HOW TO MANAGE DATA SCIENCE PROJECTS

A Quick Guide

This is a How To Manage Data Science Projects Quick Guide which explores a range of topics like the data science project life cycle, workflow frameworks, what is agile data science, how to measure a project metrics, ethics and how to lead a machine learning team. Etc.

TABLE OF CONTENTS

Note: Most of the content explanations and wordings are taken from

1. The slide material provided by Professor Saltz
2. DataScience-pm website where most of the content is produced by Professor Saltz
3. Browsing the web and taking great explanations

Since I really wanted this material to be a good reference for myself and anyone around me going forward, I tried to compile work done by different professionals to produce this document.

HOW TO MANAGE DATA SCIENCE PROJECTS

A Quick Guide

Introduction to Managing Data Science Projects

Data Science Project Management: throughout this guide we will explore, agile data science, how to lead a data science team, what is the need for an improved process, the difference and similarities between a data science project and other IT projects, why managing data science project is so important, Types of data science workflows, project frameworks, ethics and how to measure data science projects.

Also, after going through this guide one should learn to

1. Describe the key differences between data science projects and software development projects

2.Explain how to use several data science workflow processes (such as CRISP-DM and OSEMN)

3.Explain why agility is important for data science projects

4.Articulate the key aspects of Kanban, Scrum, and DDS process frameworks

5.Leverage agile concepts within a data science project context

6.Select / use the most appropriate team process framework for a specific project

7.How a data science project can effectively work with the rest of an organization

DATA SCIENCE

Data science: Generating actionable insight via the collection, preparation, analysis, visualization, management, and preservation of large collections of information.

It strongly connects with areas such as databases, statistics, and computer science, but many other skills are also needed (e.g., Domain knowledge, communication & collaboration)

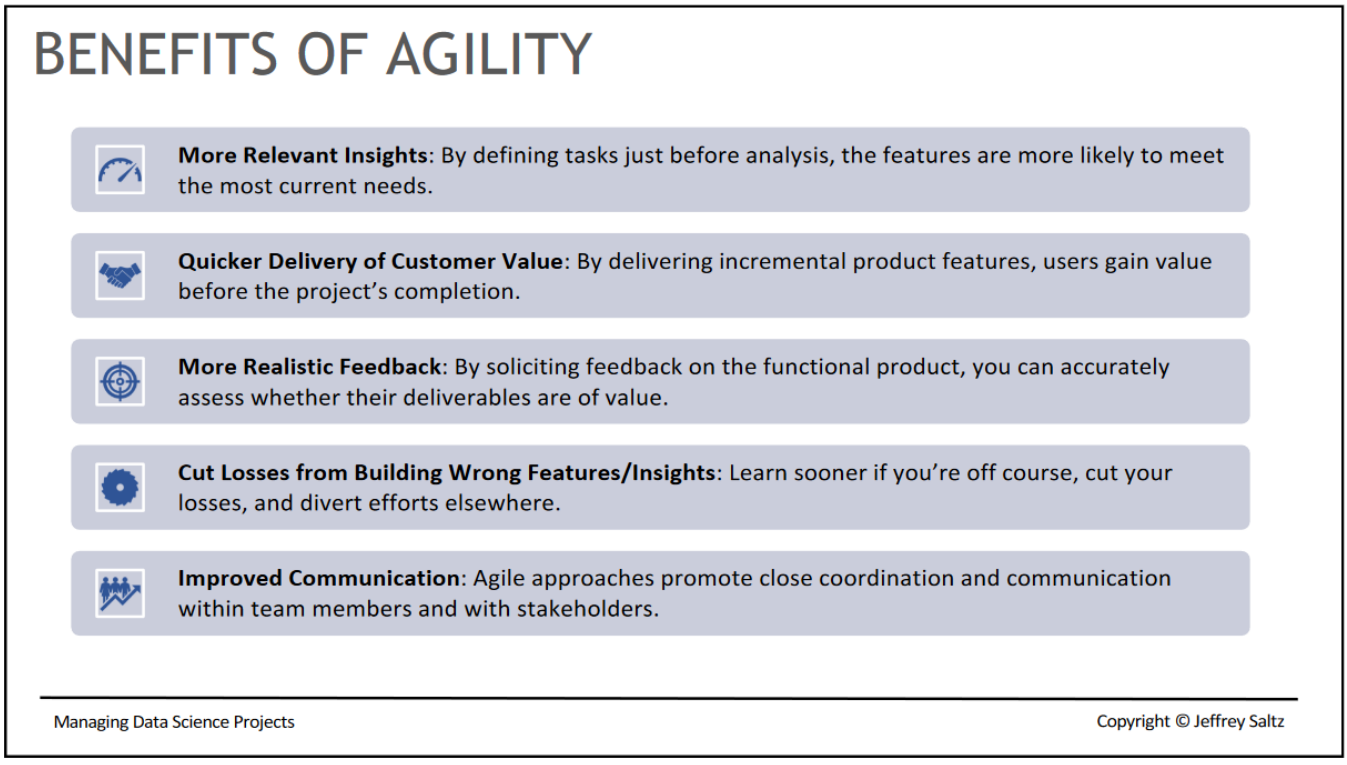
Introduction to Agile Data Science

WHAT IS AGILE DATA SCIENCE?

Agile Data Science is not just about how to ship working software, but how to better align data science with the rest of the organization. There is a chronic misalignment between data science and engineering, where the engineering team often wonder what the data science team are doing as they perform exploratory data analysis and applied research. The engineering team are often uncertain what to do in the meanwhile, creating the "pull of the waterfall," where supposedly agile projects take on characteristics of the waterfall. Agile Data Science bridges this gap between the two teams, creating a more powerful alignment of their efforts.

Agile Data Science aims to put you back in the driver's seat, ensuring that your applied research produces useful products that meet the needs of real users.

In general, Agile Data Science Means
- Rapidly deploy meaningful incremental deliverables to the stakeholders
- Adjust plans based on model assessment AND market and stakeholder feedback
- Allow the data science team members to self-organize

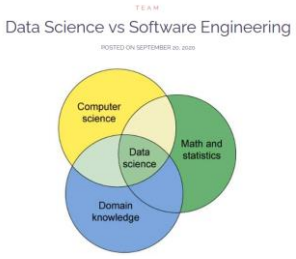


How to lead/manage a data science team?

It's an understatement that great leadership is challenging and rare. And leading data science teams has unique challenges:

- Stakeholders might get disillusioned by your team's inability to deliver magic
- The battle to recruit and retain data science talent is fierce
- Data science's ethical dilemmas are particularly perplexing
- There is not an agreed-upon process for managing data science teams
- Few people possess a solid mix of the technical chops and softer leadership skillsets
- …and the list goes on

whether you're a data science manager, a student, a tenured individual data scientist, or a businessperson branching out into this nascent field, I wish you the best on your journey to becoming a stronger and more fulfilled leader. Here are eight tips to get you started or to keep you going…



The battle between Data Science vs Software Engineering isn't really a battle. Rather these are two somewhat overlapping and complementary fields that are similar in many ways. However, dig deeper, and you'll find key differences: Data science is more exploratory. Software engineers are more focused on systems building. And data science project management should be

8 Tips to Leading Data Science Teams

1. Start with Why

How often do we jump right into "what" needs to be done and "how" to do it without understanding the deeper and more critical question of "why" we should do it in the first place? Yet, as Simon Sinek explains a clear and meaningful "why" drives action — not the "what or the "how".

As such, great leaders inspire their teams with a meaningful purpose to rally around. And when kicking off a new project, the leader should dive deeper and make sure any project they undertake has a "project why" that is consistent with the team's motivating purpose.

This takes effort. But the investment in a clear "why" yields dividends for your team in so many aspects from higher productivity, better staff retention, and ultimately to clearer analyses and results.

2. Engage Stakeholders

At the end of the day, teams need to deliver value to a set of stakeholders. The most effective leaders will:

1. Identify their stakeholders (which usually extends beyond just the obvious project requester)
2. Listen to their requests
3. Identify their needs (which is often different from their request…see the meme below)
4. …And learn how to best engage stakeholders throughout the data science lifecycle

Don't leave stakeholders in the dark or mistake their "requests" as their "needs". Rather, dig deeper. Actively engage them and uncover their needs by leveraging agile principles such as satisfying the customer "through early and continuous delivery" and having "businesspeople and developers […] work together". Which leads us to the next point…

3. Implement Effective Processes

This does not necessarily mean to implement a specific framework such as Scrum but rather that you lead your team to:

- Educate your team on the "why" behind good processes
- Discover an effective process that fits the team and its work's unique needs
- Foster a culture of continuous process improvement

4. Build the Right Data Science Team

- Like any good team, a data science team needs to have the right people to get the job done. And just like your team process, your team composition is dependent on the organizational structure, the company culture, and the type of problem you're trying to solve.
- If you're new to the field, avoid the common misconception that a fully functional data science team just has a bunch of data scientists. Rather, it has all the needed roles to deliver a solution. In some circumstances, this might indeed be heavily data scientist focused. But probably you need a diverse set of roles including business analysts, data architects, data engineers, machine learning engineers, a project manager, product manager, and of course data scientists.
- To lead a data science team, you need to understand these required roles, how to attract and retain the right talent, and how to further develop the individual team members. Technical skillsets are obviously key. Equally important are the softer skills that team members need to become effective contributors.

