

Data/AI Project Methodology

Chapter 1 : Data/IA Project Functional & Technical Methodology

Medina HADJEM

2022

Outline

- Context & Goals
- Data/IA Project Overview
- Data/AI Project Functional Methodology
 - Data/IA Project Strategy
 - Data/IA Project Lifecycle
 - Data/IA Project Teams
 - Data/IA Project Management
- Data/IA Projects Technical Methodology
 - Development Stage
 - Industrialization stage
- Key takeaways & conclusion
- References

Context & Goals

- **Context**

- Data/IA projects are specific and relatively recent in industries
- Data/IA project life cycle is different from software development life cycle
- Data/IA projects are multi-disciplinary and require various business, Analytics & IT skills
- Data is the heart of an AI project and mastering all of its aspects is a challenge
- 90% of Data/IA projects never make into production.

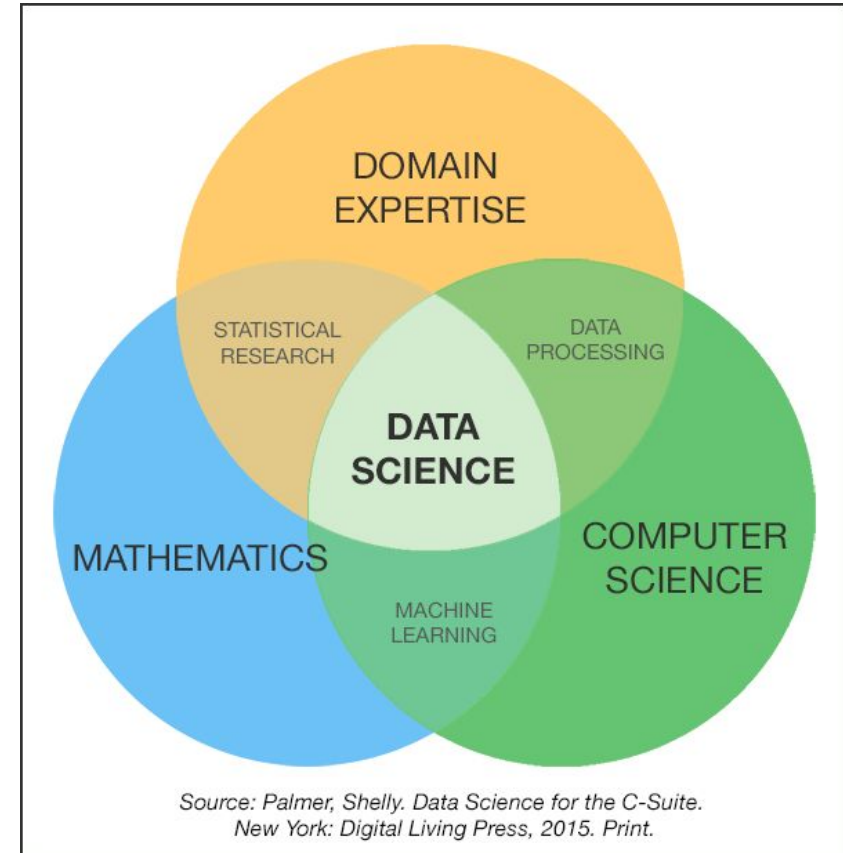
- **Goals**

- Understand the big picture of Data/IA projects and its challenges in organizations
- Understand how to go from a prototype AI model to an industrial product
- Understand the key skills and success factors of a Data/IA project
- Understand the specificities of managing a Data/IA project and be familiar with main frameworks
- Recommendations & best practices to take away

Data / IA Project Overview

What is a Data/IA Project ?

- The scientific exploration of data to extract meaning or insight, using statistics and mathematical models with the end goal of making smarter, quicker decisions.
- Data Science is a multidisciplinary field that uses scientific methods, tools, and algorithms to extract knowledge and insights from structured and unstructured data.

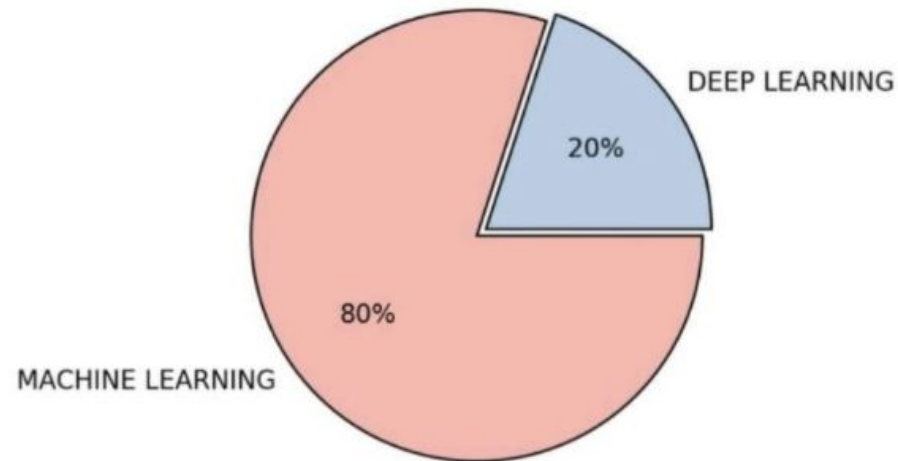


What is a Data/IA Project ?

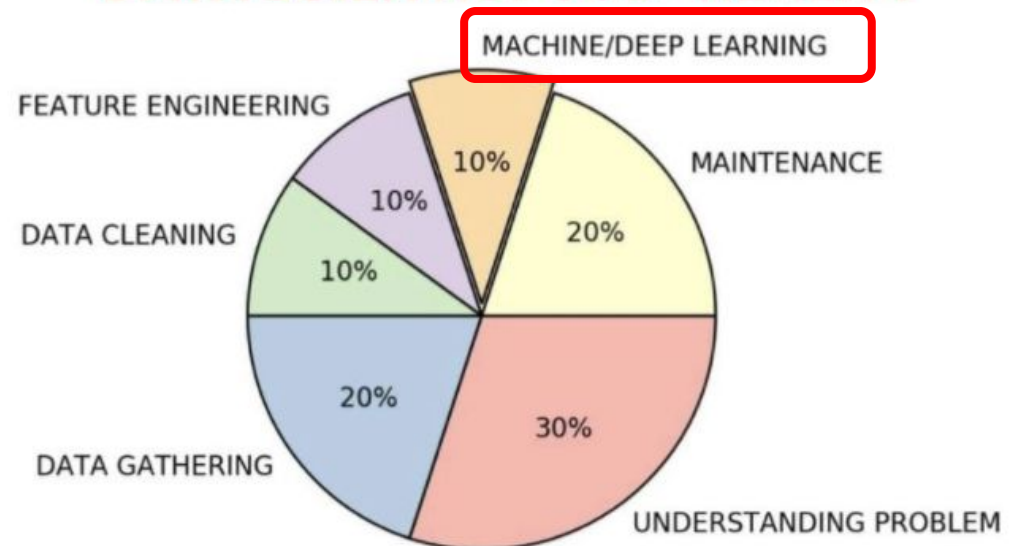
- It's not only about AI, ML and even Deep Learning !

DATA SCIENTIST JOB - EXPECTATION

@drangshu

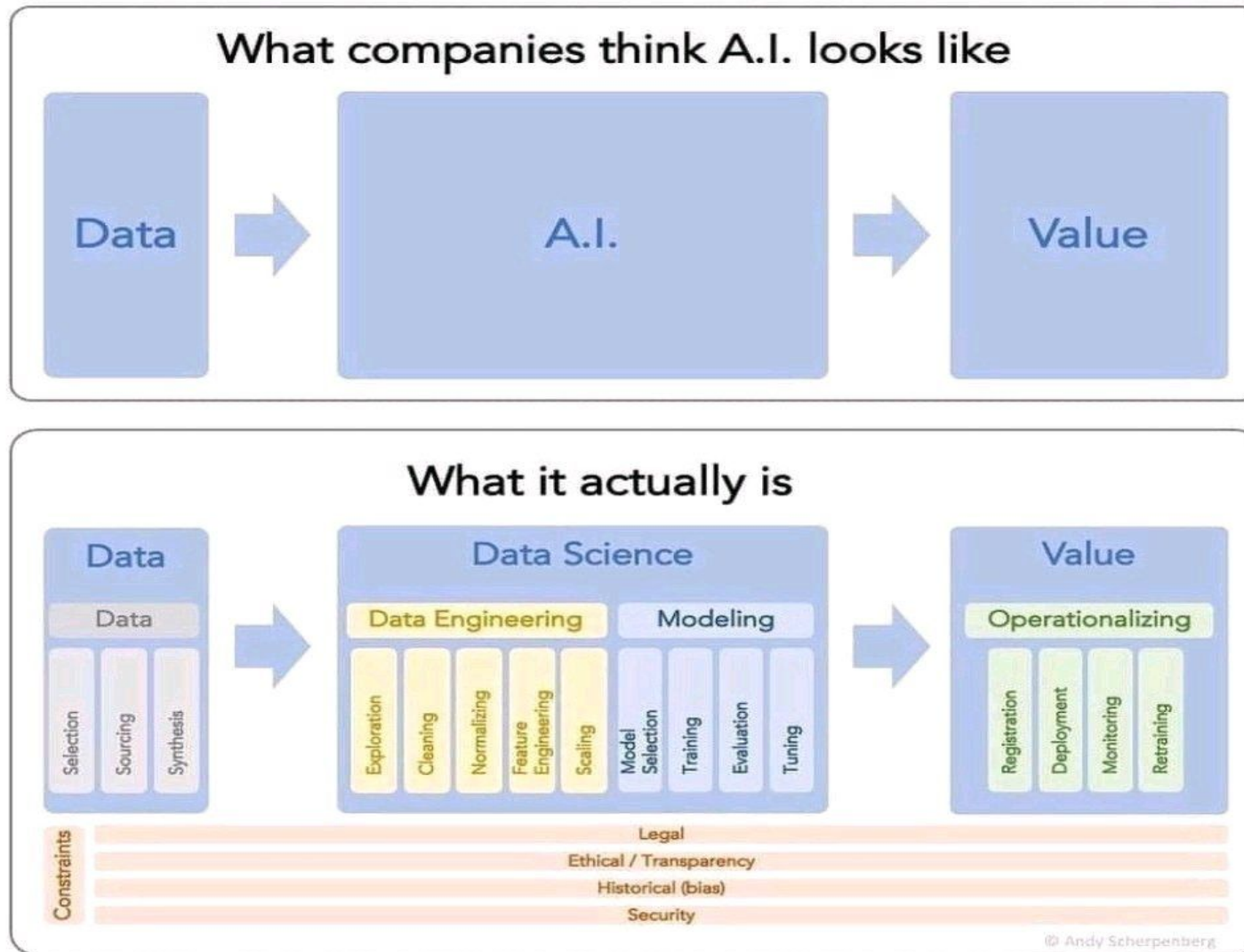


DATA SCIENTIST JOB - REALITY



Source : bit.ly/drangshu

What is a Data/IA Project ?



What is a Data/IA Project ?

