Data Science Project

Data Science Project: An Inductive Learning Approach

Prof. Dr. Filipe A. N. Verri
with contributions from Prof. Dr. Johnny C. Marques

About these slides

These slides are companion material for the book

Data Science Project: An Inductive Learning Approach

Prof. Dr. Filipe A. N. Verri

`https://leanpub.com/dsp`

All intellectual content comes from the book and is not AI-generated. Slides were produced with the assistance of Claude Code.

Licensed under CC BY-NC 4.0. You are free to modify and redistribute this work as long as you give proper credit and do not use it for commercial purposes.

Figured I could throw myself a pity party or go back to school and learn the computers.

— Don Carlton, Monsters University (2013)

Overview

Contents

- What is a project?
- CRISP-DM
- ZM approach
- Agile methodology
- Scrum framework
- Our approach

Objectives

- Explore common methodologies
- Understand agile and Scrum
- Propose a Scrum extension for data science

A data science project is a software project

- Some components are constructed **from data**
- Part of the solution is not designed by a domain expert
- Example: spam filter — learned from labeled emails
- Traditional testing (unit tests) is not sufficient
- Stochastic nature of data requires proper validation

What is a project?

What is a project?

According to the PMBOK[1]:

Definition

A project is **a temporary endeavor** aimed at creating a **unique result**, such as a product or service.

Key characteristics:

- Well-defined **beginning and end** (temporary)
- Clear, **measurable**, and achievable objectives
- Requires strong **collaboration** between teams
- Distinct from **operations** (DevOps, MLOps, DataOps)

[1] Project Management Institute (2025). ***A Guide to the Project Management Body of Knowledge (PMBOK® Guide)***. 8th. Project Management Institute.

CRISP-DM

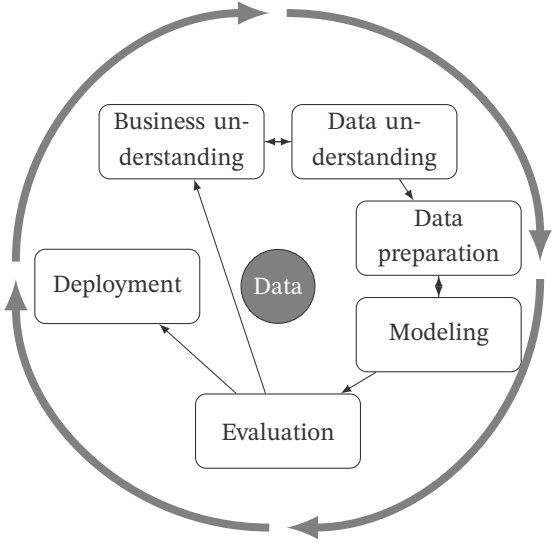
CRISP-DM

Cross Industry Standard Process for Data Mining (IBM, 1990s)

Cyclic process with six phases:

1. Business understanding
2. Data understanding
3. Data preparation
4. Modeling
5. Evaluation
6. Deployment

Diagram of the CRISP-DM process



CRISP-DM limitations

- Completely focused on data
- Does not address software development aspects
- Product = models and findings, not the full software solution
- User interface, communication, integration are not addressed
- Good starting point, but should not be followed strictly

ZM approach

ZM approach — Roles

- **Project sponsor** — main stakeholder, business interests
- **Client** — domain expert, represents end users
- **Data scientist** — sets analytic strategy, connects all roles
- **Data architect** — manages data and data storage
- **Operations** — manages infrastructure, deploys results

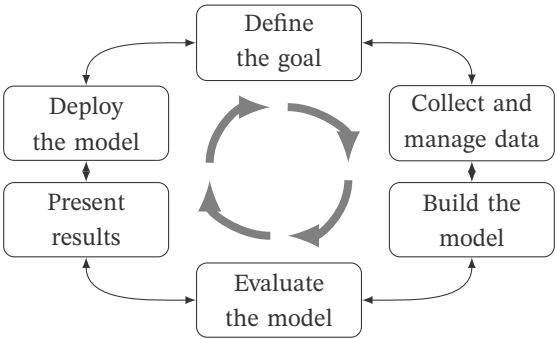
Data science projects are always **collaborative**.