5. Build a Data Science-Specific Culture

- On the surface, this is another no-brainer. And yet, data science teams are often misunderstood as software teams. While these fields indeed overlap significantly, the data scientist has a distinct mindset from that of a typical software engineer. Just a few differences:

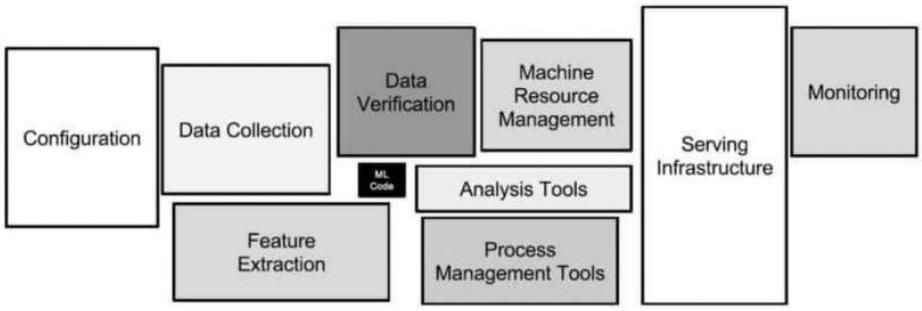
Area	Data Scientists	Software Engineers
Drive	Discovery and exploration	Implementing a solution
Ambiguity	"That's fine. It's my job to sort through the noise."	"I need clear requirements before starting."
Key skill sets	Math, stats, and some coding	Building production systems

A few differences between data scientists and software engineers

- A few differences between data scientists and software engineers
- As such, managing data scientists as software engineers will likely leave them feeling misunderstood, and shoehorning their projects as software projects will likely lead to frustrating non-productive planning exercises that can siphon time and energy away from the team. Rather, build a culture where data scientists can be at their best.

6. Focus on the Long Term

It's easy to focus on generating an interesting machine learning model. That's what data science is all about, right? Well…while the model is a key and necessary part of the overall data science process, **the model by itself is usually not sufficient to deliver value.**



A production system has much more than just ML code

Rather, to deliver sustainable value, predictive models usually should be put into sustainable and stable systems that the stakeholder can access

To ensure your team's work delivers on-going value, you'll have to balance what might seem like a never-ending firehose of stakeholders requests with the need to dedicate the time

necessary to build production systems that check incoming data, provide alerts if data is missing or out of acceptable ranges, and deliver accuracy metrics that allow the data scientists to monitor and tune the models when needed.

The stakeholders might not understand this value but it's your responsibility to educate them. Additionally, allocate development time for full-fledged systems production (or work with the team that is responsible for this). When needed, push back and say no to new incoming requests to allow time to clean up any unnecessarily accrued technical debt. You'll thank yourself later.

7. Integrate Ethics into Everything

Do you know if all your team's practices and your projects are ethical? Well, that's a tough question to digest!

As a start, ensure that your teams' and project outcomes are compliant with industry-relevant laws. But go beyond this to protect people's privacy, minimize/remove unfair bias results against certain population segments, be keenly aware of how your work impacts the broader society, and mitigate any potential adverse outcomes. Your assessments could literally mean life and death.

8. Know Where to Learn More

So what else do you need to do to lead data science teams? There's no exhaustive list but perhaps the best advice I can give is to know where you can dive deeper into the above-mentioned topics and into the numerous issues I didn't mention. Here are a few places to turn:

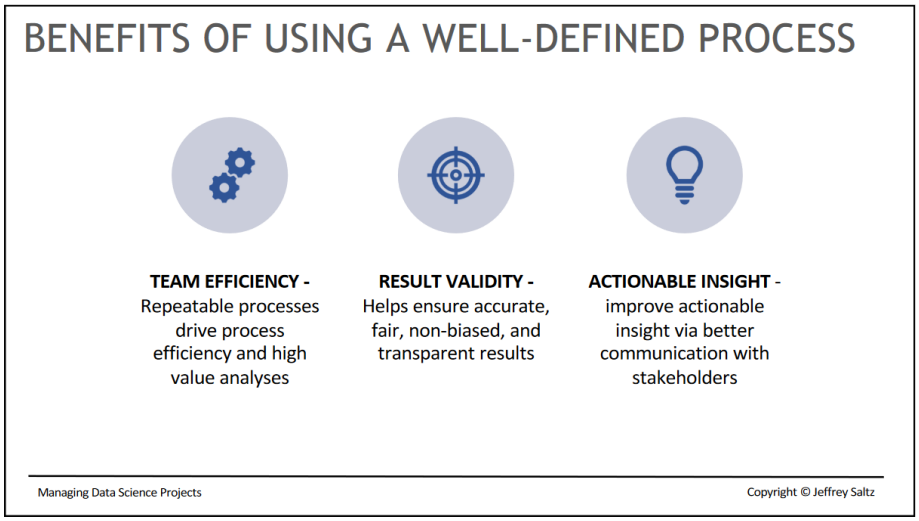
The following links are a great place to learn, but you can also dive in to any resource you think will be helpful Data Science Process Alliance which provides individual training and corporate consulting to better lead data science projects and teams.

The Need for An Improved Process

Following a structured approach to data science helps you to maximize your chances of success in a data science project at the lowest cost. It also makes it possible to take up a project as a team, with each team member focusing on what they do best. Take care, however: this approach may not be suitable for every type of project or be the only way to do good data science.

Not just for a data science project, you know there is a need for an improved process when you see the following signs of poor process.

- When many projects are being juggled at once, which leads to projects taking very long time to complete and tracking project status becomes burden
- When stockholders think generated insight is not useful, don't trust the data or model, DS team is not productive, or the DS team is not focused on the highest priority task
- When Communication "clunky" with broader organization



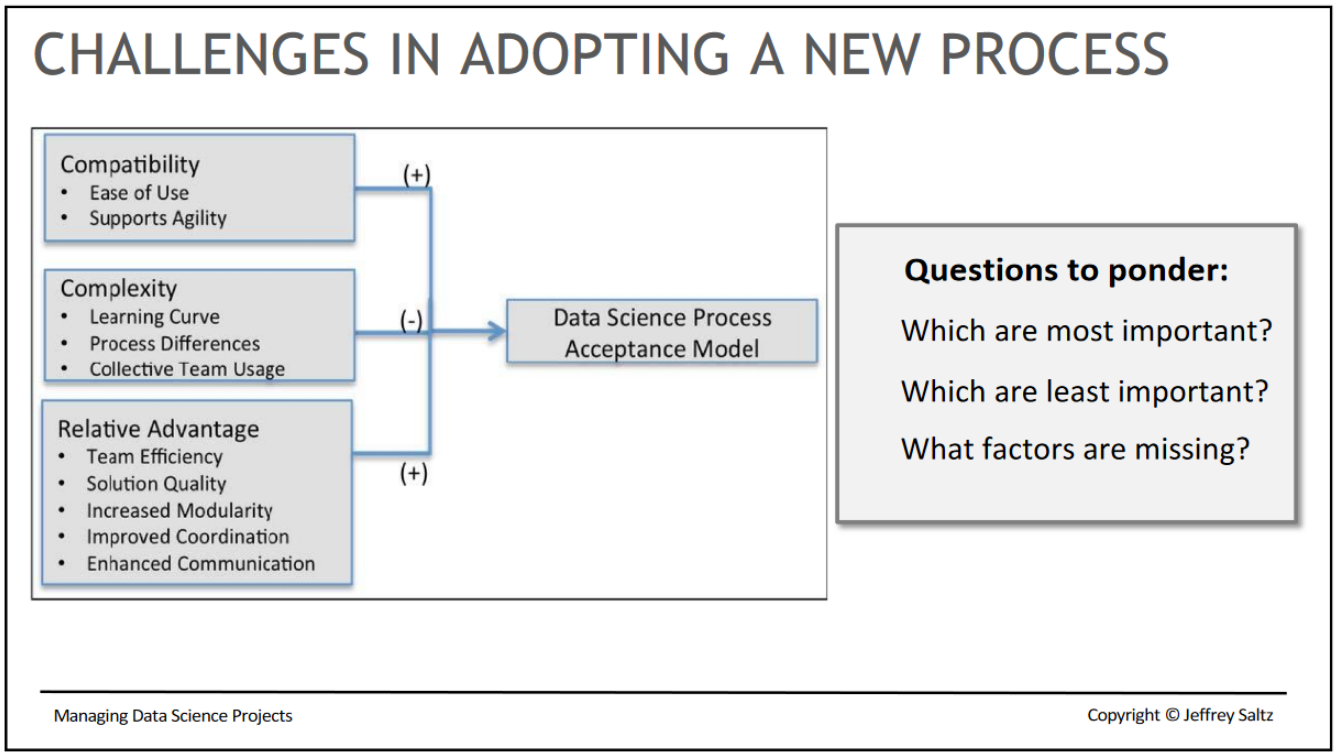
Another point in this section is, just for the sake of having a process, if you try to fit your data science project into your traditional software development process, it won't work because

Data science IS NOT EQUAL TO Software development

The fact that

- 80% of AI projects are run by wizards whose talents will not scale in the organization -Gartner (2019)
- 62% of high performing AI organizations collaborate across teams-McKinsey (2019)
- 85% of data scientists think adopting an improved process would improve results-Big Data Science Conference Survey (2018)
- Only 48% of data science organizations have established standardized processes –Corinium Execute survey (2020)