- **What is the difference with a software development project ?**

- **Uncertainty**

- Data Quality / Quantity
- Results of the project (R&D process)
- Usage of the results

- **Scope Changing**

- Strict specifications is useless for Data/IA project
- Adaptive projects especially to Data

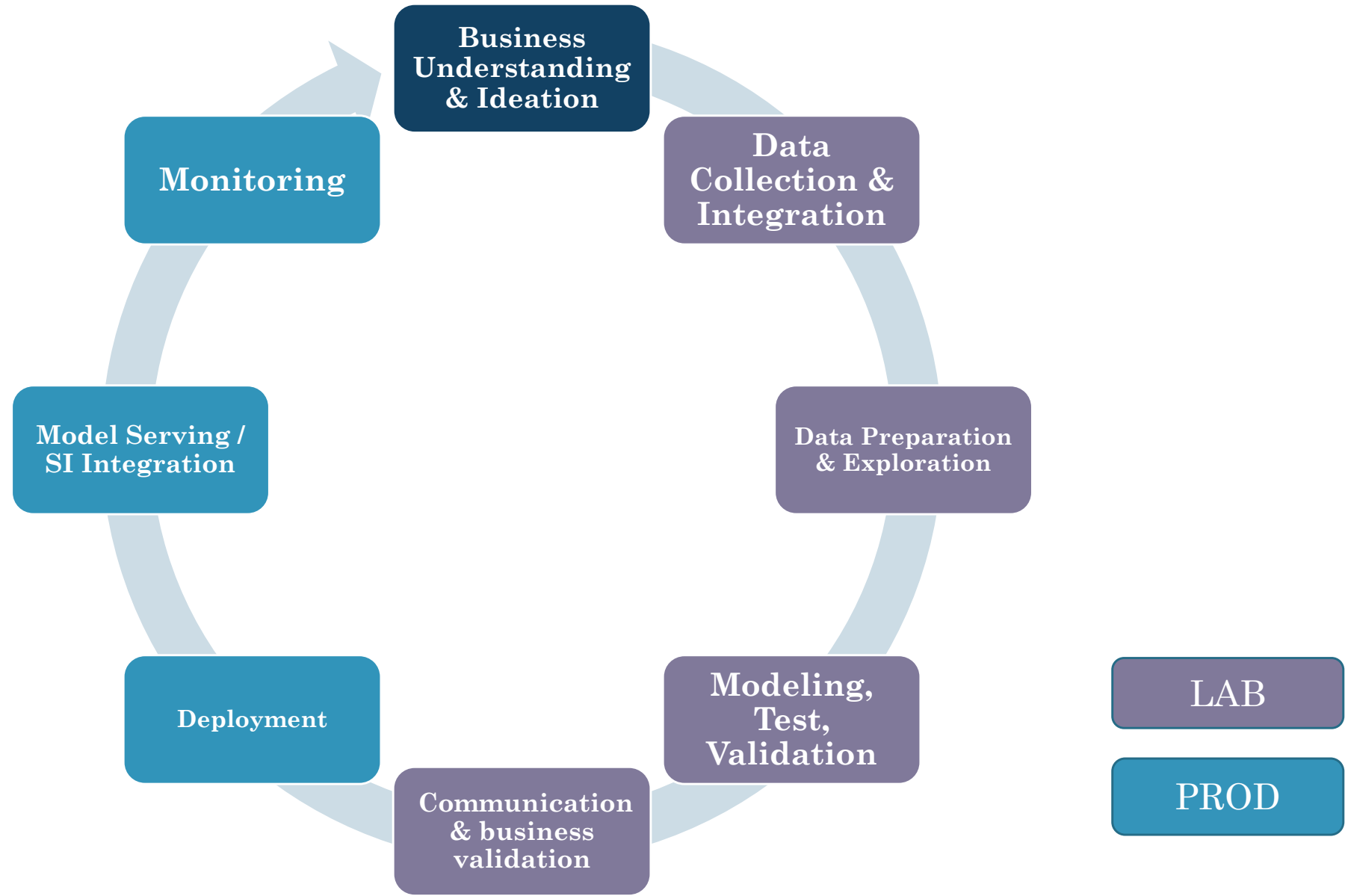
- **Iterative & Incremental**

- Generally starts with an POC / MVP*
- Product is improved with better data / better methods / better results

- **Question Answering Oriented**

- Frequent Q/A with business to challenge the need / communicate /validate the results
- Always having as final target a response to a business need with an ROI*.

What is a Data/IA Project ?



Data/IA Project Strategy

Strategy = Ends (Goals) + Ways (Actions) + Means (Resources)

Data/IA Project Strategy

- **The Ends - Goals**
 - Defining the correct objective
 - Estimate Cost of Failure
 - Unintended Risks
 - Big Data means also Big Risks & costs => Start first with small data
 - Listen to your customers (Ex : Use a Design Thinking Methodology)
 - What are Business constraints ?
 - What is the Priority / Urgency of the project ?
 - What are the consequences of a Model error ?
 - What volume of data is available to feed the model ?
 - What is required time for model results ?
 - How the model results will be accessible for users
- Choose an adapted Project Management Method

Data/IA Project Strategy

- **The Ways - Skills**

- Is the team well skilled for the project ?
- Are the people with the right skills available for the project ?
- Is a training required for the team ?
- What will be the consequences if the team is not well skilled ?
- How difficult would it be to hire a new resource ?
- How much of a delay would hiring cause to the project?

Data/IA Project Strategy

- **The Means - Data**

- Has the team worked with the project data before?
- What is the source of the data? does it currently exist within the organization, are you expected to effectively obtain this data ?
- What is the quality level of the data ?
- Would you get a better result with more data? How much would it cost to gather more data? How long would it take?
- Will you have permission to use the data when the model is implemented?
- When you implement the model, will the data be refreshed as often as you need the model to be refreshed?

Data/IA Project Lifecycle



Data/IA Project Life Cycle

- **Business Understanding & Ideation (1/3)**

- **Business Problem**

Example :“How can we identify customers who are more likely to buy our products?”

- **General Questions**

- What is the end result that you are trying to achieve? Why do this project?
- How will it help your client?
- How would life/business be different if the project is successful?
- Does it make sense for it to be a data science project?
- What methods have been tried the past? Why weren't they successful?

- **Problem specific questions**

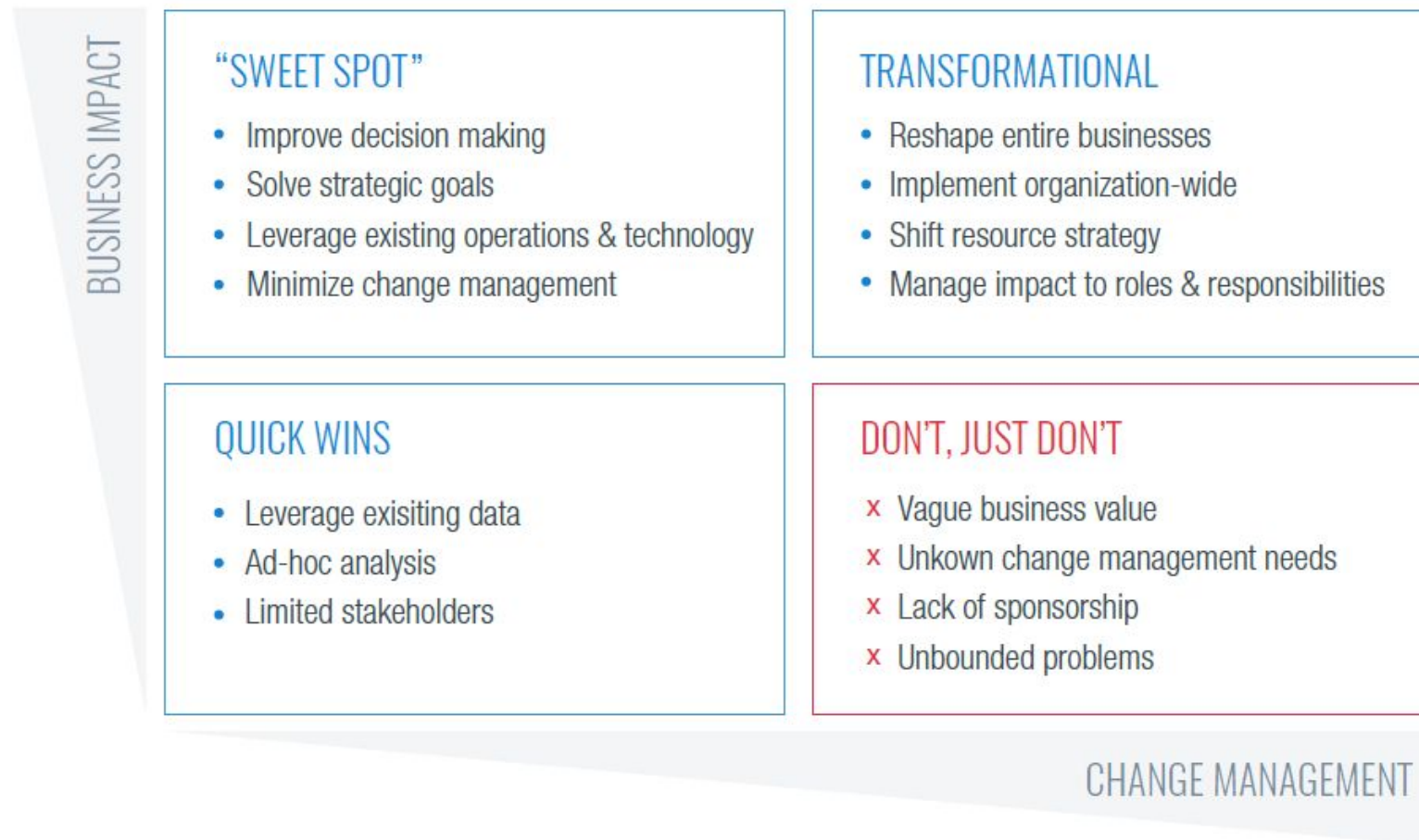
- Who are the target market and the customers?
- How do you approach the target market?
- How does the sales process look currently?
- What information do you have about the target market?

Data/IA Project Life Cycle

- **Business Understanding & Ideation (2/3)**
- **Value Estimation**
 - What is the expected value / ROI estimation ? In K€, in number of enrollments, in churn reduction, in customer satisfaction etc.
 - What is the project cost ? In K€, resources, time, change effort etc.
- **Feasibility Assessment**
 - Do we have the necessary prerequisites (Data, Technology, Resources, Budget, Time) ?
 - Do we have the ability/capability to use the results obtained ? to engage concrete actions ?

Data/IA Project Life Cycle

• Business Understanding & Ideation : Value Estimation (3/3)



- Source : Domino DataLab, <https://www.dominodatalab.com/wp-content/uploads/domino-managing-ds.pdf>

Data/IA Project Life Cycle

- **Data Collection / Integration (1/2)**

- **Data Understanding**

- Which Data is needed ?
- At which frequency the data is needed (real time, monthly, weekly, daily, other) ?
- Are data sources identified ?
 - Internal Data sources : Databases, Data Lake, Data warehouse
 - Third party
 - Web / Social Media
 - Open Data

- **Data Availability**

- Is data already available ? At the right frequency ?
- Is it available inside or outside the SI ?
- Does it require collection processes ?

Data/IA Project Life Cycle

- **Data Collection / Integration (2/2)**

- **Data Accessibility**

- Is Data technically accessible ?
 - Do we/can we have necessary habilitations / credentials to access this Data ?

- **Data Usability**

- Are there any technical issues using this data (Format, Documentation, etc.) ?
 - Are there any privacy issues using this data ?

- **Tools**

- ETL (Extract, Load, Transform)
 - SQL / HQL
 - Streaming Data ingestion (NIFI, Kafka, ...)

Data/IA Project Life Cycle

- **Data Preparation**

- Raw Data ☐ Parsed Data
 - Cleaning : errors, inconsistent formatting etc.
 - Missing values
 - Aberrant values
 - Merging Data Sets
 - Reducing Data

- **Data Exploration**

- Visualization
- Exploratory Data Analysis (EDA) : Statistics, Data distribution, feature correlation

☐ **In practice : depending on the data, ~70% to 80% of IA time is spent on this stage**

Data/IA Project Life Cycle

- **Modeling / Test / Validation**

- Modelization use case definition (Classification, Regression, Time Series, etc.)
- Features Engineering
- Method/Algorithm selection
- Model building (on training dataset)
- Model validation (on validation dataset)
- Model Evaluation (on test dataset)
- Model tuning
- Results Interpretation

- **Iterative/R&D process**

Data/IA Project Life Cycle

- **Communication**

- **What**

- Do the results make sense ?
 - Can we tell a story ?
 - What did we learn ?

- **How**

- With all the stakeholders (Business Team, Managers, Sponsors, ..)
 - Adapted at various level of expertise (business, technique,)
 - Synthetic : for example a report of few pages using appropriate visualizations
 - Adopt the storytelling way

Data/IA Project Life Cycle

- **Deployment**

- Deploy a Data/IA project into production environment
 - Deployment Pipeline building : prepare scripts, packages, requirements, resources.
 - With respect to all pre-deployment tests required for a **software engineering** project :
 - Unit tests,
 - Integration tests,
 - End to end technical and business tests.