ZM approach — Processes

- Define the goal — what problem are we solving?
- Collect and manage data — what information do we need?
- Build the model — what patterns may solve the problem?
- Evaluate the model — is it good enough?
- Present results and document — how did we solve it?
- Deploy the model — how to use the solution?

Back-and-forth is possible at any stage.

Diagram of the ZM data science process



ZM approach — Limitations

- Suited for consulting or strongly hierarchical organizations
- Maintenance and monitoring delegated to operations
- Does not address software development aspects
- Unclear boundary between project end and operations
- Assumes a pilot/demo — delegates final software to another group
- Modern orgs expect data scientists to build **production-ready** software

Agile methodology

Agile methodology

Alternative to the waterfall (sequential) methodology.

Four values of the Agile Manifesto:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

Items on the right are not discarded, but items on the left are valued more.

Scrum framework

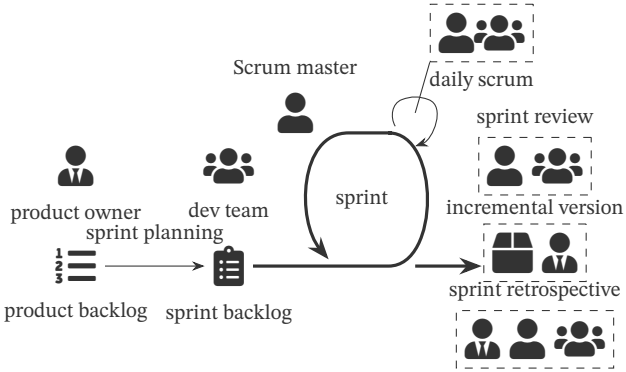
Scrum framework

- Iterative, incremental process
- Three pillars: **transparency**, **inspection**, **adaptation**
- Organized in cycles called **sprints**
- Defined roles, ceremonies, and artifacts

Scrum roles

- **Product owner**
  - Defines product vision, manages product backlog
  - Balances business requirements and technical capabilities
- **Scrum master**
  - Facilitator and coach, not a project manager
  - Removes impediments, fosters self-organization
- **Development team**
  - Cross-functional, self-managing
  - Delivers shippable increments each sprint

Scrum framework overview



Sprints and backlog

- **Sprint** — time-boxed iteration (1–4 weeks)
- **Product backlog** — prioritized list of all desired work
- **Sprint backlog** — subset committed for the sprint
- **Daily scrum** — brief daily progress meeting
- **Burn down chart** — visual progress tracking
- **Sprint review** — demonstrate work, gather feedback
- **Sprint retrospective** — reflect and improve

Scrum for data science?

- Some consider Scrum inadequate for data science
- Argument: Scrum assumes known requirements
- Exploratory phases are not well supported
- Counter: Scrum is a **framework**, designed to be adapted
- Need a compromise between autonomy and a detailed plan

Our approach

Limitations of existing methodologies

- **CRISP-DM** — only data mining stages, no UI or data collection
- **ZM approach** — addresses presentation but delegates software dev
- **Scrum** — good for software dev, lacks exploratory phases
- None fully addresses a data science *software* project

Two additional values

Beyond the Agile Manifesto:

- **Confidence and understanding** of the model over performance
- **Code version control** over interactive environments

Items on the right are not discarded, but items on the left are more important.