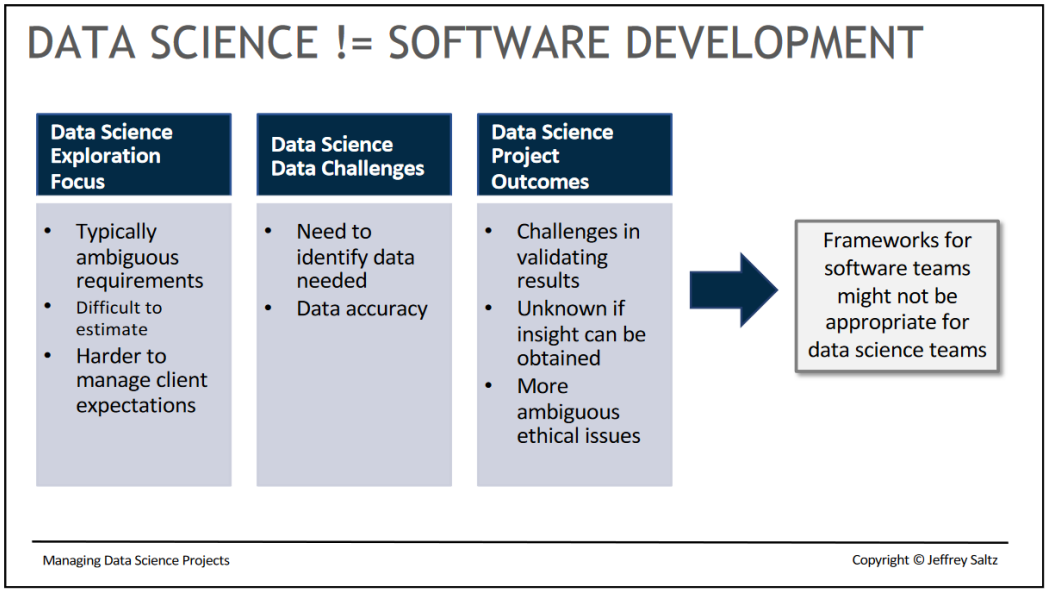
Show a clear need for Data Science Process Improvement now the question is what are some of the challenges we might face while trying to adopt a process?



HOW IS A DATA SCIENCE PROJECT DIFFERENT FROM OTHER IT PROJECTS? AND HOW MIGHT IT BE SIMILAR?

In a traditional IT project, a business case is identified, a system is developed to meet the needs of the business case, timelines for deliverables are drawn up, and everyone enlisted in the project is tasked with work that must conform to documented requirements and come in on time. There are few ambiguities in well-constructed IT projects, and everyone understands marching orders.

This isn't always the case in data science, in which business cases can be drawn up but arriving at the desired results isn't always straightforward and predictable. In fact, the only hard metric that seems to exist for most data science projects is that the results derived from algorithms operating on data must be at least 95% "right" when compared with an accepted standard for determining correctness.



DIFFERENCES

- Age: The field of software engineering is much older, stemming from the 1940s. And, although the concept of data analytical processes stems back centuries, the modern field of data science is much newer. The term "data science" itself didn't even crop up until the past two decades.
- Organizational Understanding: Related to the prior point, organizations often have less of a clear idea of what is possible from data science and what to expect from their data science teams.
- Establishment: Many complain that "data science" is a buzzword, and Merriam Webster doesn't even define the term. However, no one questions software engineering's existence as a distinct field.
- Problem space: Data science focuses on exploration and discovery (such as "finding insight in the data", and identifying new data sources that can be integrated into predictive models), while software engineering typically focuses on implementing a solution that addresses specific requirements (perhaps defined incrementally).
- Domain Focus: Although both fields rely on data, math, and code, data science emphasizes the data and math while software engineering is more heavily code-oriented.

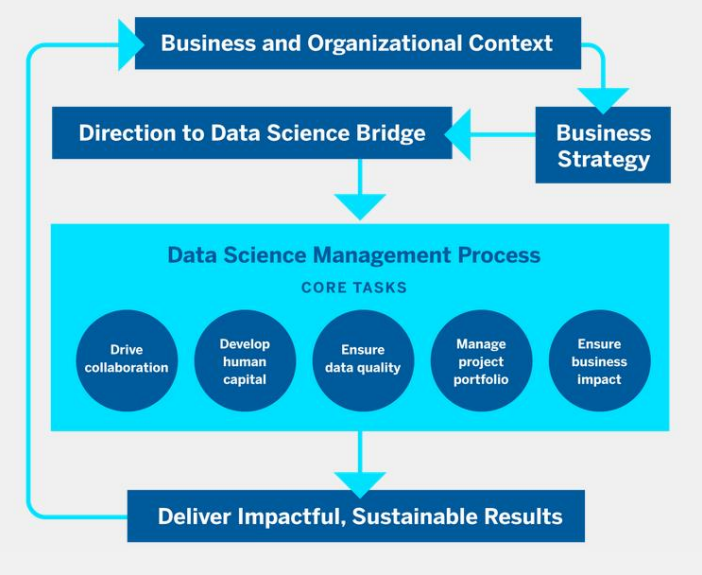
SIMILARITIES

- Data Science and Software Engineering both involve programming skills
- Both fields rely on data, math, and code
- Both uses some sort of IDE environment
- Both are high-tech fields that are largely built on top of the fields of computer science and mathematics.

WHY MANAGING A DATA SCIENCE PROJECT IS IMPORTANT?

It is increasingly clear that companies and government agencies do not know how to manage data science at the enterprise level. Many are still stuck doing pilots. Some take on projects that are beyond their capabilities. And too often, excellent work dies on the vine during implementation. Companies must take action to address the structural and process issues that hold them back that is why we need to manage data science projects

The management of data science projects should be a continuous loop: An organization's overall strategy feeds into the directions given to the "data science bridge," the team that oversees all projects. That team engages in five core tasks to manage the portfolio. Results then loop back to provide new insights for the organization's overarching strategy.



From the above diagram, if we focus on the Data Science Management Process section, you can clearly see that managing data science projects will have the following importance

- It will help drive collaboration
- It will help develop human capital
- It will help ensure data quality
- It will help manage project portfolio
- It will help ensure business impact

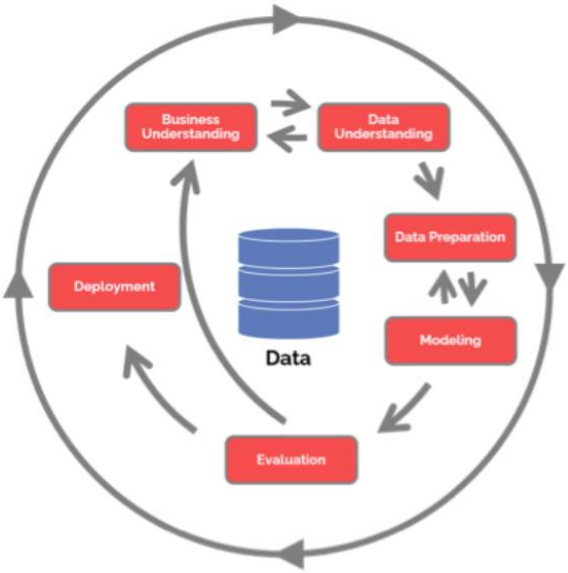
In addition to the above advantages, by managing data science projects the poor process signs we discussed earlier can- mostley be avoided.

Project Frameworks and/or Data Science Workflows

Types of frameworks that could be used on data science projects

A data science workflow is what determines the steps in a data science project. There are many data science workflows and we will discuss some of them here

1. CRISP-DM: Defined to standardize a data mining process across industries, CRoss-Industry Standard Process for Data Mining (CRISP-DM) is the most well-known framework used to define a data science workflow. As shown in the standard CRISP-DM visual workflow, it describes six iterative phases. Each phase (Business understanding, data understanding, data preparation, modeling, evaluation and deployment) has its own defined tasks and set of deliverables such as documentation and reports. Each phase has its own defined tasks and set of deliverables (including documentation and reports). Projects can "loop back" as needed to a previous phase.



I. BUSINESS UNDERSTANDING
- Determine business objectives
- Assess situation
- Determine data mining goals
- Produce project plan

The initial focus helps to align technical work with business needs and to steer data scientists away from jumping into a problem without properly understanding business objectives.

II. DATA UNDERSTANDING
- Collect initial data
- Describe data

- Explore data
- Verify data quality

Data Understanding helps ensure there is discussion on what underlying data is available and what might be linked

III. DATA PREPARATION
- Select data
- Clean data
- Integrate data
- Format data
- Construct data (feature engineering)

Data preparation is often the most time-consuming aspect of a project – cleaning / munging the data

IV. MODELING
- Select modeling technique
- Generate test design
- Build & tune model
- Assess model

Modeling is what many people think of when they think of data science –using techniques such as machine learning to gain insight from the data.