Data/IA Project Life Cycle

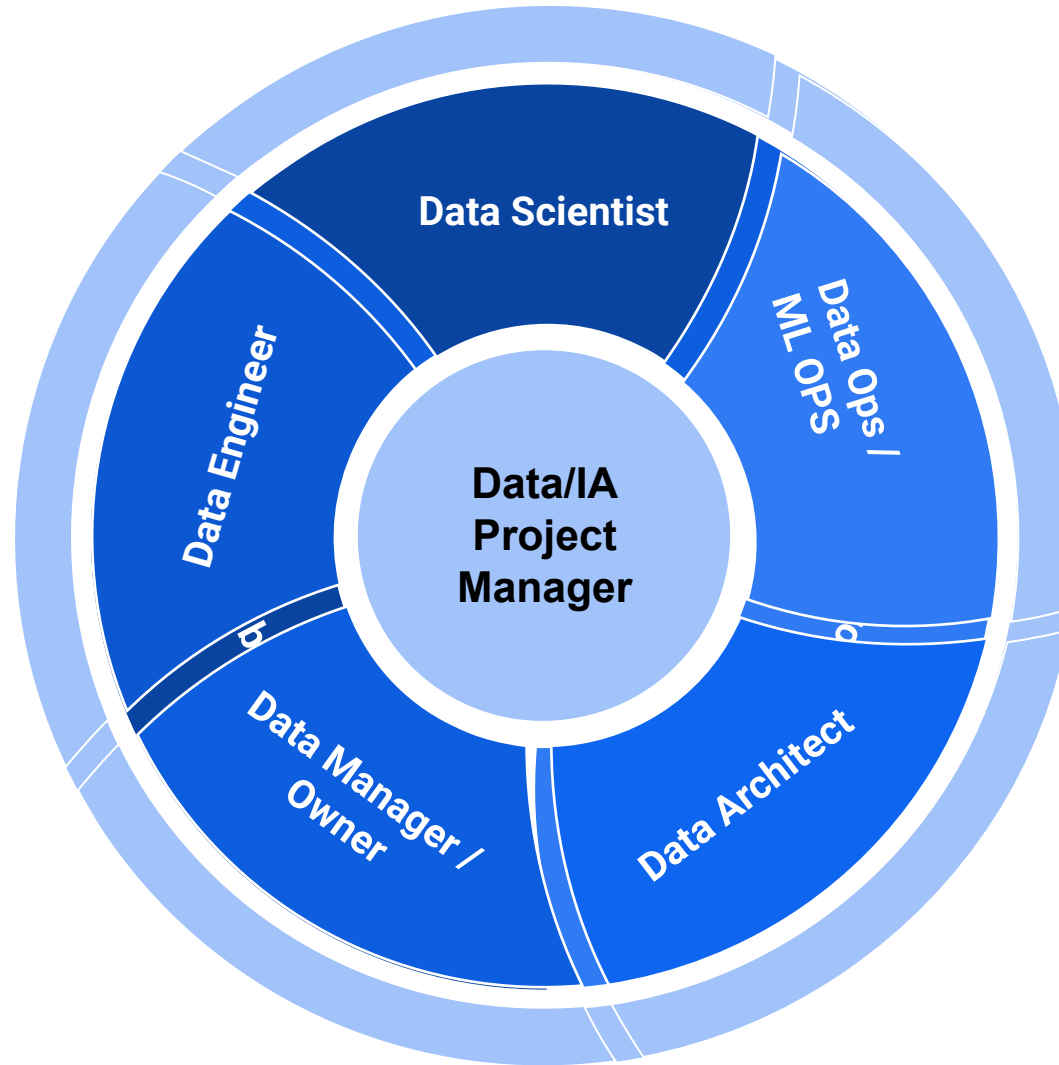
- **Model Serving / SI Integration**
 - **Embedded Model** : integrated to the consuming application
 - **Model Deployed as a separate service** : independently from the consuming application, remote invocation required for each prediction
 - **Model published as data** : independently from the consuming application but data is ingested at runtime

Data/IA Project Life Cycle

- **Monitoring / Backtesting**
 - Maintain/operate your project in production environment
 - **Data monitoring**
 - **Code/Programs monitoring**
 - **Models monitoring**
 - Models Performance / Backtesting
 - Models Retraining
 - Models Usage

Data/IA Project Team

Data/IA Project team : Roles



Data/IA Project Team : Hard Skills

Business Understanding

- Business Domain Knowledge

Data Understanding

- Data Domain Knowledge
- Data Querying
- Data Tooling (Data Dictionary, Data Catalog, Data Lineage, Data Quality...)

Data Exploration

- Data Querying
- Data Visualization
- Data Intuition and Analysis

Maths / Stats / ML

- Statistics / Mathematics
- Machine Learning / Deep Learning
- Optimization

Infrastructure

- System Implementation
- Platforms & Storage (Bigdata, Cloud, databases, ..)
- Devops

Software Engineering

- Programming Tools
- Front End / back End development concepts
- Web Services / API

Data/IA Project Team : Soft Skills

Soft Skills

- Communication
- Team Working
- Autonomy
- Curiosity / Learning

Data/IA Project Team : Roles vs Skills

Skills Matrix	Business Understanding	Data Understanding	Data Manipulation	Maths/ Stats / ML	Infrastructure	Software Engineering
Data Science Project Manager	★ ★ ★	★ ★	★ ★	★	★	★
Data Scientist	★ ★	★ ★	★ ★ ★	★ ★ ★	★	★ ★
Data Engineer	★ ★	★	★ ★ ★	★ ★	★ ★	★ ★ ★
Data Architect	★	★	★ ★	★	★ ★ ★	★ ★ ★
Data Manager	★ ★	★ ★ ★	★ ★ ★	★	★	★
Devops	★	★	★ ★	★	★ ★ ★	★ ★ ★

Data/IA Project Team : Shared Strategy

- **Team Context**

- How much motivation currently exists?
- What skills currently exist in the team? Are they a good match for the data that is usually available?

- **Alignment with Organization goals**

- Can we use the team's strategy to explain how the project helps the organization achieve its goals?
- If we follow the team's strategy, will I automatically achieve the organization's goals?
- Will the organization's strategy be stable over the intended period of the team's strategy?

- **Strategy documentation & communication**

- Make strategy document easily available to your team
- Ensure the formatting of the document make it easy to read
- Plan a communication of the new strategy with the team members and discuss how it relates to their individual work

- **Culture**

- What are the team's rituals? Do they help or hinder in achieving the team's goals? Do they make the team open to change, or do they reinforce a team filter bubble?
- Do this rituals include only the team or others around ?

- **Acting for your strategy**

Data/IA Project Team Efficiency

- A shared way of doing things
 - **Standard definitions of target variables:**
 - create a shared team vision with input from everyone on the team
 - **Standard terminology:**
 - Standard terms for data/AI concepts and business concepts for use within the team
 - Standardized understanding of priorities
 - **Standard Tools:**
 - You've probably decided on a standard platform/language, but if you've gone for R or Python for example, have you standardized on preferred libraries for particular common tasks?
- **The skills your team needs**
 - Asses the skills required in your data/IA team, considering the skills that are available elsewhere in your organization to ensure your team members develop the right skill set
- **Learning from previous projects**
 - Team retrospective process

Data/IA Project Communication/ Promotion

- **Promote the Data/IA Project**
 - Whitepapers
 - Talking about your work (local meetup, internal events, ..)
 - Presenting to the outside world (blogs, conferences, ..)
 - Regular presentation of team progress to the business
 - Differing audiences for documentation

Data/IA Project Management Frameworks

Data/IA Project Management Frameworks

- Framing each Step of the Data Science Life Cycle is **Crutial** for project success.
- Many Framework exist for DS projects Framing

- **Traditional Frameworks**

- Waterfall
- CRISP-DM

- **Agile Frameworks**

- SCRUM
- KANBAM

- **Hybrid Frameworks**

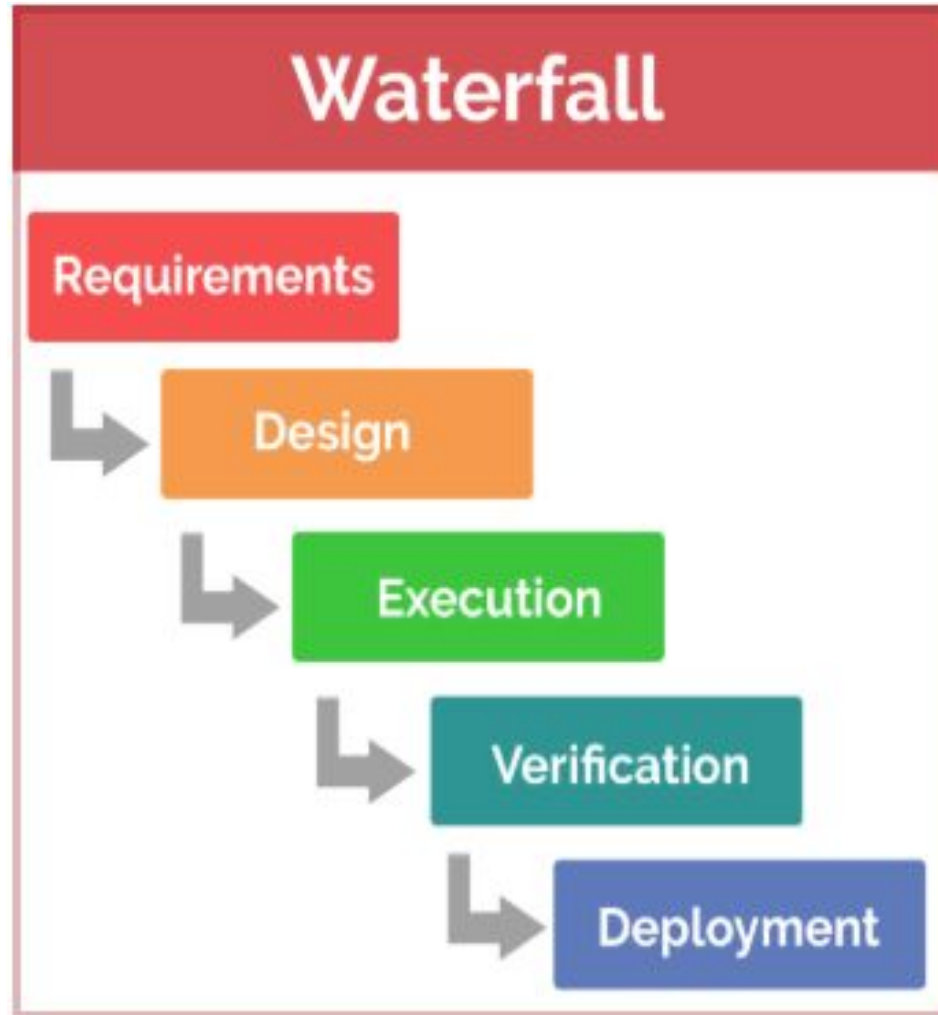
- Waterfall-Agile
- R&D

- **Emerging Frameworks**

- TDSP
- Domino DataLab

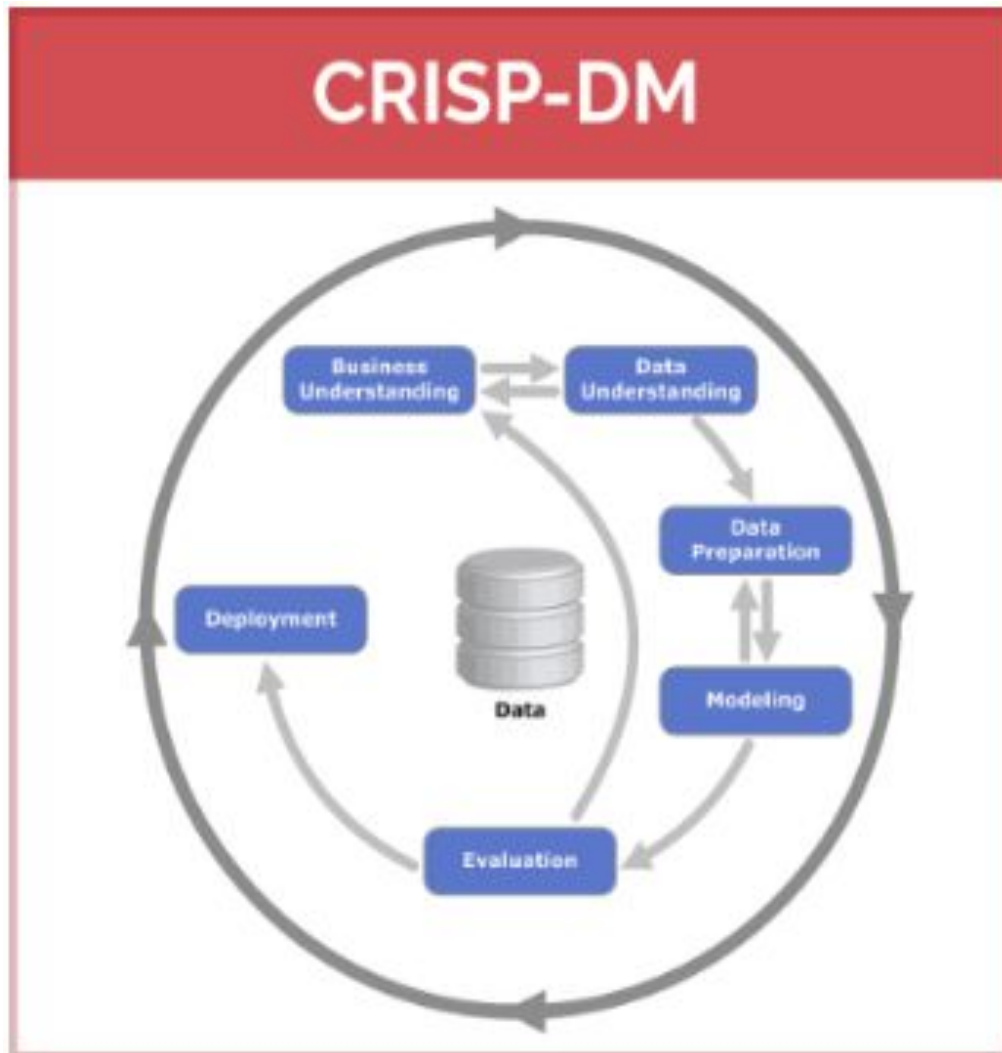
- Source : <http://www.datascience-pm.com/pm-guide-overview/>

