team might use a horizontal slice to do an iteration that focuses on business understanding, and then do vertical slices, where iterations focus on the other phases (i.e., each future iteration would include data understanding, data preparation, modeling, evaluation and deployment).

Also, with respect to DDS's create, observe, and analyze, both of these approaches are focused on the create aspect of an iteration. It is after something is created that the team should observe and then analyze that creation. So, for example, if the team horizontally slices an iteration (which is focused on business understanding), the team might observe via the description of the desired organizational impact of the project, and then the analyze step would be the business sponsors reviewing that description. For a vertical slice (e.g., exploring the creation and use of a new model), the observe task might be collecting information on how a model is performing, and the analyze task could be to review those model outcomes.

6. Pilot Study

This section reports on an organization adapting their data science process to use the DDS framework. We focused on two key research questions:

RQ1: Based on the knowledge gained with respect to data science processes and DDS, will the team understand the key lean agile concepts (as it relates to data science)?

RQ2: Based on the knowledge gained with respect to how to improve their process, how will the team adapt / refine their process?

6.1 Organizational Background Context

The organization is a small (~30 person) consulting company based in Mexico. The company uses "artificial intelligence (AI) models to solve business problems for medium and large companies". Their clients come from a range of industries, including retail, telecommunications, manufacturing, hospitality and supply chain logistics. They use AI to solve challenges such as inventory optimization, pricing and promotion, sales forecasting, and labor planning. In short, like many AI/data science consulting organizations, they apply Machine Learning (ML), Deep Learning (DL), Bayesian Methods and Evolutionary Algorithms to tackle problems with clear goals and ROI definitions.

6.2 Previous Project Management Approach

Before using DDS, the organization managed data science projects using a waterfall approach. That is to say, they first defined the requirements, then collected the data, then did the modeling, etc. A key reason for their use of that approach was that their clients wanted to know the cost and timeline of the project upfront. As a result, as noted by the manager in charge of their data science efforts, "project proposals related to complex models included specific solutions to the problem and described specific results without having full knowledge of the data available". Perhaps not surprisingly, this led to many execution challenges for their data science projects (e.g., missed project deadlines, teams significantly overworked, and a feeling that the results "could have been better").

According to the data science team manager, to address these challenges, the team "tried different management approaches: daily meetings, weekly meetings, status reports, Gantt status reports, and many other approaches, but nothing really improved our process and minimized our project execution issues". In other words, none of the previous approaches were useful.

In terms of team communication, everyone tried their best to have open and frequent communication, but it was not done in an organized way, and as a result, "stakeholders gave feedback throughout the project to different members of the data science team". However, this feedback was hard to properly integrate into the project. This was also partially because the roles of DS team members were assigned within the DS team but not socialized with the rest of the stakeholders.

6.3 Data Science Process / DDS Training

To improve their data science process, the senior manager in charge of their data science efforts received data science process training from one of the researchers of this study. This training included key concepts such as (1) what is lean agility and why is it useful within data science projects, (2) a review of some typical life cycle frameworks (e.g., CRISP-DM, TDSP) and why are they useful, (3) an exploration of some typical coordination frameworks (e.g., Scrum, Kanban) and their strengths and weaknesses, and (4) an overview of Data Driven Scrum and how it could be used within their organization.

6.4 The Updated Data Science Team Process

Based on this new insight into how to better execute data science projects, the manager decided to use DDS, and led an internal effort to refine their current process for analytics and data science developments in projects.

6.4.1 Software development vs data science. The first area of agreement within their team was that their software development efforts were different from their data science efforts, in that their data science projects had more uncertainty in many areas. These projects had more ambiguous requirements, increased difficulty in measuring project success, time estimation uncertainty (e.g., in data cleansing and model development), more uncertainty with respect to the quality of the available data (or even knowing if the data was relevant for the desired predictive model), and uncertainty if their data science efforts could even improve the current state of their client's situation. This suggested that their data science projects should have a different process (than their software development team process).

6.4.2 Process guidelines and principles. The next area of agreement within the team was with respect to their interpretation of the lean agile principles, which they documented as:

- Consider all stages of the project
- Focus on small incremental improvements

- Focus on value-generating activities
- Improve communication and collaboration
- Ensure work is useful for the customer
- Change how the customer does things
- Communicate progress to all stakeholders

As can be seen by these statements, the team internalized the key aspects of a lean agile approach as discussed in *Section 4.1*.