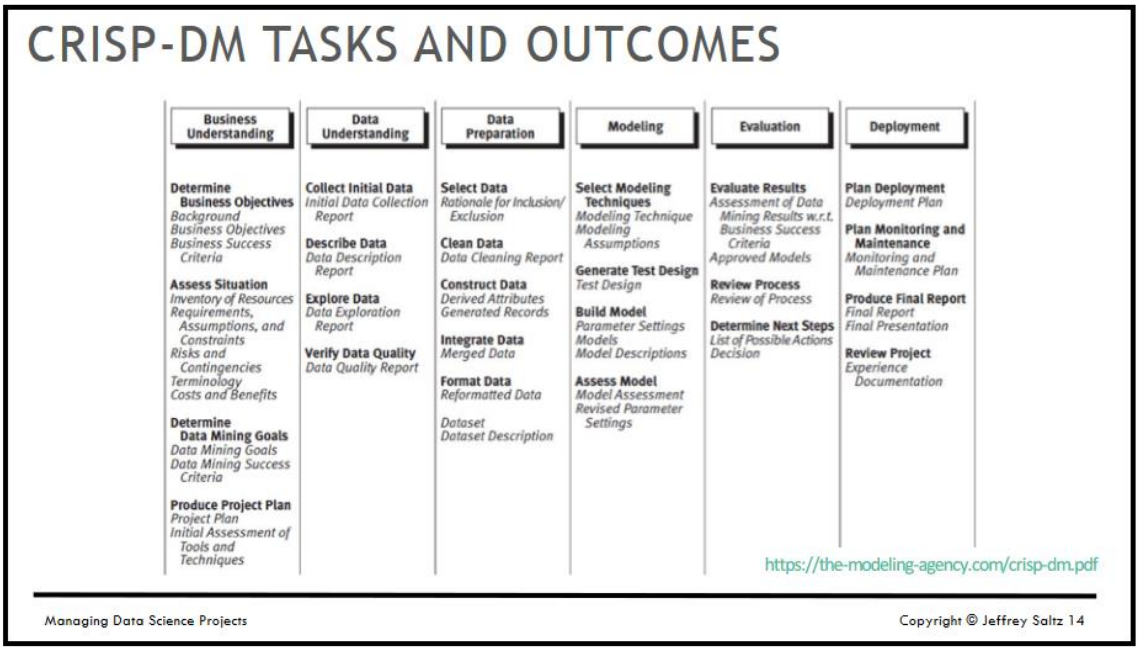
V. E VA L U AT I O N
- Evaluate results
- Review process
- Determine next steps

The goal of Evaluation is to assess to what extent the model meets the business outcomes. This might include steps like doing initial A / B testing on a subset of customers to see how the model performs in the wild.

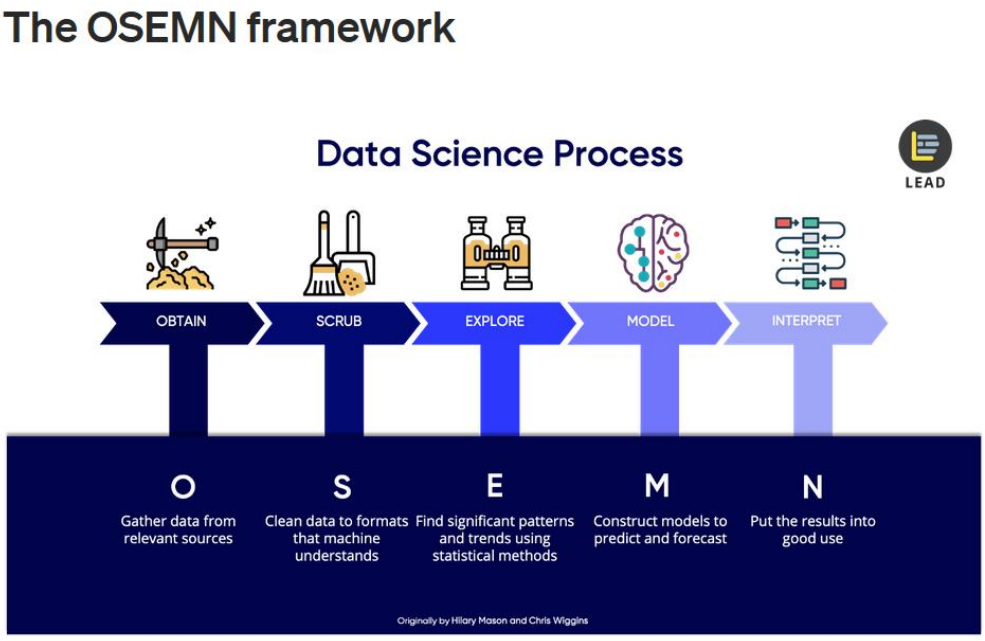
VI. DEPLOYMENT
- Plan deployment
- Plan maintenance and monitoring
- Product final report
- Review project

Deployment varies by project –ranging from real-time prediction to one-off analysis.

This picture is a great summary of what will be done in each CRISP-DM phase



2. OSEMN: means Obtain, Scrub, Explore, Model, and iNterpret and it is described in terms of skills of a data scientist (not about a team)



Data Science Process (a.k.a the O.S.E.M.N. framework)

I. Obtain: The very first step of a data science project is straightforward. We obtain the data that we need from available data sources.

II. Scrub Data: After obtaining data, the next immediate thing to do is scrubbing data. This process is for us to "clean" and to filter the data. Remember the "*garbage in, garbage out*" philosophy, if the data is unfiltered and irrelevant, the results of the analysis will not mean anything.

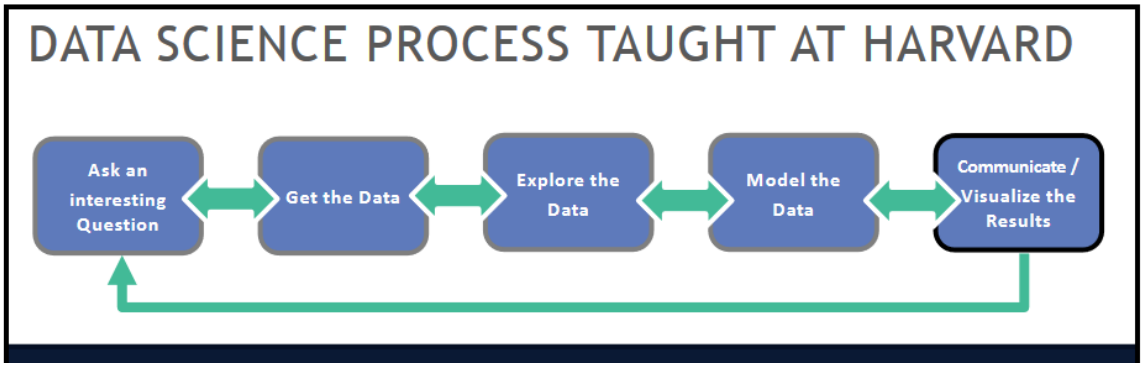
III. Explore Data: Once your data is ready to be used, and right before you jump into AI and Machine Learning, you will have to examine the data.

IV. Model Data: This is the stage where most people consider interesting. As many people call it "*where the magic happens*".

V. Interpreting Data: We are at the final and most crucial step of a data science project, interpreting models and data. The predictive power of a model lies in its ability to generalize. How do we explain a model depends on its ability to generalize unseen future data.

3. Harvard's Phased-based Workflow

The process mentioned in Harvard's introductory data science course, it's a 5 Phase Framework & it Integrates and loops between the phases



I. Ask interesting questions: questions like, what is the goal? What would you do if you had all the data? & What do you want to predict, or estimate? are some of the questions that can be asked in this phase

II. Get the Data: questions like How were the data sampled? Which data are relevant? Are there privacy issues? Can be asked

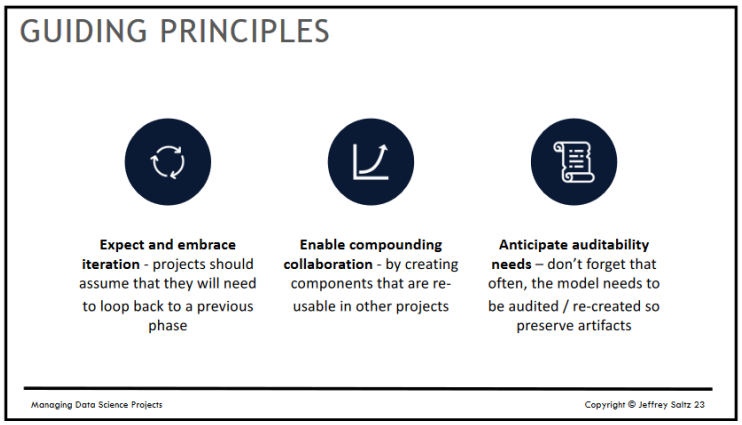
III. Explore the Data: Plot the data, are there anomalies or egregious issues? Are there patterns?