Data/IA Project Management : Traditional Framework



- **Traditional software development life cycle (SDLC)**
 - Originally from manufacturing and construction and was applied to software engineering projects in the 1960s
 - Highly-structured
 - Horizontally-layered development phases
 - Extensive up-front planning (often based on a Gantt chart)
 - Commitment to follow the plan

Data/IA Project Management : Traditional Framework

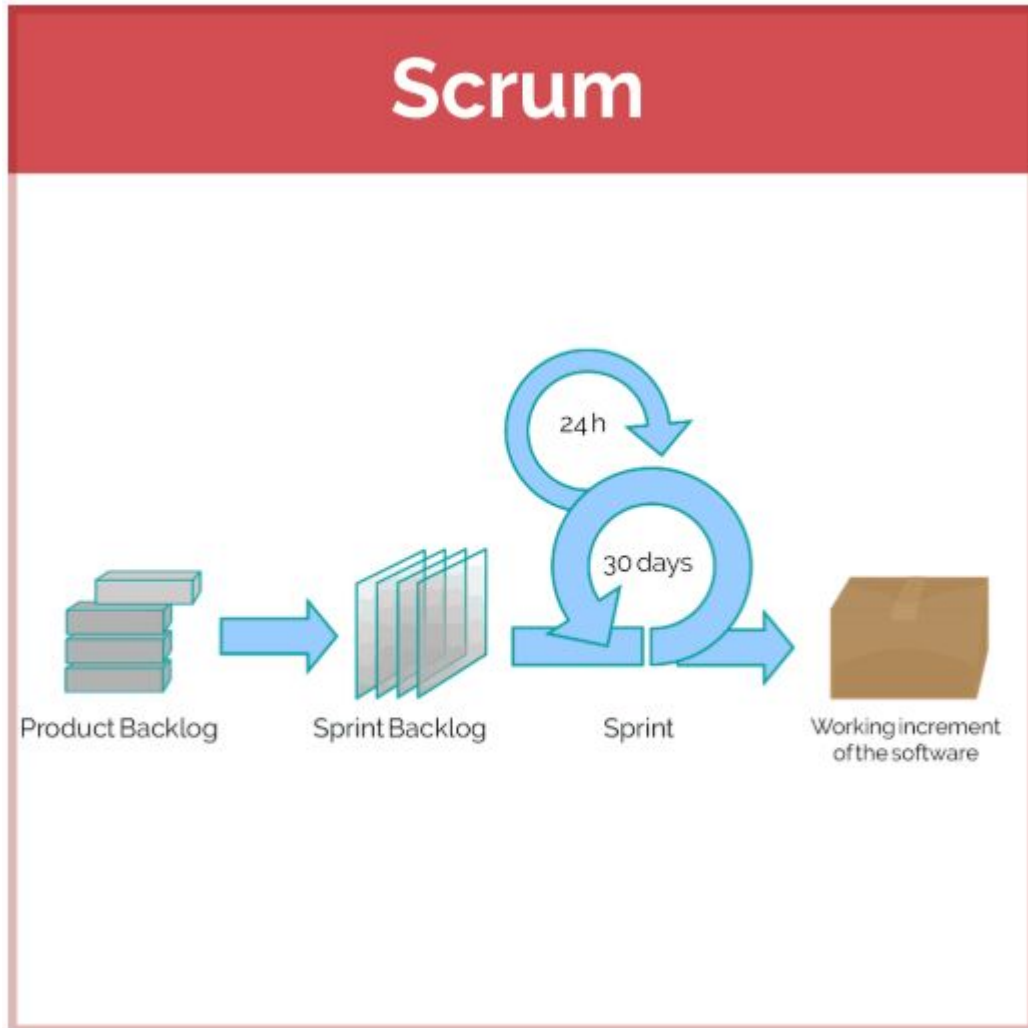


- **The C**ross-Industry standard process for data mining (**CRISP-DM**)
- First widely-accepted process methodology for data mining
- Most popular Knowledge Discovery in Database (KDD) methodology today
- Task focused and lacks team-based processes
- Phased approach with heavy documentation
- Can be easily implemented in organizations

Data/IA Project Management : Traditional Framework

Approach	Description	Strengths	Challenges	Best For...
Traditional Approaches				
Waterfall	<ul style="list-style-type: none"> • Set your plan up-front in detail, lock it in, and follow your plan 	<ul style="list-style-type: none"> • Simple, well-organized, and easily understood • Matches traditional corporate culture 	<ul style="list-style-type: none"> • Inflexible • Not suitable for data discovery processes • Delayed testing phases increase risk • Heavy documentation 	<ul style="list-style-type: none"> • When requirements and technology are known and aren't likely to change (Rarely a fit for data science)
CRISP-DM	<ul style="list-style-type: none"> • Break data science process into six iterative phases 	<ul style="list-style-type: none"> • Natural process for data science • Easy to use • “De facto” process w/ long track record 	<ul style="list-style-type: none"> • Does not prescribe teamwork processes • No update since 1990s • Phased approach like waterfall 	<ul style="list-style-type: none"> • Individuals or small teams • Teams looking for an established practice • Use with agile processes

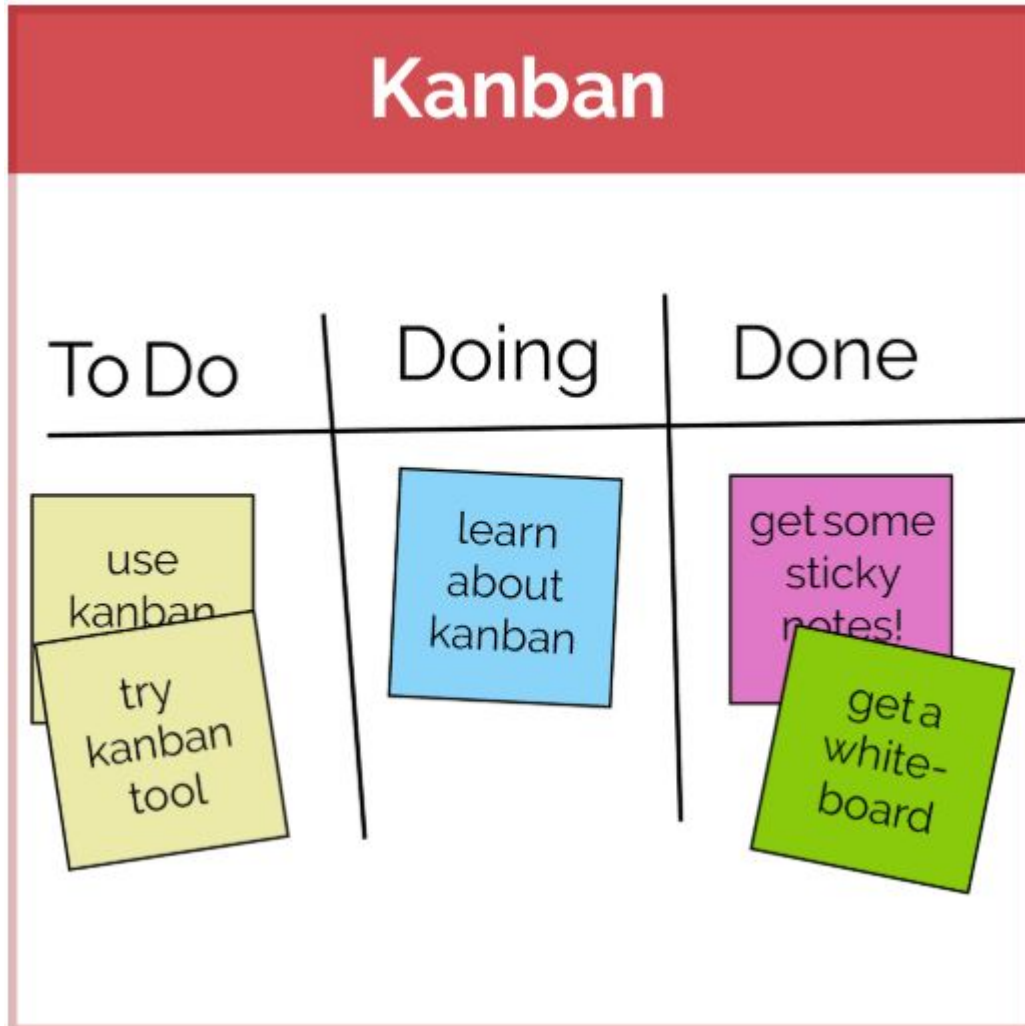
Data/IA Project Management : Agile Framework



• SCRUM

- Founded in 1990
- Most popular Software engineering agile approach
- Based on Scrum Guide
- Defines 3 roles (Product Owner, Scrum Master, Development team)
- Divides projects into fixed length tasks (sprints) and manage them based on ceremonies (sprint planning, review, ..)
- Rigid time boxing may be an issue for data science projects

Data/IA Project Management : Agile Framework



- **KANBAN**

- Supply chain and inventory control system for Toyota manufacturing in the 1940s
- Project is divided in a list of features (TO DO)
- More flexible than scrum
- Can be effective for several DS projects especially if combined with other processes

Data/IA Project Management : Agile Framework

Approach	Description	Strengths	Challenges	Best For...
Agile Approaches				
Scrum	<ul style="list-style-type: none"> • Develop potentially shippable increments during short, iterative cycles • Empower teams 	<ul style="list-style-type: none"> • Adaptive • Strong customer feedback loop • Builds sense of team ownership 	<ul style="list-style-type: none"> • Challenges cultural norms • Adhering to sprint time-boxing • Challenging to implement 	<ul style="list-style-type: none"> • Agile teams who need discipline provided by fixed time cycles • Radical innovation cultures
Kanban	<ul style="list-style-type: none"> • Visualize workflow • Decrease cycle times and work in progress • Implement small, continuous changes 	<ul style="list-style-type: none"> • Very flexible • Easy to use • Improves coordination 	<ul style="list-style-type: none"> • Does not prescribe customer interaction • Kanban columns tricky for data science 	<ul style="list-style-type: none"> • Teams transitioning to agile • Process-oriented teams who don't need many prescribed practices

Data/IA Project Management : Hybrid Approaches



- **BIMODAL : Waterfall-Agile**
 - Combine the best of waterfall & Agile
 - Pure Agilists discredit their use
 - Can be useful for Data/IA projects with specific constraints like regulation or organizational policies.