6.4.3 Roles. With respect to roles, the organization more explicitly defined different roles, and then assigned specific roles to the project team according to the characteristics of the project. The roles included the Project Leader / Director, Data Engineer and Architect, Data Scientist, Data Analyst, Process Expert, and a Product Owner. A person could have multiple roles, but the roles needed to be clearly defined at the start of the project.

6.4.4 Project Phases. Projects were defined in four phases:

1. ***Business understanding / project proposal*** – explore the business problem and collectively create an initial list of possible experiments / iterations (i.e., item backlog), which are prioritized by the product owner (via estimated potential business impact). In addition, this phase also makes sure that the customer/client understands the challenges of the project (so their expectations are aligned with what might be possible).
2. ***Project preparation / launch*** – agree to start the project and assign team members / roles.
3. ***Project execution (multiple iterations)*** – divide the project into subsets of work ordered by priority, using the DDS framework. Each iteration consists of the life cycle phases after business understanding (i.e., acquire / prep data, analyze / model, evaluate / test, deploy / monitor). The goal is that each iteration will deliver incremental results to the client, and in each iteration, the team validates the scope and objective of the project and the value of that iteration (via the analyze and observe steps of DDS).
4. ***Knowledge management for retrospective*** – as iterations start to generate insights, document as appropriate, and explore how to improve their current process. This focus produces both internal and customer insights and documentation.

Hence, their new process is a combination of horizontal slicing (for the business understanding phase), followed by vertical slicing for their project execution (i.e., all the other phases of their project life cycle). Note that their project preparation / launch was focused on resource allocation and team roles, which is typically not part of current data science life cycles. Finally, note that their knowledge management phase focuses on improving future projects not just from a process perspective but also from a technical (e.g., modeling) perspective.

7. Discussion and Conclusion

7.1 Comparing DDS to Kanban and Scrum

DDS adheres to the Kanban principles (e.g., there is a Kanban board to track an iteration, teams need to limit WIP, and work items flow across the board). However, the framework provides more structure than defined by Kanban, such as defined iterations as well as a more defined framework (ex. roles and meetings). These more clearly defined processes leverage agile best practices which can help teams implement a lean process in a more consistent and repeatable manner.

DDS can also be viewed as consistent with the official Scrum Guide, with a few notable exceptions. The most important exception is that the Scrum Guide requires all iterations (sprints) to be of equal length in time and to not overlap. However, iterations in DDS are capability-focused, and hence, can have a flexible duration. This acknowledges that some items are exploratory in nature, and hence, effort estimation can be difficult. Furthermore, a capability-focused iteration acknowledges that some items require substantially more time than others (ex. data collection and cleaning might take weeks, whereas some exploratory analysis might take one day). The other notable exception is that retrospectives and item reviews are not done at the end of every iteration, but rather, on a frequency the team deems appropriate.

Furthermore, in many Scrum implementations, observing, analyzing and reacting to feedback is solely the responsibility of the product owner, with only marginal support from the rest of the team. This part of the product owner's job largely falls outside of the codified process. Drawing appropriate conclusions from an iteration is a crucial part of the data science process and by defining these steps directly into the core workflow, DDS should help teams make better data-driven decisions.

7.2 Conclusion

This paper defines a new process methodology that can improve how teams execute data science projects. The paper first defines four lean data science principles and then describes Data Driven Scrum (DDS), a new lean agile iteration-based framework for data science projects which supports the lean data science principles that were defined as part of this research. The paper then explores how DDS can be used within a real-world context.

With respect to the pilot use case, based on the knowledge gained with respect to DDS, the organization did understand the key lean agile concepts (as it relates to data science), and adapted their data science process to support a lean agile process via using the DDS framework (thus addressing RQ1 and RQ2).

The main limitation in this research was that DDS was only observed within one organization, and the focus was on how the team adapted their process. Future research should explore how other organizations view DDS, how they implement the DDS framework, and the impact of using DDS (within this organization as well as other organizations).

References

[1] Saltz, J. and Stanton, J. (2017). *An Introduction to Data Science*. SAGE Publications.

[2] Ransbotham, S., Kiron, D. and Prentice, P. K. (2015). Minding the analytics gap. *MIT Sloan Management Review*, *56*(3), 63.

[3] Ahmed, B., Dannhauser, T., & Philip, N. (2018). A Lean Design Thinking Methodology (LDTM) for Machine Learning and Modern Data Projects. In *2018 10th Computer Science and Electronic Engineering (CEEC)* (pp. 11-14). IEEE.