IV. Model the Data: Build a model, fit the model & validate the model

V. Communicate/Visualize the result: What did we learn? Do the results make sense? Can we effectively tell a story?

4. Domino's Phased-based Workflow: is one organization's view of "how to do data science, published in 2017 and defines three guiding principles and six project phases

See the 3-guiding principle of this framework below



Do has 6 phases

I. IDEATION: Identify problem, Scope project, Select Artifacts, Go / No Go Decision

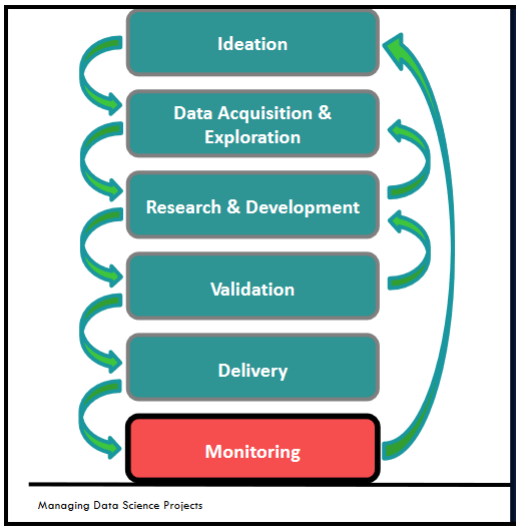
II. DATA ACQUISITION & EXPLORATION: Identify data (Connect, Capture, Buy?), Explore data, Prepare data

III. RESEARCH & DEVELOPMENT: Generate hypothesis, Get tools, Share insight

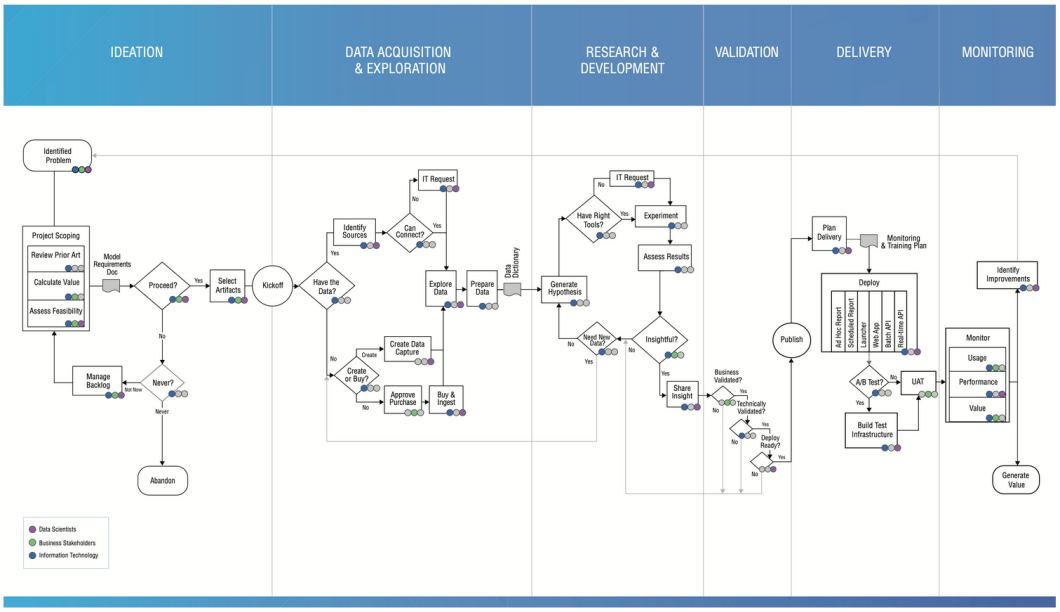
IV. VALIDATION: Business validation, Technical validation, Deployment ready?

V. DE LI VE RY: Plan delivery, Deploy, A/B Test, UAT

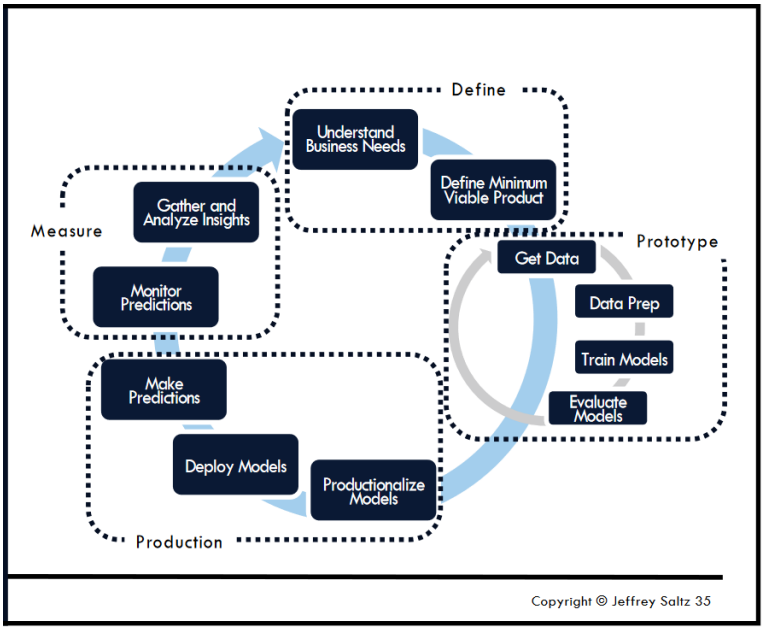
VI. MONITORING: Monitor usage, Monitor performance, Monitor value, Identify improvements



This framework is a little bit different from what we have seen above because of the monitoring phase. Also, see the detailed diagram as to what is done in each phase



5. Uber's Phased-based Workflow: As you can see from the diagram below, Uber has a Define phase, Prototype Phase, Production Phase & Measure Phase which we will discuss each phase in detail



I. DEFINE: Understand customer needs / problems, Define the minimal viable insight or product (key metrics and minimum features)

II. PROTOTYPE: Get data and prepare data (cleaning, munging, feature engineering, Train models and evaluate models (build models, conduct error analysis)

III. PRODUCTION: Productionalize and deploy the models (store and package model), Make predictions (use the model)

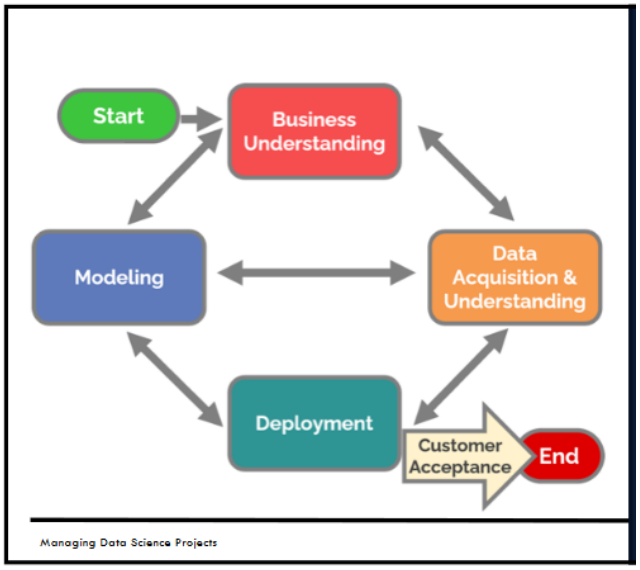
IV. MEASURE: Monitor predictions (including reactions to predictions), Gather & analyze insights (evaluate model and next steps)

Uber's Phased-based Workflow looks like very modern and the measure phase makes it different from other frameworks

6. Team Data Science Process (TDSP): Was launched in 2016 by Microsoft, it has 5 phases, 4 roles and 10 artifacts

- Microsoft also provides Tools, Utilities & Infrastructure for project execution
- TDSP can be applied in other (non-Microsoft) environments
- This discussion will focus on the general framework, not Microsoft's tools

The customer acceptance phase makes it unique, see figure below



I. BUSINESS UNDERSTANDING: Define objectives, Identify data sources

ARTIFACTS Needed
•Charter Document
•Data Resources
•Data Dictionary

II. DATA ACQUISITION & UNDERSTANDING: Ingest data, Explore data, Set up data pipeline

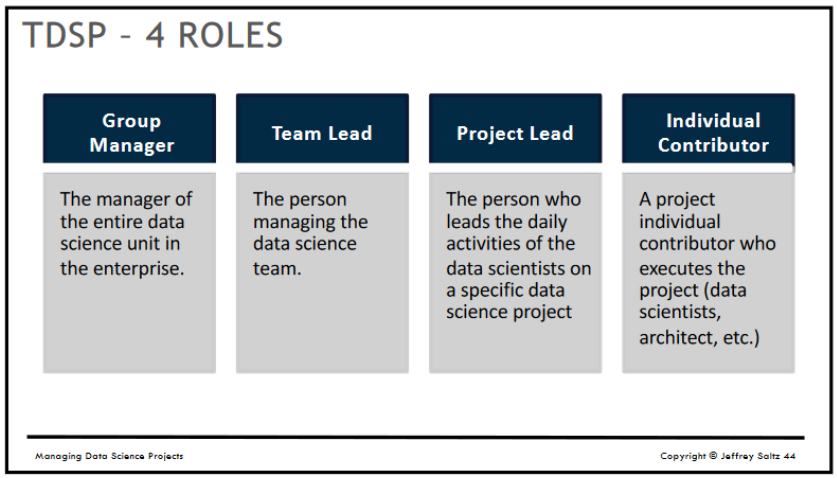
ARTIFACTS Needed
- Data quality report
- Solution architecture

III. MODELING: Feature engineering, Model training, Assess suitability for production
ARTIFACTS Needed
- Feature sets
- Model report

IV. DEPLOYMENT: Operationalize the model
ARTIFACTS Needed
- Status dashboard
- Final modeling report

V. CUSTOMER ACCEPTANCE: System validation, Project hand-off

ARTIFACTS Needed
- Exit report

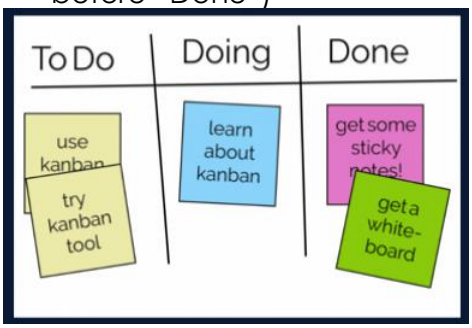


KEY FRAMEWORKS THAT MIGHT BE USED - USING THE TYPE OF FRAMEWORK- STRUCTURE DEFINED ABOVE

I will be discussing the 3 frame works that could be used with the workflows above

1. Kanban: is a popular framework used to implement agile and software development. It requires real-time communication of capacity and full transparency of work. Work items are represented visually on a kanban board, allowing team members to see the state of every piece of work at any time. Also, its an implementation of lean and data science teams are adopting it.