Data/IA Project Management : Hybrid Approaches



- **R&D Process**

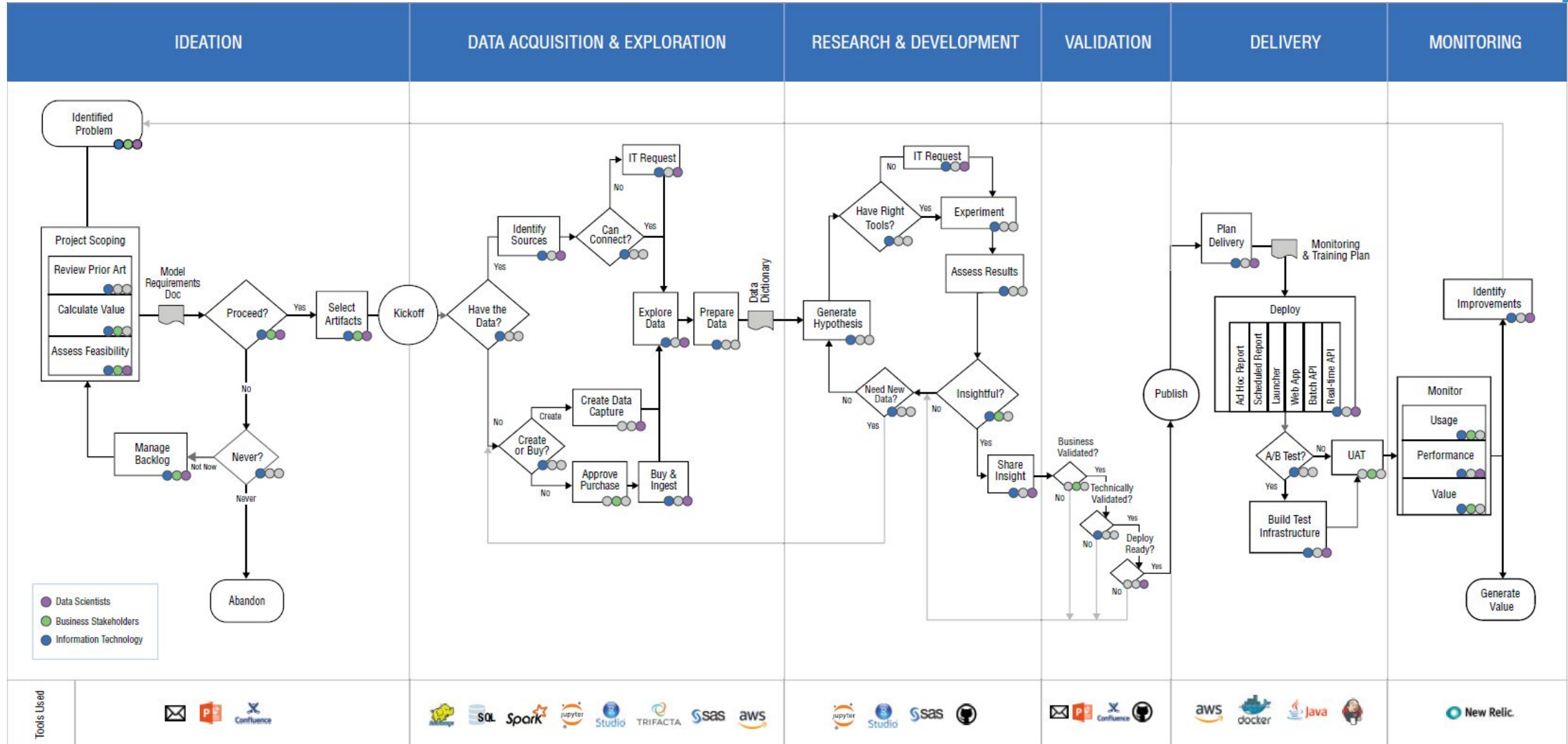
- Data science process can also be viewed as a research project that transitions into an engineering project
- Combines Research process (unstructured) and development process (structured)
- Particularly appropriate for mature teams (Ex: Google Brain)

Data/IA Project Management : Hybrid Framework

Approach	Description	Strengths	Challenges	Best For...
Hybrid Approaches				
Bimodal: Waterfall- agile	<ul style="list-style-type: none"> Combine best practices from waterfall and agile 	<ul style="list-style-type: none"> Can be tailored to specific team needs 	<ul style="list-style-type: none"> Often poorly implemented Negative reputation 	<ul style="list-style-type: none"> Specific situations like highly-regulated projects that require some waterfall elements
Research & Development	<ul style="list-style-type: none"> Treat data science as “research” Once problem is understood, then transition to “development” 	<ul style="list-style-type: none"> Does not try to force a methodology onto data science Comfortable for data scientists 	<ul style="list-style-type: none"> Difficult to monitor High trust and discipline required Could suffer from being too ad hoc 	<ul style="list-style-type: none"> Mature teams who don’t need heavy oversight Research-focused teams needing freedom

Data/IA Project Management : Emerging Framework

2021-2022



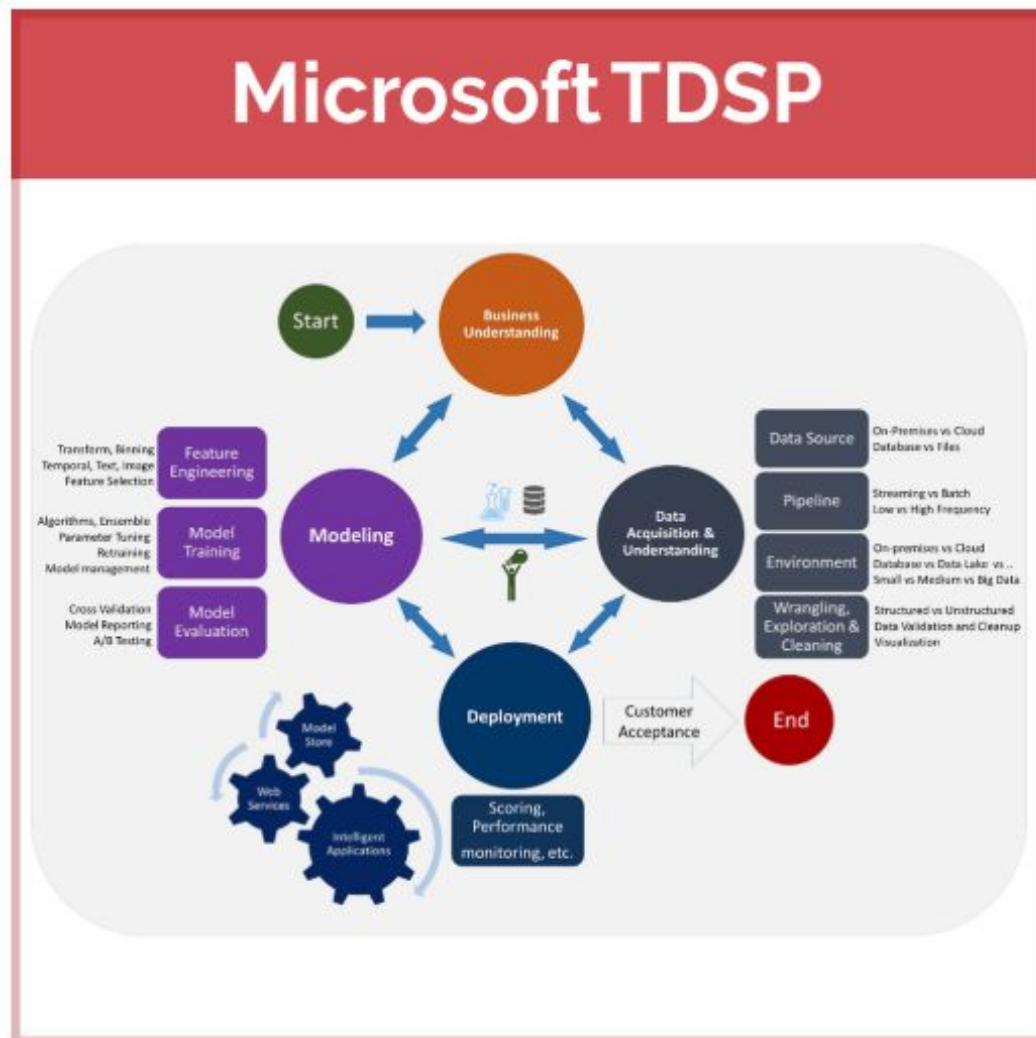
- Source : Domino DataLab, <https://www.dominodatalab.com/wp-content/uploads/domino-managing-ds.pdf>

Data/IA Project Management : Emerging Framework

- **DOMINO Data science life cycle**

- Introduced by DOMINO Data Lab (Silicon Valley DS Platform Provider)
- Dedicated to Data Science Life Cycle (from ideation to delivery & monitoring)
- Combines
 - CRISP-DM
 - Agile concepts
 - 20 teams practice experience
- Based on three principles
 - Expect and embrace iteration
 - Enable compounding collaboration
 - Anticipate auditability needs

Data/IA Project Management : Emerging Framework



- **Microsoft Team Data Science Process (2016)**

- Largely inspired from CRISP-DM and Scrum.
- Project Lifecycle framing similar to CRISP-DM
- Include team definition
- Standardized resources
- Detailed Documentation
- Most mature CRISP-derived project management

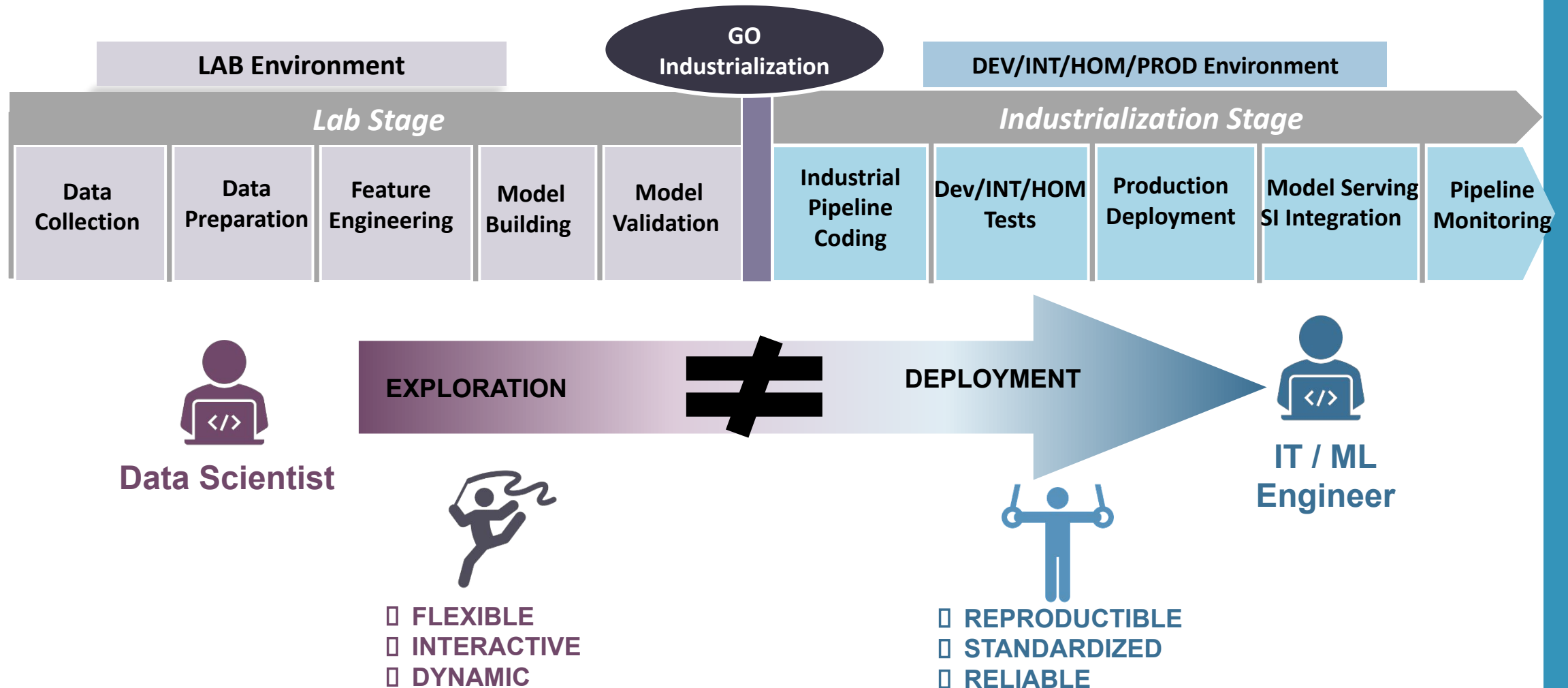
Data/IA Project Management : Emerging Framework

Approach	Description	Strengths	Challenges	Best For...
Emerging Approaches				
TDSP	<ul style="list-style-type: none"> Combine CRISP-DM and Scrum practices and tailor to data science 	<ul style="list-style-type: none"> Comprehensive open-source documentation that defines processes, templates, and team roles 	<ul style="list-style-type: none"> Not yet publicly vetted with success track record 	<ul style="list-style-type: none"> Medium-large projects whose teams seek a well-defined process to follow
Domino Data Lab	<ul style="list-style-type: none"> Combine CRISP-DM and general agile practices and tailor to data science 	<ul style="list-style-type: none"> Similar to TDSP but with less role and template definition 	<ul style="list-style-type: none"> Not yet publicly vetted with success track record 	<ul style="list-style-type: none"> Medium-large projects whose teams seek a process flow without full definition