[4] Chen, H., Kazman, R. and Haziyev, S. (2016). Agile Big Data Analytics for Web-based Systems: An Architecture-centric Approach, *IEEE Transactions on Big Data*.

[5] Ries, E. (2011). The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses. *New York: Crown Business*.

[6] Saltz, J., Hotz, N., Wild, N. and Stirling, K. (2018). Exploring Project Management Methodologies Used Within Data Science Teams. *Americas Conference on Information Systems (AMCIS)*.

[7] Ponsard, C., Majchrowski, A., Mouton, S., & Touzani, M. (2017). Process Guidance for the Successful Deployment of a Big Data Project: Lessons Learned from Industrial Cases. In *IoTBDS* (pp. 350-355).

[8] *Why do 87% of data science projects never make it into production?* (2019) https://venturebeat.com/2019/07/19/why-do-87-of-data-science-projects-never-make-it-into-production

[9] Akred, J. (2016). *Using Agile development techniques for data science projects*. (B. Lorica, Interviewer), https://www.oreilly.com/ideas/using-agile-development-techniques-for-data-science-projects

[10] Cao, L. (2017). Data science: challenges and directions. *Communications of the ACM*, *60*(8), 59-68.

[11] Angée, S., et al. (2018). Towards an Improved ASUM-DM Process Methodology for Cross-Disciplinary Multi-organization Big Data & Analytics Projects. In *International Conference on Knowledge Management in Organizations* (pp. 613-624).

[12] Womack, J., and Jones, D. (2003). Lean thinking: banish waste and create wealth in your corporation (2nd, revised and updated ed.). *New York, NY: Free Press*.

[13] Poppendieck, M. & Poppendieck, T. (2003). Lean software development: an agile toolkit. *Addison-Wesley*.

[14] Anderson, D., "Kanban: Successful Evolutionary Change for Your Technology Business". Sequim, WA: Blue Hole Press, 2010.

[15] Ahmad, M. O., Kuvaja, P., Oivo, M., & Markkula, J. (2016, January). Transition of software maintenance teams from Scrum to Kanban. *In Hawaii International Conference on System Sciences* (HICSS)

[16b] West, D., 2017. Scrum Guide Update, Scrum.org, www.scrum.org/resources/blog/scrum-guide-update-november-2017

[17] Rigby, D. K., Sutherland, J., & Takeuchi, H. (2016). *Embracing Agile*. Retrieved from: https://hbr.org/2016/05/embracing-agile

[18] Sutherland, J., & Schwaber, K. (2020). *The Scrum Guide*. Retrieved from scrumguides.org: https://scrumguides.org/scrum-guide.html

[19] Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C. and Rudiger, W. (2000). *CRISP-DM 1.0*. Retrieved from The Modeling Agency: www.the-modeling-agency.com/crisp-dm.pdf

[20] Saltz, J. (2020). RISP-DM is Still the Most Popular Framework for Executing Data Science Projects. Retrieved from: https://www.datascience-pm.com/crisp-dm-still-most-popular

[21] Mason, H., Wiggins, C. (2010). A Taxonomy of Data Science. Retrieved June 2019, from www.dataists.com/2010/09/a-taxonomy-of-data-science.

[22] Microsoft. (2021). *Team Data Science Process from Microsoft*. Retrieved from https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/

[23] Saltz, J., and Shamshurin, I. (2019). Achieving Agile Big Data Science: The Evolution of a Team's Agile Process Methodology. In 2019 *IEEE International Conference on Big Data (Big Data)* (pp. 3477-3485).

[24] Ikonen, M., Pirinen, E., Fagerholm, F., Kettunen, P., & Abrahamsson, P. (2011). On the impact of Kanban on software project work: An empirical case study investigation. In *Engineering of Complex Computer Systems (ICECCS*), 2011 16th IEEE International Conference on (pp. 305-314). IEEE.

[25] Lawler, J., & Joseph, A. (2017). Big Data Analytics Methodology in the Financial Industry. *Information Systems Education Journal*.

[26] Baijens, J., Helms, R. and Iren, D. (2020). Applying Scrum in Data Science Projects. In 2020 IEEE *22nd Conference on Business Informatics*

[27] Saltz, J., & Sutherland, A. (2020). SKI: A new agile framework that supports DevOps, continuous delivery, and lean hypothesis testing. In *Proceedings of the 53rd Hawaii International Conference on System Sciences* (HICSS).