I. Principle 1: Visualize the Flow
- The team starts with a list of potential tasks in the "To Do" column
- In a simple three column Kanban board:
  - When a team starts working on the task, the Kanban card (task) is moved from the "To Do" to the "Doing" column
  - When the team completes its task, it is moved to the "Done" column
- Teams often define more columns (e.g., a validation column before "Done")



II. Principle 2: Limit WIP
- Uncompleted work is known as work in progress (WIP)
- WIP limits define the maximum number of tasks that can simultaneously exist in each column

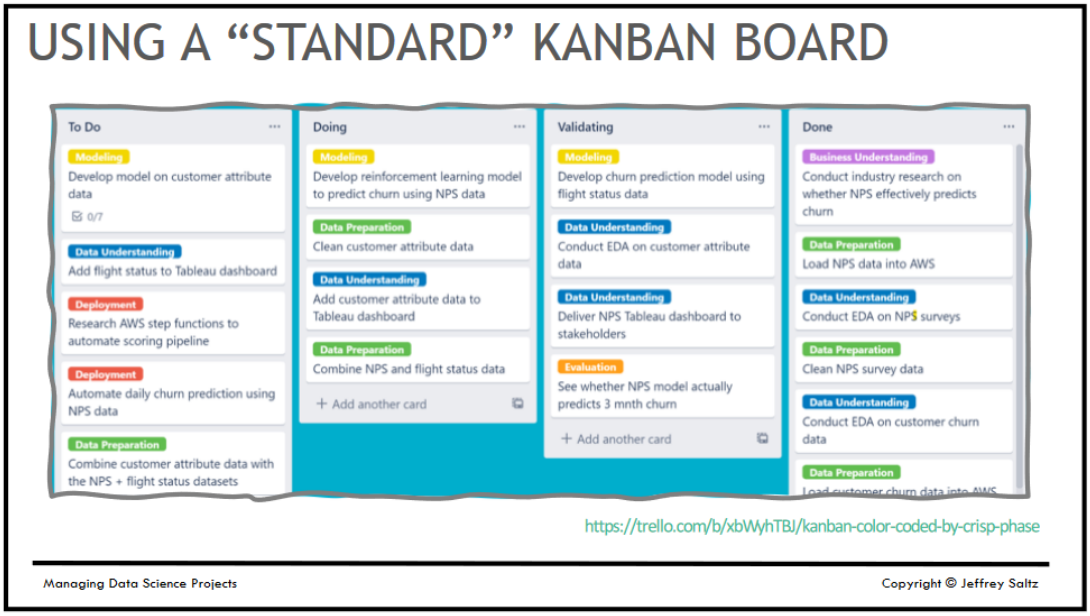
III. Additional Principle: Manage & Measure the Flow
- Done directly on the Kanban board
- Focus on identifying and addressing bottlenecks and blockers
- Prevent too much work piling up at a certain phase
- Encourages members to own the overall value flow
- Facilitates team members to expand skillsets into other project phases
- Measure lead and cycle times (how quickly value is delivered)

IV. Additional Principle: Make Process Policies Explicit
- Important since Kanban doesn't define process policies
- Examples of policies include: WIP Limits, definition of done, how to prioritize tasks

Kanban can be integrated with any of the workflows discussed above and used for data science. For example: Kanban can be integrated with CRISP-DM workflow in this case what happens is, every phase will have a task decomposed and then prioritized in the Kanban "To Do" step then when work started it will be moved to "Doing" and whan its done , it will be moved to "Done".

Also, its possible to customize the Kanban board to your need: for example you can add columns like "Validating" and you can also decide to limit WIP. WIP is the maximum number of tasks that can simultaneously exist in each column.

Picture shows Kanban integrated with CRISP-DM



- As you can see Kanban board columns are defined as CRISP-DM phases
- Each high-level analysis goes through the CRISP-DM process

Pros of using Kanban

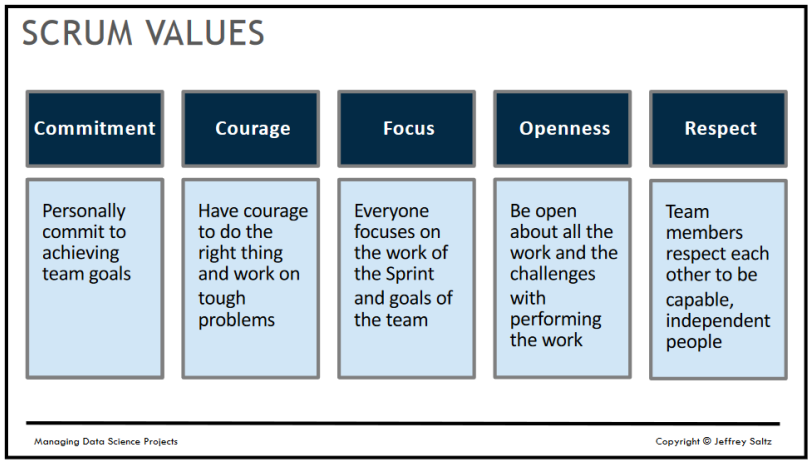
- Improves Communication, No Time Box, Enables Agility & Its Adaptable

Cons of Using Kanban

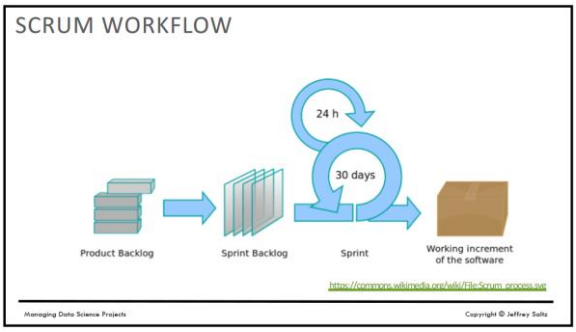
- It doesn't define: A project life cycle, Timelines or iterations, Team roles & Team ceremonies / meetings

2. SCRUM: It is a framework utilizing an agile mindset for developing, delivering, and sustaining products in a complex environment, with an initial emphasis on software development, although it has been used in other fields including research, sales, marketing and advanced technologies.

   Scrum is founded on empiricism (knowledge from experience) and lean thinking (reduce waste) where the empirical Scrum pillars are Transparency, Inspection & Adaptation



SCRUM Framework has 3 roles (Developer, Product Owner and Scrum Master), 5 events (The sprint, Sprint Planning, Daily Scrum, Sprint Review and Retrospective) & 3 artifacts(Product Backlog, Sprint Backlog and Increament). It can also be integrated with Kanban



Integrating Scrum with Kanban help items from each sprint get in to the "To Do" then move to "Doing" when its started and to "Done" once its done.

BENEFITS USING SCRUM FOR DATA SCIENCE

- Leverages empirical evidence
- Focuses on customer value
- Regular cadence
- Promotes autonomy
- Provides multiple inspection points
- Creates a constant sense of urgency

CHALLENGES USING SCRUM FOR DATA SCIENCE

- Hard to know what can be done in a sprint (time-based iteration)
  - Task estimation is unreliable ("what goes into a sprint")
- Some tasks take longer than others (but sprints are fixed duration)
  - A sprint does not allow smaller (or larger) logical chunks of work to be completed and analyzed in a coherent fashion
- Product backlog re-prioritized only after each sprint
- A sprint should be an intense focus ... but this doesn't make sense for some data science tasks (e.g. model might take model days to train)

Same way we integrated Kanban with CRISP-DM, SCRUM can also be integrated with any data science workflow including CRISP-DM

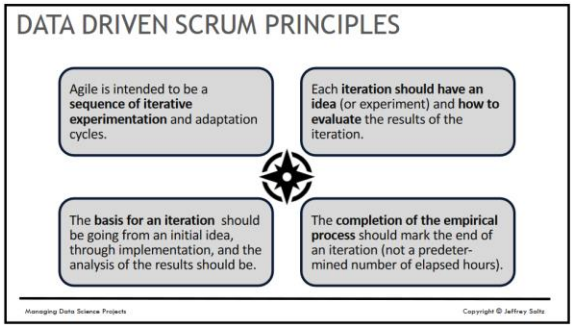
Unlike Kanban, Scum defines: A project life cycle, Timelines or iterations, Team roles & Team ceremonies / meetings.

The issue using Scrum with data science project is mostly its timeboxed nature which doesn't work that well with data science projects.