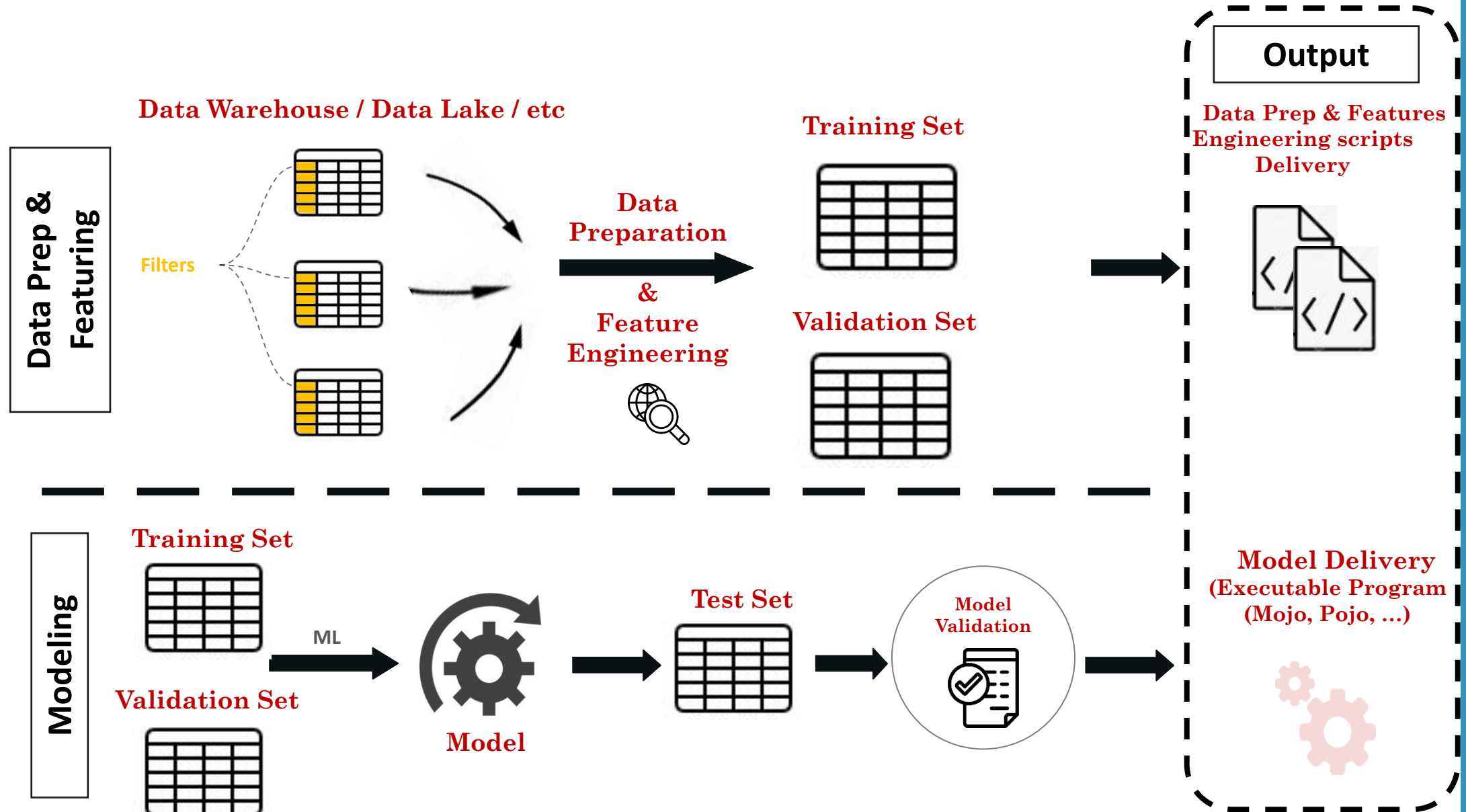
Data/IA Project Development & Industrialization

Data/IA Project Life Cycle :

2 main stages



Data/IA Project Development



Data/IA Project Development :

Coding best practices

- Versioned code
- Reproducible code
- Quality Code
- Documented Code
- Tested Code
- Coding responsibilities

Data/IA Project Development : Coding Best Practices

Versioned Code

- **Use GIT to version control your code**
 - Create a repository at the beginning of the project
 - Define a branching workflow and stick with it
 - Create small and well-defined branches
 - Push/commit regularly, at least daily
- **Create clear guidelines for code development**
 - Define a GIT branching workflow^{1,2}
 - Implement rules for merging code³
- **Data Storage and versioning**
 - Data is stored in a single repository
 - Naming convention for versioning
 - Define a data versioning scheme and keep track of the version used to train each milestone model

1. <https://www.atlassian.com/git/tutorials/comparing-workflows>, <https://nvie.com/posts/a-successful-git-branching-model/>

2. <https://guides.github.com/introduction/flow/>

3. <https://help.github.com/en/articles/configuring-protected-branches>

Data/IA Project Development : Coding Best Practices

Versioned Code : Git best practices

1. Commit and push often: aim for daily at least
2. Don't push any credentials to the repo, use “.gitignore”
3. Don't use Git for data versioning
4. Keep features and PRs specific and small (<500 lines)
5. Write clear, action-oriented commit messages¹
6. Delete branches after merging them
7. Delete old/dead code, use git tag
8. Pull often to ensure you have a local up-to-date copy
9. Write short but clear branch names

Data/IA Project Development : Coding best practices

Reproducible Code

- Code that runs easily on new machines scales faster
 - Easy installation : Aim for few installation steps and little configuration
 - Accessible : All team members are able to run the code
 - Abstracted : Code agnostic to the directory structure and infrastructure scales to larger datasets easier
- Structure your notebooks
 - Structure and create modules
 - Import your modules in notebooks

Data/IA Project Development : Coding best practices

Reproducible Code

Do's

- Create a reproducible environment (**conda create** or **pipenv**)
- Export an environment file with specific package versions and include it in version control
- Run code from a single entry-point (**main.py** or similar)
- **Use an IDE** for development such as PyCharm or VSCode (or Rstudio for R) to leverage functionalities such as breakpoints, watch variables, remote debugging
- Test early on a environment similar to the target production one
- Use relative file paths (**./data/results**)
- Define **parameters for a run in a config file** (see **ConfigArgParse** for example)
- Specifies the seeds of the random number generators

Don'ts

- **Keep stable code in notebooks**
- Use hardcoded paths in the code (**C:\MyProjects\ProjectXXX\Data\data.csv**)
- Develop solely on your local machine
- Assume it is IT's problem to run your code. It's a shared responsibility!

Data/IA Project Development : Coding best practices

Reproducible Code

- Move code from notebooks to .py files early and take advantage of IDE functionalities
- Test your code on a project VM or the Data Lake on a regular basis. Or even better, develop remotely.
- No hardcoded file and folder paths in the code
- Create a env.yml for easy deployments on new computers
- Log Git commit hashes associated with key result files

DS Project Development : Coding best practices

Quality Code

- Begin a project by setting up a directory framework, cookiecutter are good to automatize project startup : <https://drivendata.github.io/cookiecutter-data-science/>
- Modular code is easier to maintain, scale, understand, and test (decouple data ingestion, data preprocessing, model training steps, etc)
- Formatting code makes it easier for other people to work on your code
 - PEP8 (Python Enhancement Proposal) is the Python standard, use it!
 - IDE environments such as PyCharm and VSCode will automatically provide code style suggestions
 - Enforce code style and quality conventions by creating a pre-commit linter¹ that automatically blocks PRs that violate code style
 - Google has published a helpful style guide² (exists also for R)

1. Use the pre-commit library along with Black and PyLint / Flake8

2. Google Python Style Guide : <https://google.github.io/styleguide/pyguide.html>

Data/IA Project Development :

Coding best practices

Quality Code

- 1 Avoid global variables
- 2 Use concise and descriptive variable names
- 3 Vectorize or use list comprehensions instead of for loops
- 4 Dictionaries are fast, nested lists are slow
- 5 Avoid nested functions with multiple levels of if statements
- 6 Use descriptive file names : train.py vs Untitled2_4.ipynb
- 7 Use asserts and logging
- 8 Lint¹ your code

1. A tool analyzing source code to flag potential bug and stylistic errors (e.g. PyLint, linter)

Data/IA Project Development : Coding best practices

Documentation

- Start documentation early and keep it up to date
 - Keep documentation tightly coupled with code
 - Docs follow same workflow as code
 - Write for both technical and non-technical audiences
- Creating documentation is easiest with Python packages
 - Standardize docstrings : PEP 257 - Docstring Style Guide for Python : <https://lnkd.in/g2PThdq>
 - Automatically generate organized HTML or PDFs from code, documentation, and docstrings : Sphinx - Python Documentation Generator : <https://www.sphinx-doc.org/en/master/index.html>

Data/IA Project Development : Coding best practices

Documentation : easier with python packages

- Sphinx has many extensions to speed up documentation generation
 - Extract docstrings from .py files (autodoc)
 - Automatically find the .py files in a given directory (apidoc) and generate a table of contents
 - Include mathematical formulas, web links and images (snapshots of slides explaining the case logic for example)
- You can setup Git repository to auto-deploy doc changes
- Regardless of tool, always document functions, classes, and modules in your code
 - Choose a style such as NumPy's docstrings1
 - Documentation usually lives in a ./docs directory

1. https://sphinxcontrib-napoleon.readthedocs.io/en/latest/example_numpy.html

Data/IA Project Development :

Coding best practices

Documentation

Do's

- Start the documentation early and keep it up to date
- Use a toolkit like sphinx to link code and documentation
- General explanations contain significant value: case logic, overall code structure, main process steps, ...
- Documentation includes detailed descriptions about how to install the project and run it
- Usage examples for functions and objects

Don'ts

- Avoid obvious comments
- Not enough details for anyone but the developer himself

Data/IA Project Development :

Coding best practices

Test your Code

- **Tools**

- Directory structure contains a test folder (see Code best practices)
- Pick a testing framework such as Nose or Pytest (or testthat for R)
- Define on naming conventions (test_*)
- Setup IDE to automatically run tests
- Setup automated testing suites to execute on every pull request, examples: GitLab CI, Jenkins
- CI tooling catches mistakes before they become a problem and ensures consistently high code quality

- **Write test for all your code**

- Tests run quickly and independently of each other
- Tests run on a very small subset of sample data, a few rows are often enough
- Tests are easier to write when I/O (reads, writes, SQL) is outside the function
- Testing for fully identical outputs is not always appropriate as the models change
- Check the proportion of null output values
- Check simple statistics of output values (mean, range, etc.)
- If you're behind on tests, begin by writing an integration or end-to-end test, and add unit tests whenever code is modified or refactored. Whenever you encounter a bug, write a test for it

Data/IA Project Development : Coding best practices

Coding responsibilities

- Roles in charge of code and data quality should be defined on a datalab or project basis
 - **Code Master**
 - Sets up and manages overall code structure
 - Owns codebase
 - Supervises Git repo
 - Regularly reviews code
 - Defines testing protocols
 - Makes sure team agrees on code review
 - Prevents code duplication
 - **Data Master**
 - Creates and updates overall data structure
 - Understands data sources, provenance and governance
 - Establishes data quality checks
 - Ensures data versioning
 - Prevents data duplication

Data/IA Project Development : Coding best practices

Coding responsibilities

- Code reviews happen regularly, at least whenever branches are merged and versions are tagged and it benefits to all :
 - The Project
 - Great bug fighting practice
 - Project knowledge is better spread within the team
 - The Author
 - Receive concrete peer advices
 - Help to take a step back on project logic and code structure
 - The Reviewer
 - Learn new technics by examples
 - Develop mentoring skills

Data/IA Project Development : Coding best practices

Synthesis

- **Versioned:** Use Git to save changes, manage snapshots, and provide the ability to revert, named schema for data versioning
- **Reproducible:** Code easily runs on another computer without significant effort. Prior analysis (code AND data) can be recreated
- **Standardized:** Project structure, coding convention, and tooling follow best practices. Code is easy to execute and extend
- **Documented:** Automated documentation keeps end-users and data scientists up-to-date
- **Tested:** Unit tests in place, ideally including a continuous integration process
- **Clear responsibilities:** Define code and data master roles in each datalab or project

Data/IA Project Industrialization

• **What an Industrialized Data/IA application/product ?**

As a software application, a Data/IT application is industrialized (in production) when:

- It serves the company's daily business.
- It evolves constantly with changing business needs.
- It is stable.
- It is well integrated with other systems.
- Its Integration/deployment is agile and widely automated.
- It is monitored

Data/IA Project Industrialization : Specific challenges

- Most of the data/IA projects remain in the Prototyping/Lab stage (turn prototypes into production applications is often a challenge), Why ?

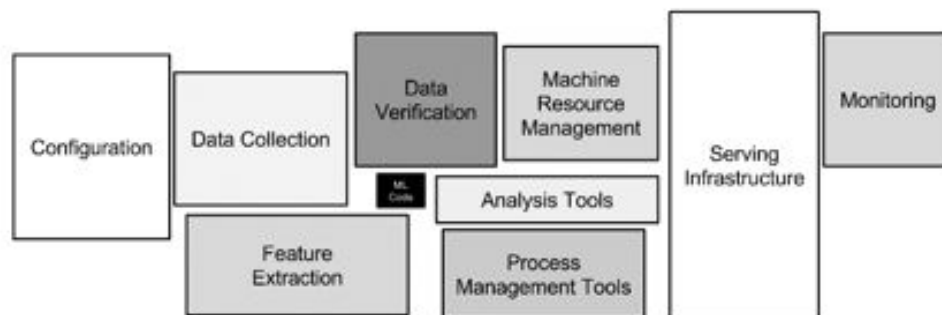


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

- A Data/AI applications is a sub-group of software applications.
 - How to integrate the Data/AI part into the company's vast and already complicated IT system?
- AI/ML models' performance drifts over time.
 - How can we monitor models' performance and re-train them so that we can keep a satisfying quality of our production services?