Roles

Our approach	Scrum	ZM approach
Business spokesman	Stakeholders	Sponsor and client
Lead data scientist	Product owner	Data scientist
Scrum master	Scrum master	–
Data science team	Dev team	Data architect & operations

Principles (1/2)

- **Modularize the solution** — front-end, back-end, dataset, solution search
- **Version control everything** — code, data, documentation
- **CI/CD** — automated testing and deployment
- **Reports as deliverables** — version controlled, reproducible
- **Setup quantitative goals** — avoid forever improving the model

Principles (2/2)

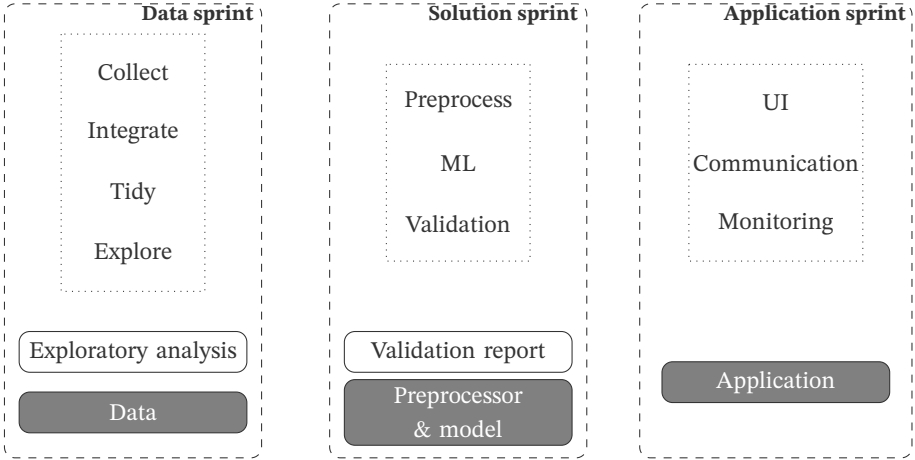
- **Measure exactly what you want** — custom metrics, beware of common pitfalls
- **Report model stability** — understanding > performance
- **Mask DS terminology in UI** — use domain-specific terms
- **Monitor in production** — concept drift, retraining
- **Use appropriate infrastructure** — start simple, scale as needed

Three types of sprints

- **Data sprint** — collection, integration, tidying, exploration
- **Solution sprint** — preprocessing, machine learning, validation
- **Application sprint** — UI, communication, monitoring

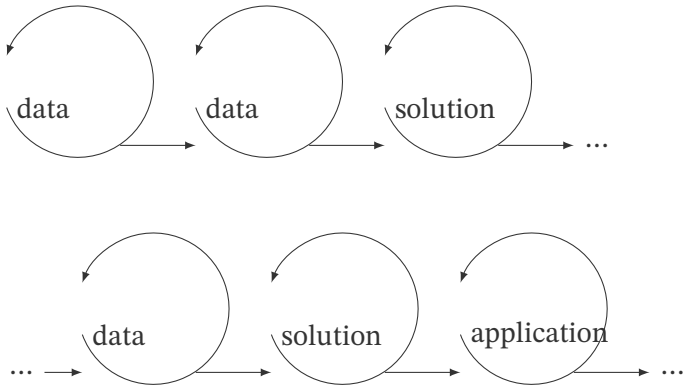
Sprints are sequential; no mixed sprints.
Back-and-forth between sprint types is possible.

Tasks and results for each sprint type



Dark nodes are products fed into the next sprint. **Light nodes** are presented in the sprint review.

Example of sprints of a data science project



Relationship with other methodologies

CRISP-DM	ZM approach	Our approach
Bus. understanding	Define the goal	Product backlog
Data understanding	Collect/manage data	Data sprint
Data preparation	Collect/manage data	Data/solution sprint
Modeling	Build the model	Solution sprint
Evaluation	Evaluate the model	Solution sprint
	Present results	Sprint reviews
Deployment	Deploy the model	Application sprint

Takeaways

- A data science project is a software project
- Modern methodologies should address software development
- Scrum can be adapted for data science with three sprint types: data, solution, and application
- Confidence and version control are key values
- The end result must be a **complete, production-ready** software product

Questions?