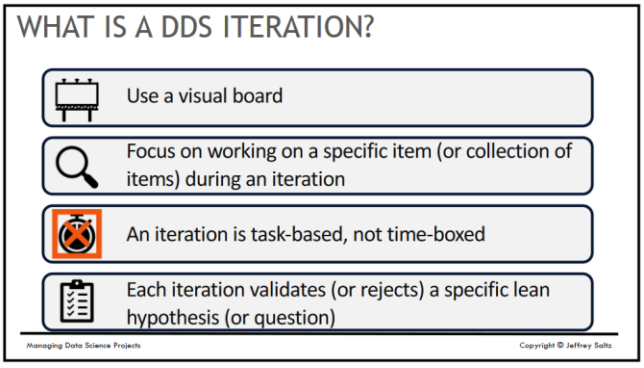
3. Data Driven Scrum(DDS): First published in 2019 (Saltz & Sutherland), Focus each iteration on create, observe and analyze something, Addresses two key challenges when trying to use Scrum for data science

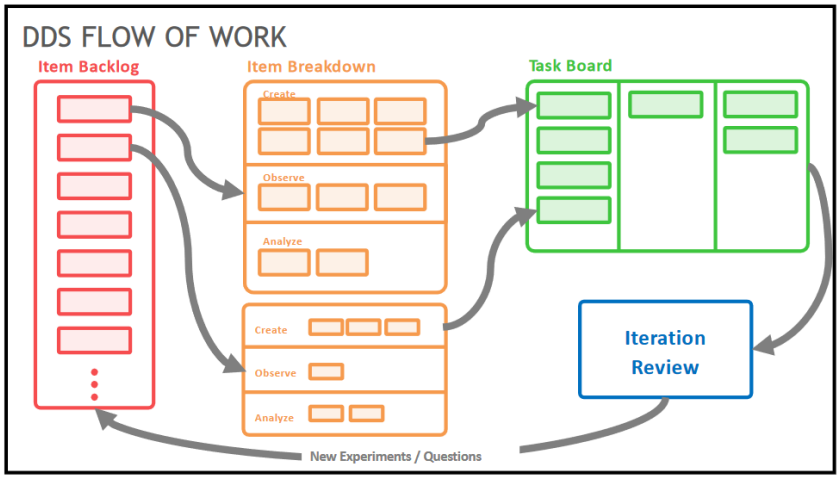
    - Task estimation is unreliable ("what goes into a sprint")
    - A sprint does not allow smaller (or larger) logical chunks of work to be completed and analyzed in a coherent fashion

Also, it decomposes tasks in the following categories (CREATE, OBSERVE, ANALYZE).



Key Pillars of Data Driven Scrum: Only require high level item estimation, Decouple meetings from an iteration & Allow capability-based iterations

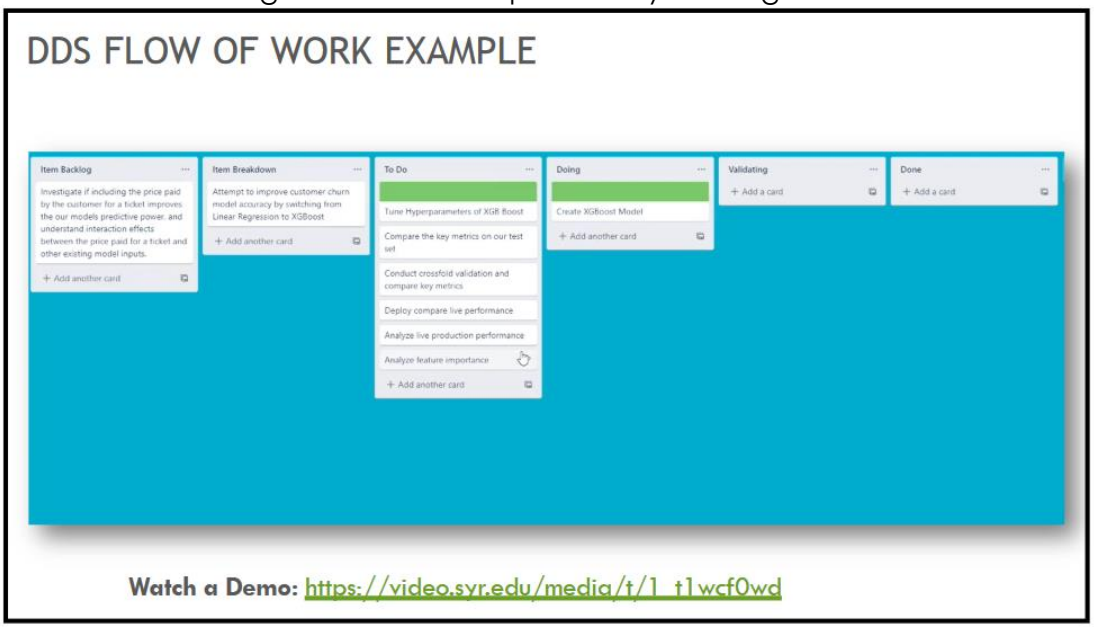




DDS has 3 artifacts namely Product Backlog Item (PBI), Item Backlog and Task Board. It has Product owner, process expert and development team roles and also it has 4 events: the iteration, Iteration review, Daily Meeting and Retrospective.

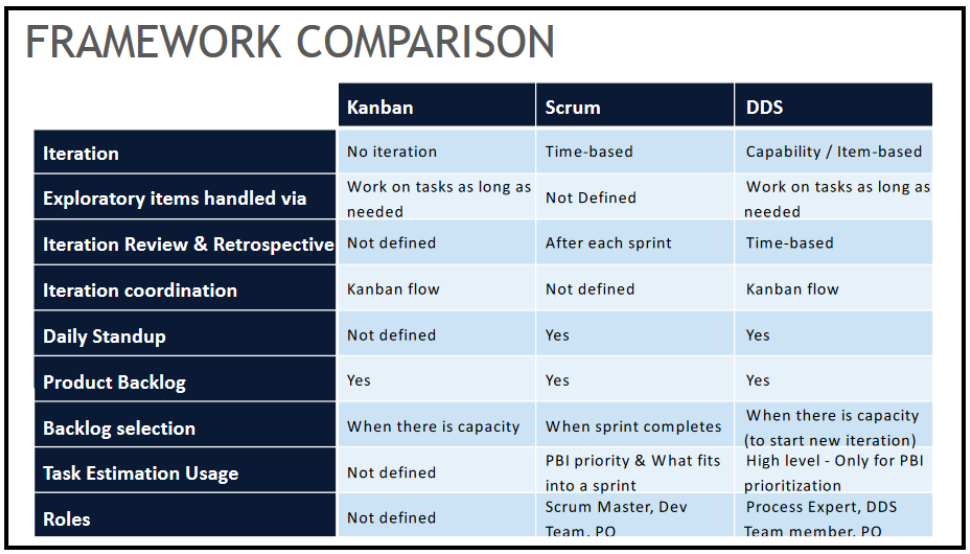
Like Kanban and Scrum, Data Driven Scrum can also be integrated with any of the data science workflows and used for data science projects

Let's continue using the same example and try to integrate DDS with CRISP-DM



In the Item Backlog, we can put tasks from the 6 phases of CRISP-DM, and then we breakdown each task/item in to Create, Observe and Analyze and then we prioritize it in the "To Do" then start working and move items to the "Doing" section then to "Validating" and finally to the "Done" section.

Benefits of Data Driven Scrum: Functional Iteration, Flexible Estimation, Collective Analysis and Iteration independent meetings.



FRAMEWORK COMPARISON

	Kanban	Scrum	DDS
Iteration	No iteration	Time-based	Capability / Item-based
Exploratory items handled via	Work on tasks as long as needed	Not Defined	Work on tasks as long as needed
Iteration Review & Retrospective	Not defined	After each sprint	Time-based
Iteration coordination	Kanban flow	Not defined	Kanban flow
Daily Standup	Not defined	Yes	Yes
Product Backlog	Yes	Yes	Yes
Backlog selection	When there is capacity	When sprint completes	When there is capacity (to start new iteration)
Task Estimation Usage	Not defined	PBI priority & What fits into a sprint	High level - Only for PBI prioritization
Roles	Not defined	Scrum Master, Dev Team, PO	Process Expert, DDS Team member, PO

Data Driven Scrum Survey

For the DDS framework, answer the following questions

1. On a scale of 1 to 5 (5 is highest) How likely are you to suggest using DDS
   Very likely (5 out of 5)

2. Do you think DDS is better or worse than Scrum (for data science project)?

   It's definitely better than Scrum because of so many reasons: Given what I know now which is DDS is not time boxed it is rather item/capability based weighs so much on my decision. Most of the time data science projects are explorations and you might not have any idea on how to produce the required product at first, and I can't imagine how forcing the data scientist to give an estimation will probably kill his/her creativity which is what we basically need to generate useful insights

   The fact that we have Iterations and Iteration reviews and retrospective makes me feel the scrum in DDS

   Another reason is having that kanban board as part of the framework saves teams lots of time, otherwise, you would have to choose the framework and the framework to get to what DDS already have if we go with any other framework. As far as I am aware, I think DDS is the only framework which has both

3. Explain your thoughts in how you answered these two questions.
   Answering the two questions was a no brainer for me. Even though I am not a Data Scientist right now, as soon as I learnt DDS it makes a perfect sense to me after I understood, how a data science project is very different from a a traditional software development project and

how data science is mostly exploration and there is so much unknown, in this case letting the data scientist to have the freedom to experiment and get creative and not force him/her to give a definite estimate will only increase the chance of us getting the insight we wanted to get and what that means is every one is happy and project is successful.

Data Driven Scrum all the way!!!!!