Data/IA Project Industrialization : Specific challenges

- **Technical Challenges**

- **Code/Programs**

- Data/IA projects code is often NOT production level code.
 - Libraries of AI programming languages are often not installed/managed in Production

- **Models**

- ML/AI models management (tracking and storage)

- **Data**

- Availability / Accessibility / Governance / Quality in Production

- **Functional Challenges**

- We can't graft the data/IA lifecycle directly onto the software development lifecycle

Data/IA Project Industrialization



Data

+



Model

+



Code

Data :

Data can be :

- Labelled data for train and evaluation
- Unlabeled data for prediction mode
- Structured / Unstructured

Model :

Set of parameters used by a AI/ML algorithm

- Created ('learned') through a 'training' process
- Distributed as a binary file loaded by the machine learning code in « Prediction mode » (usually large – 10, 100, 1000 Mo)

Code :

- Python/Scala/R/Other files executed to train a AI/ML algorithm and run it for prediction (the code for train and predict is necessary the same)
- The size of the code files is usually very small (a few Ko)

Data/IA Project Industrialization :

Main steps

- **Architecture**

- Generally designed for production environment
- End to End project overview (Data, Processes, Resources, flows, ..)

- **Coding**

- Production ready Code
- Industrial DS pipeline coding
- Model Serving Coding (abstract access, expose prediction capabilities, Horizontally scalable)

- **Devops for ML/AI (MLOps)**

- Integration (build / release)
- Deploy on prod infrastructure
- CI/CD for data science projects

- **Monitoring**

- Production Data monitoring
- Models quality monitoring
- Pipeline monitoring

Data/IA Project Industrialization : Architecture

- **Role of an Architecture**

- Help design and frame the industrialization of the project
- Facilitate understanding and information sharing
- Often a prerequisite of many steps for the project industrialization

- **Functional Architecture**

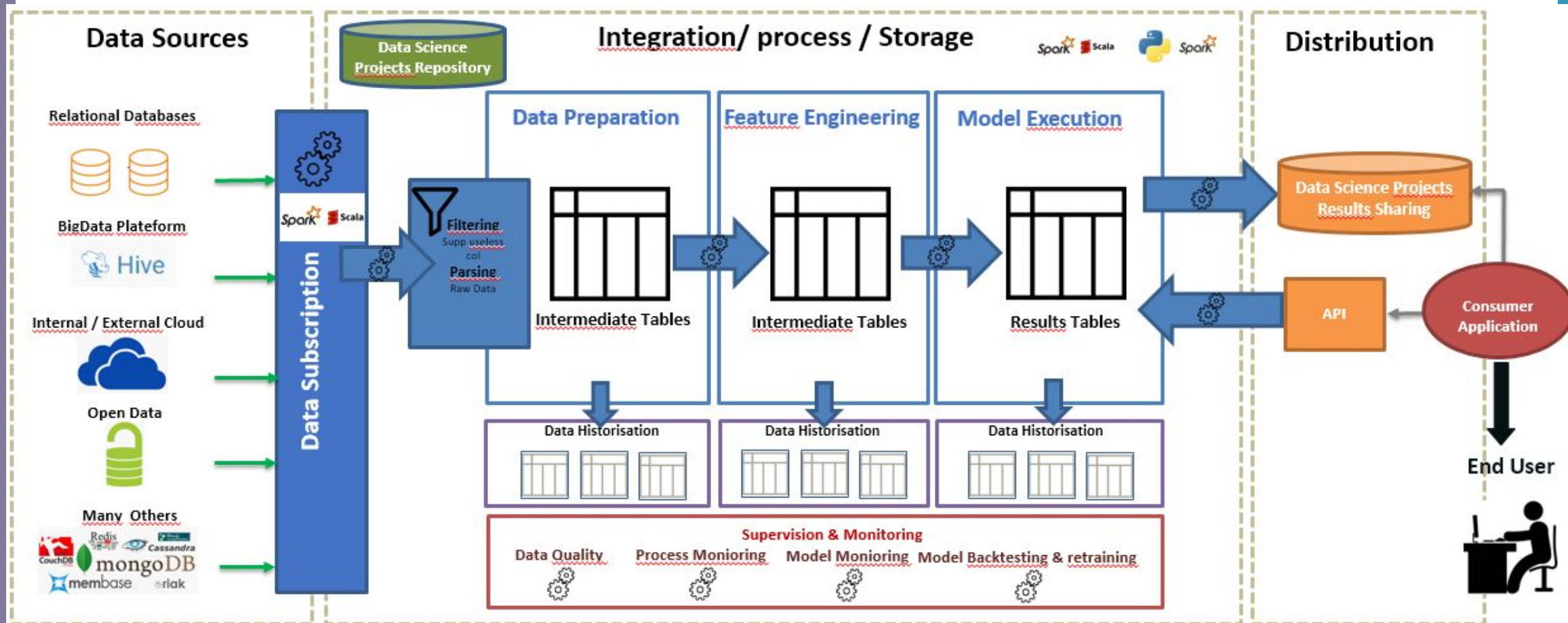
- Describe the business architecture (Business Data, process, results, business output)

- **Technical Architecture**

- Describe Data sources infrastructure
- Processes (scripts, scheduling, ...)
- Storage Strategy for Data & Processes
- Flows

Data/IA Project Industrialization : Architecture

- Example of an Industrialized Data/IA project Architecture



Data/IA Project Industrialization : Coding

- Important coding aspects to define for industrialization pipeline code :
- **Coding Language**
 - Make the right choice in the **early beginning** of Lab developments :
 - Python / PySpark
 - Scala / Spark Scala
 - SAS, R => cost of industrialization ?
- **Coding Structure**
 - Use Modularity/ Micro Services to separate main processes : Data Ingestion, Data Prep, Feature Engineering , Model execution, Model serving, Monitoring
- **Coding Tests**
 - Unit tests (DEV Environment)
 - Integration tests (INT Environment)
 - Validation/Functional Tests (Homologation or Pre-Production Environment)

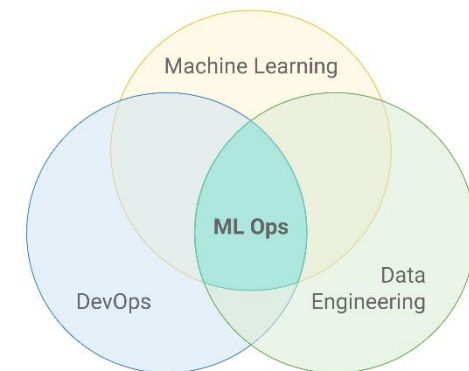
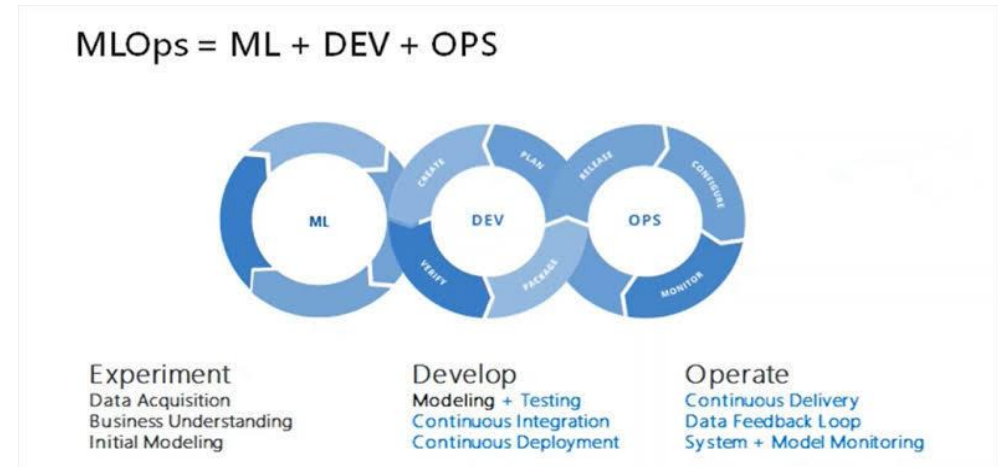
Data/IA Project Industrialization: Devops for ML (MLOps)

• Goals of Devops

- Achieve faster time to market
- Lower failure rate of new releases
- Shorten lead time between fixes
- Improve mean time to recovery

• Devops best practices

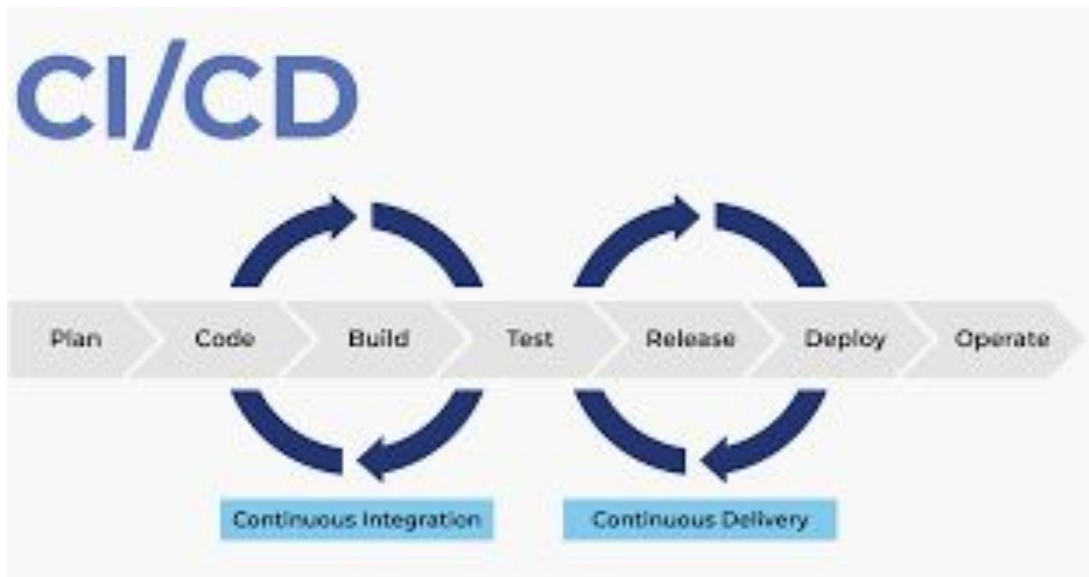
- Continuous Integration / Continuous Delivery (CI/CD)
- Microservices
- Infrastructure as Code
- Monitoring and Logging
- Communication and Collaboration



<https://ml-ops.org/>

Data/IA Project Industrialization: CI/CD

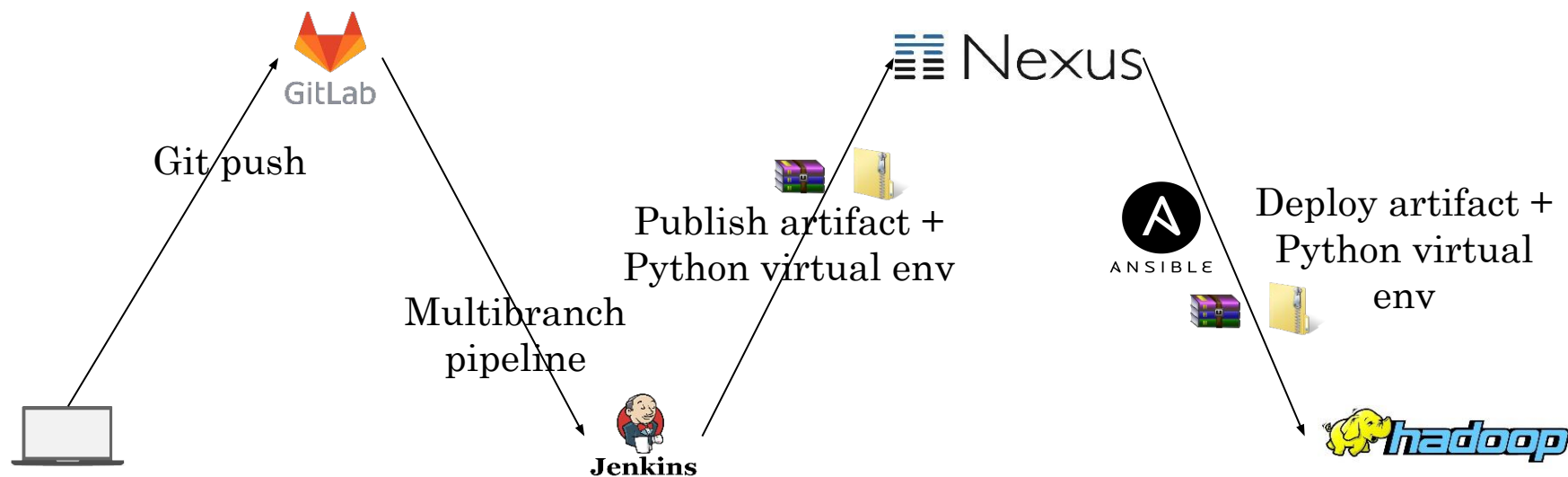
- **What is Continuous Integration / Continuous Deployment (CI/CD) ?**



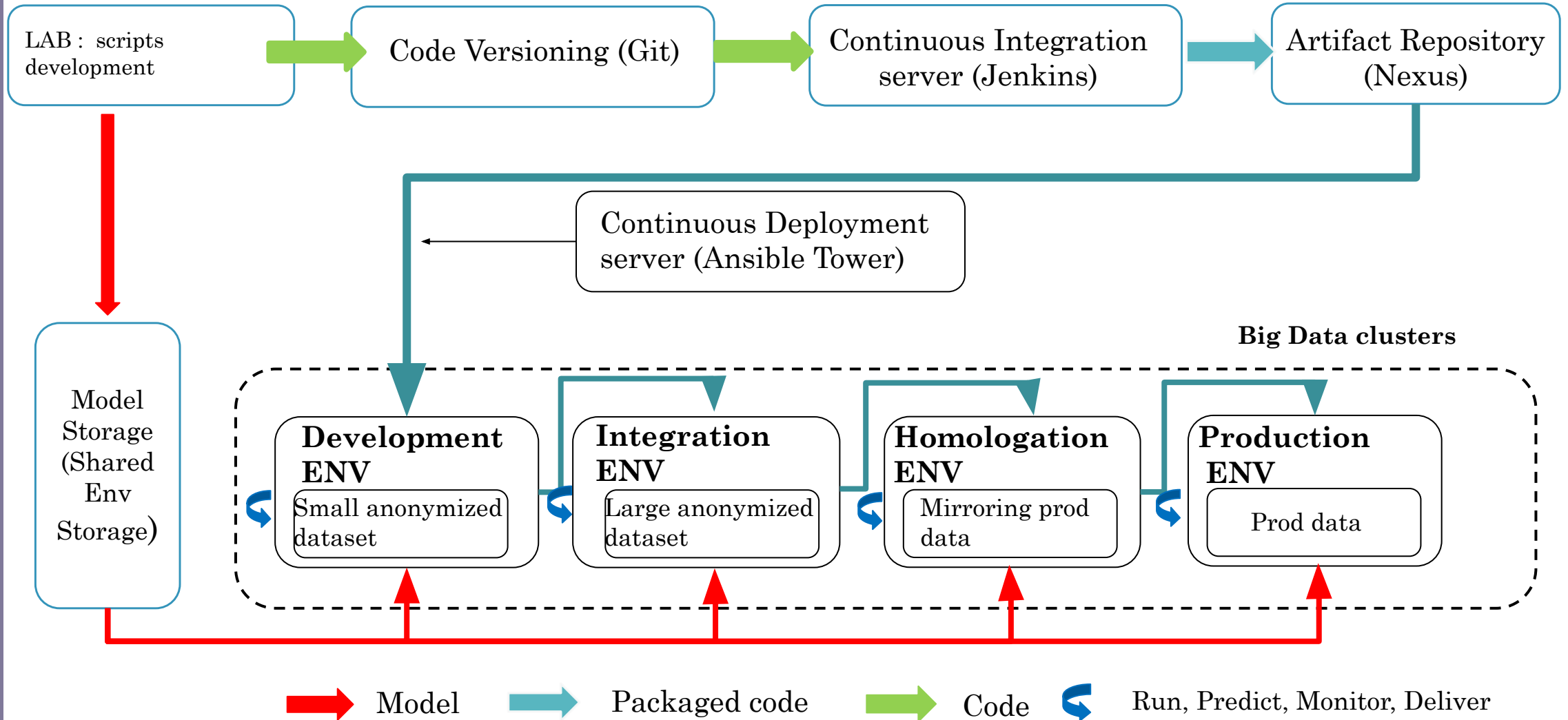
- Systematically ensure the Code Quality
- Have a single artifact for all environments (DEV / INT / HOM / PROD)
- Allow Data/IA team to manage the application from development to deployment
- Automate deployment to avoid human error
- Simplify code libraries deploying by managing it as an artifact
- Be more productive and agile

Data/IA Project Industrialization: CI/CD

- Example of CI/CD python architecture :



Data/IA Project Industrialization: CI/CD



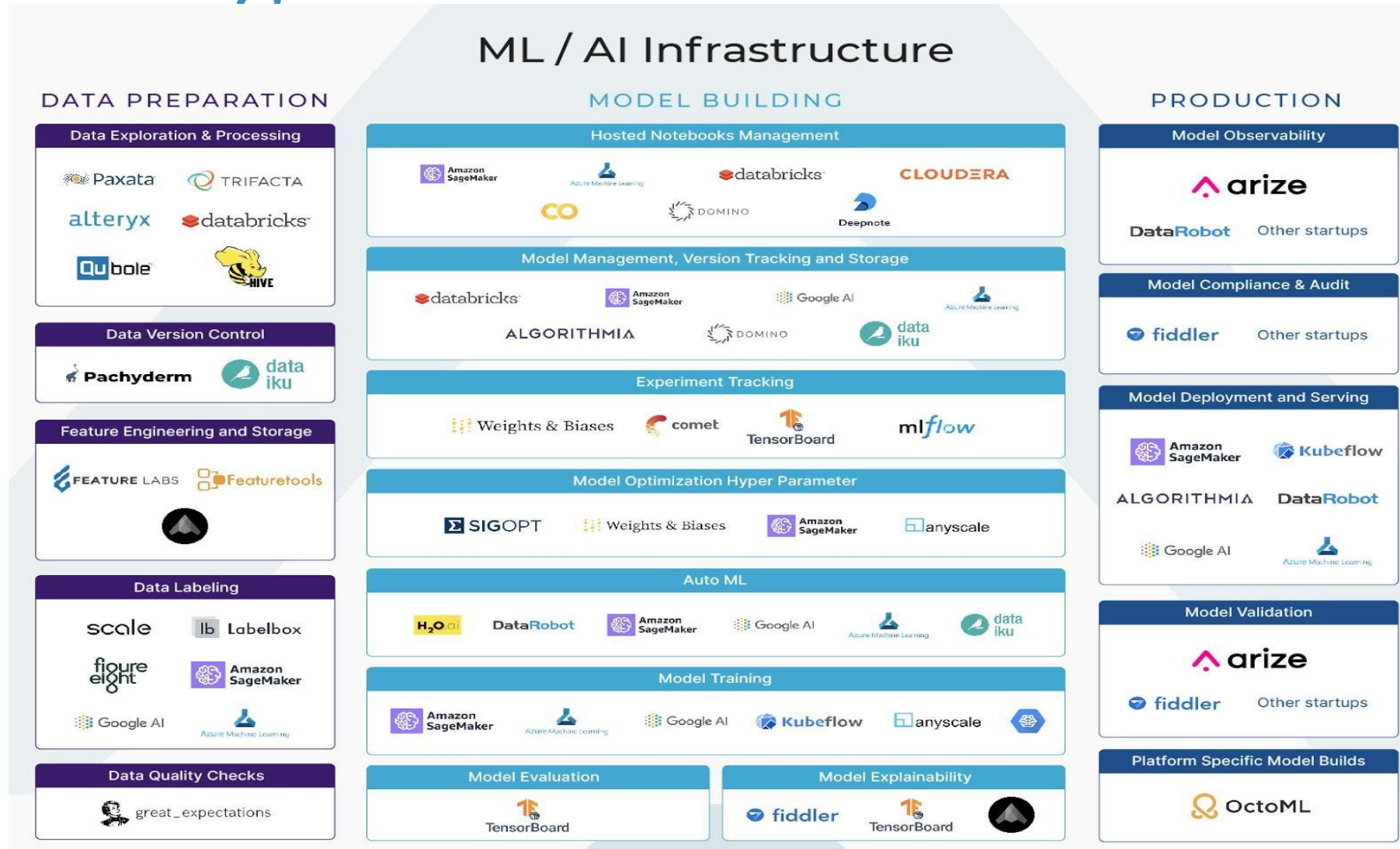
Data/IA Project Industrialization: Monitoring

- Monitor dependency changes throughout the complete pipeline result in notification.
- Monitor data invariants in training and serving inputs: Alert if data does not match the schema, which has been specified in the training step.
- Monitor whether training and serving features compute the same value.
- Monitor the numerical stability of the ML model.
- Monitor computational performance of an ML system. Both dramatic and slow-leak regression in computational performance should be notified.
- Monitor how stale the system in production is.
- Monitor the processes of feature generation as they have impact on the model.
- Monitor degradation of the predictive quality of the ML model on served data. Both dramatic and slow-leak regression in prediction quality should be notified.









Data/IA Project Industrialization: MLops solutions

- Options for models MLops typically fit into a couple of different types:
 - **Internally built executable (PKL File/Java)** : containerized & non-containerized
 - **Cloud ML Provider** : Amazon SageMaker, Azure ML, Google AI
 - **Batch or Stream**
 - **Hosted & On-Prem** like Algorithmia, Spark/Databricks, Paperspace
 - **Open Source framework**— TensorFlow Serving, Kubeflow, Seldon, Anyscale, etc.

Data/IA Project Industrialization : Existing Platforms



Data/IA Project Industrialization : Common open source Toolkits

LOGO	NAME	<i>DATA SCIENTIST USE</i>	<i>IT USE</i>
	Python	Development programming language : data visualisation, data processing, model tuning / training	Deployment programming language : model service
	Gitlab	Git repository manager : version code and ensures collaboration and consistency of the codebase	Starting point of the CI/CD chain : test, build and deploy apps
	Conda	Package manager : ensure consistency between package dependencies	Portability of the code dependencies from an environment to another
	Mlflow	Tracks parameters, artifacts, and metrics during exploration for reproducibility & comparison	Serve model as API or batch and keep track of its use
	Pytest	Validate that code meets specifications	Prevent regression between version. Executed in CI/CD.
	Sphinx	Generate documentation from code	N/A
	Kedro	Common template with lots of utilities to integrate all other tools smoothly	The command line that launches the project
	VS Code	IDE with best integration of other tools (especially git)	N/A

Data/IA Projects : Best Practices

- **Product**

- Business Problem First (models and tools after)
- Integrate & communicate with product and business owners

- **People**

- Build fully skilled project team
- Team co-working and co-localization from the beginning of the DS project

- **Process**

- Adapt PM framework to the organization
- Define KPI (Key Performance Indicator)
- Start simple first

- **Pre-requisites**

- Focus on Data (wrong insights from the data is worse than no insights at all)
- Ensure right tooling is available

- **Privacy**

- Integrate Laws & Ethics in all the life cycle

Data/IA Projects : Key Takeaways

- Consultant mindset
- think Product rather than Project

Data/IA Projects : key takeaways

- **As a Data/IA team member and/or project manager**
 - Be problem Solving First - not tools, technologies, and models.
 - Data will take main of your project time - it will never be clean or easily available. Data gathering and cleaning will take 80% of your time and efforts.
 - Don't underestimate the power of simplicity - Simple tools & models - will be good enough for the majority of the problems. You don't need neural networks to solve every problem.
 - Team working from the early beginning is a key – you can't deliver any DS project alone
 - Learn Data visualization and develop your communications and storytelling skills - people may not appreciate your great work if you can't convince them in simple terms.
 - Writing production ready code is your responsibility and don't underestimate the industrialization stage and manage it as a project.
 - Data Science field is evolving rapidly. Learn continuously.

References

- The Practical Guide to Managing Data Science at Scale, DOMINO DataLab, <https://www.dominodatalab.com/wp-content/uploads/domino-managing-ds.pdf>
- An Introduction to Machine Learning Models in Production, By Jason Slepicka, Publisher: O'Reilly Media
- Building Machine Learning Pipelines, Automating Model Life Cycles With TensorFlow, By Hannes Hapke, Catherine Nelson, Publisher: O'Reilly Media, Release Date: October 2019
- Deploying Machine Learning Models as Microservices Using Docker, By Jason Slepicka, Mikhail Semeniuk, Publisher: O'Reilly Media
- Agile Data Science 2.0, By Russell Journey, Publisher: O'Reilly Media, Release Date: June 2017
- Building Data Science Teams By Paco Nathan, Publisher: O'Reilly Media, Release Date: November 16, 2015, Language: English
- DS PM Guide : <http://www.datascience-pm.com/pm-guide-overview/>