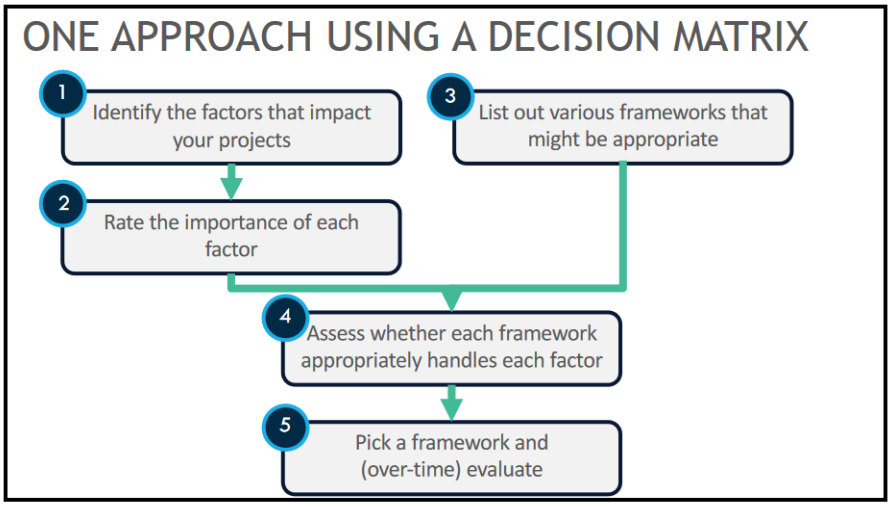
HOW TO SELECT A FRAMEWORK - HINTS ON HOW / WHY TO CHOOSE THE DIFFERENT ALTERNATIVES

Here are great examples of metrics

EXAMPLE METRICS

Category	Question	Example
Traditional metrics	How are we performing relative to plan?	Time, budget, and scope variance to plan
Agile metrics	How frequently are we providing value?	Cycle times (how fast work gets done) Lead time (time from request to delivery) Throughput (how much is delivered)
Financial metrics	Are we creating organizational financial value?	Revenue and cost metrics, payback period, ROI, NPV
Organizational goals	Is my project impacting organizational goals?	Varies widely
Artifact creation	Are we creating re-useable artifacts?	Number / value of artifacts created
Competencies gained	Are team members gaining valuable skillsets?	Number / value of competencies gained
Stakeholder satisfaction	Are my project stakeholders satisfied?	Net promotor score; "gut feel" assessment
Software metrics	What is the quality of the overall system being developed?	Defect count, defect resolution rate, latency, test coverage
Model performance	How are the models performing?	RMSE, F1, recall, precision, ROC, p-value

One approach can be using a decision tree



1. Identify factors that impact your project: list out all the factors that you think will affect your project
2. Rate the impact of each factor: Impact can be Low, Medium or High, label each factor with the rate
3. List out all frameworks that might be appropriate for this project
4. Then run the factors against each framework
5. Pick a framework and over time evaluate: so here we will be picking a framework that run against high impact factors and performed well, once we picked a framework we should be using it for some time and evaluate it to see if its working for us or if we have to change it.

Take a look at the below table which shows, the factors listed, the factors rated as low, high, medium or N/A and then run against the frameworks selected and finally pick a framework that performed best

SELECTING A TEAM PROCESS

After learning about DDS, the team adopted it. Relative to Kanban it provided more structure which the team's culture aspired to and it better supported business and documentation requirements

Factor	Level of Importance	CRISP - DM	Scrum	Kanban	Waterfall-Agile	Research - Agile	Data Driven Scrum
Culture (need for structure)	Low	●	●	⊙	●	●	●
Size (large)	High	○	●	●	●	○	●
Maturity (mixed)	Mod	●	●	●	●	○	●
Biz Requirements (unclear)	High	○	⊙	⊙	●	○	●
Doc Requirements (minimal)	Low	●	⊙	○	⊙	○	⊙
Extern Process Req. (minimal)	N/A	X	X	X	X	X	X
Resource Allocation (uncertain)	High	●	○	●	○	●	●
Expl Analysis (major focus)	High	●	○	●	○	○	●
Data Processing Req (minimal)	N/A	X	X	X	X	X	X
Data Coll & Clean (not a focus)	N/A	X	X	X	X	X	X

Key: ● Very Appropriate ⊙ Partially Appropriate ○ Not Appropriate × Factor was not important

The reason we have to choose from the different alternative is because, its only that way we know one is better than the other and that based on our most important high impacting factors, the chosen framework is good. If we don't have alternative to choose from, we might not know if our framework is the best framework out there or not.

AN ethics primer

Frequently asked questions (FAQ)

1. Describe the key differences between data science projects and software development projects

- **Age:** The field of software engineering is much older, stemming from the 1940s. And, although the concept of data analytical processes stems back centuries, the modern field of data science is much newer. The term "data science" itself didn't even crop up until the past two decades.
- **Organizational Understanding:** Related to the prior point, organizations often have less of a clear idea of what is possible from data science and what to expect from their data science teams.
- **Establishment:** Many complain that "data science" is a buzzword, and Merriam Webster doesn't even define the term. However, no one questions software engineering's existence as a distinct field.
- **Problem space**: Data science focuses on exploration and discovery (such as "finding insight in the data", and identifying new data sources that can be integrated into predictive models), while software engineering typically focuses on implementing a solution that addresses specific requirements (perhaps defined incrementally).
- **Domain Focus:** Although both fields rely on data, math, and code, data science emphasizes the data and math while software engineering is more heavily code-oriented.
- Typically, it has an ambiguous requirement it is not like I want a system which does A, B& C this is a typical software development requirement
- In software development, you will normally have an explicit requirement, you know what data you should use.etc, given this thing won't be clear incase of data science its hard to estimate how long it take

- Given how data science field is new and how most customers used to the traditional software development process, it is very hard to manage customer expectation, from not having a definite estimate as to how long it will take to the risk of not getting the required insight after spending all that time makes it hard for the customer to understand and for the DS team to manage expectations

2.Explain how to use several data science workflow processes (such as CRISP-DM and OSEMN)

3.Explain why agility is important for data science projects

Data Science work is exploratory and can be difficult to scope but agility can help the project because

- More Relevant Insights: By defining tasks just before analysis, the features are more likely to meet the most current needs
- Quicker Delivery of Customer Value: By delivering incremental product features, users gain value before the project's completion
- More Realistic Feedback: By soliciting feedback on the functional product, you can accurately assess whether their deliverables are of value
- Cut Losses from Building Wrong Features/Insights: Learn sooner if you're off course, cut your losses, and divert efforts elsewhere
- Improved Communication: Agile approaches promote close coordination and communication within team members and with stakeholders.

4.Articulate the key aspects of Kanban, Scrum, and DDS process frameworks

Kanban:is an implementationof leanwhich has 3 simple columnsTo Do, Doing & Done. When a team startsworking on a task, it willbe moved to Doing and when it is done it will be moved to Done. Kanban is flexible for us to add more columns and customize it to our needs. While Kanban is very helpful, it doesn't defineproject lifecycle, timeline or iterations and role.Scrum:ismost popularframework for software developmentwhere it focuses on a sprint and iterations to get things done.Transparency, inspection and adaptation are thepillarsof Scrum. Also it has values such as Commitment,courage, focus, openness and respect.In scrum the process is; there is a product backlog, then when its brokendown we have item back log, we have the sprint which can be set up by the team (weekly, bi weekly ...etv) and thenwe have the working incrementof the software. Unlike Kanban, it has roles, it is time boxed and there is a sprint review and retrospective.Onething I want to mention here, given ourexplanation above as to how estimating data scienceprojectsis hard, it will be difficult to use the timeboxed sprints with ScrumData Driven Scrum: It focuses each iteration on Create, Observe & Analyzeand it addressessome of thekey challengeswithusing scrumfor data science1.Use a visual board2.Focus on working on a specific item (or collection ofitems) during an iteration3.An iteration is task-based, not time-boxed4.Each iteration validates (or rejects) a specific leanhypothesis (or question)The process flow is: We have Item backlogging, Item breakdown where we breakdown each item what to do inCreate, Observeand Analyzephases. Then we have a Task board where we take individual items from Create,

Observe, Analyzeandprioritize in the To Do section and move to Doing whenDone when it is done and then the iteration back toItem

backlog. It has 3 roles, its item based not timebased and, it has iterations, iteration review and retrospective and I think this makes it standout from other Data Science frameworks.Also, I will keep this comparisontable becauseI think its very good in identifying key features and comparing it across

FRAMEWORK COMPARISON

	Kanban	Scrum	DDS
Iteration	No iteration	Time-based	Capability / Item-based
Exploratory items handled via	Work on tasks as long as needed	Not Defined	Work on tasks as long as needed
Iteration Review & Retrospective	Not defined	After each sprint	Time-based
Iteration coordination	Kanban flow	Not defined	Kanban flow
Daily Standup	Not defined	Yes	Yes
Product Backlog	Yes	Yes	Yes
Backlog selection	When there is capacity	When sprint completes	When there is capacity (to start new iteration)
Task Estimation Usage	Not defined	PBI priority & What fits into a sprint	High level - Only for PBI prioritization
Roles	Not defined	Scrum Master, Dev Team, PO	Process Expert, DDS Team member, PO

5.Leverage agile concepts within a data science project context

6.Select / use the most appropriate team process framework for a specific project

7.How a data science project can effectively work with the rest of an organization