# Being Agile in a Data Science Project

Renato Cordeiro[1(✉)], Isaque Alves[1], Samara Alves[2], and Alfredo Goldman[1]

[1] University of São Paulo, São Paulo, SP, Brazil
{renatocf,isaque.alves,gold}@ime.usp.br
[2] Fundação Oswaldo Cruz, Rio de Janeiro, Brazil
samara.alves@fiocruz.br

**Abstract.** Applying agile practices in data science requires adaptations. This paper describes challenges and lessons learned in two applied machine learning projects developed in the XP Lab course at University of São Paulo in Brazil. It compiles six suggestions for educators and practitioners who want to bring agility to their data science initiatives.

**Keywords:** Agile · Data Science · Machine Learning · Software Engineering · Extreme Programming · Scrum · Kanban · XP Lab

## 1 Introduction

Since 2001, the Institute of Mathematics and Statistics of the University of São Paulo has been offering the eXtreme Programming Laboratory (XP Lab) course. The goal of the course is to teach Agile Methods in practice [3]. Students are divided into teams and build a semester-long project for real customers.

During twelve weeks of practical activities, teams are instructed to follow the original XP practices [6]. Most teams also adopt management practices from Scrum and Kanban, learned by many students in the industry.

Given its structure, the XP Lab course provides an environment for testing the use of Agile practices in non-traditional contexts, such as in the development of the Linux kernel [2]. Since 2020, to follow the industry trend, the course organizers seek proposals for data science projects as alternatives to be developed during the course. This paper describes these experiences.

Section 2 and 3 describe the challenges and lessons learned with data science projects in the 2020 and 2021 editions of the XP Lab course. Section 4 highlights suggestions for educators and practitioners based on these experiences. Finally, Sect. 5 summarizes the main contributions from this experience report.

## 2 First Attempt: The Civil Police Project

In 2020, the XP Lab course organizers made their first attempt to bring data science projects to participate in the course. One proposal came from the technicians of the Intelligence Department of the São Paulo Civil Police. The goal


P. Kruchten and P. Gregory (Eds.): XP 2022/2023 Workshops, LNBIP 489, pp. 51–59, 2024.
https://doi.org/10.1007/978-3-031-48550-3_6

was to create a new tool to recognize license plates of vehicles near crime scenes, so the police could track people involved for investigations.

Given a photo captured by a security camera, students should use Machine Learning, specifically Computer Vision techniques, to separate the license plate and recognize its characters (numbers and letters). This task was particularly challenging for two reasons: photos taken by security cameras usually have low resolution and can show cars in different environments, angles, and light.

In total, six students composed the Civil Police project team. The project was successful insofar as the team delivered a demo API and built a training pipeline for a model that could receive a photo with a vehicle and output the characters from the license plate. For that, they relied on open-source libraries such as OpenCV[1] to make image transformations and TensorFlow[2] to train a neural network model to recognize characters.

Unfortunately, there were many challenges throughout the development. First, the team did not get access to images from the Civil Police department. Consequently, they spent lots of time collecting a dataset of photos from the internet that could emulate – albeit imperfectly – what the Civil Police technicians would collect. This affected their ability to create a model to solve the client's actual problem.

Second, both the team did not have practical experience working with data science projects, while the Civil Police technicians did not know about Computer Vision models. As a consequence, the team spent much more time researching techniques and exploring the basics, hindering their ability to improve the model.

The first attempt with a data science project in the XP Lab course taught two important lessons. First, students should have an initial dataset to work with, or otherwise the project will be dedicated to collecting data rather than using it. Second, students should have technical guidance to help them to explore machine learning techniques and apply the data science workflow.

## 3 Second Attempt: The Fiocruz Project

In 2021, the XP Lab course organizers seek once again data science projects. One proposal came from the researchers of the Cellular Communication Laboratory of the Oswaldo Cruz Institute in Rio de Janeiro. The goal of the project was to create a new tool to complement the Fiocruz researchers' ongoing effort to identify emerging technologies in scientific papers.

Given a set of articles, the Fiocruz project team should use Machine Learning, specifically Natural Language Processing techniques, to identify tech-related terms from a set of preselected articles. For that, the new tool has to cluster words from documents. Therefore, the problem requires using unsupervised learning algorithms, such as topic models, to be solved. Since the researchers were familiar with this set of techniques, they could help the team during their development.

[1] https://opencv.org.
[2] https://tensorflow.org.

Based on the previous experience, the XP Lab course organizers guided the Fiocruz researchers to do preparations before the project started. Particularly, the researchers built a web crawler to compile a dataset so the team could start working on the project without concerns about data collection.

In total, 17 out of 48 students were interested in the project. After the selection, six students composed the Fiocruz project team. The team reported that they felt there was a lot of value and purpose in uniting technology with the health area, learning about how they could use data science to help with this research field. They also noted that having no previous experience with Machine Learning was a contributing factor in their choice.

### 3.1 Development Process

The Fiocruz team developed its data science project experimentally and incrementally, following the steps described by CRISP-DM [4] The report below describes the main activities made by the team during each development sprint, up to the end of the course. In total, there were eight one or two-week-long sprints, in which the team worked on average eight hours a week.

Sprint 1 focused on understanding the problem proposed by Fiocruz researchers and the data they provided. This sprint was used as a preparation for the team, so there was no software deliverable for the clients. The main goal was to identify the requirements and analyze what the data could offer. First, the team split into pairs to analyze the data, with multiple people doing the same task. Then, the team did Mob Programming to discuss insights, identify inconsistencies, and report discoveries about the data. In the end, the team prototyped their first data processing functions.

After collecting feedback from Fiocruz researchers, Sprint 2 focused on consolidating the data processing. The team reimplemented their prototype – a data pipeline – into a Python script, creating new functions based on insights gained from constant experimentation with the data. The team then started another research cycle, creating tasks to define the most viable techniques to handle the textual data. Finally, the team took the results to the Fiocruz researchers, so they could assist them in choosing the best tools for the job.

With a data processing pipeline mature enough, Sprints 3 and 4 consisted of exploring and applying the techniques and tools discussed previously. The Fiocruz team improved their text preprocessing using spaCy[3]. After that, they carried out experiments that resulted in implementing the TF-IDF (Term Frequency, Inverse Document Frequency) algorithm, a statistic that reflects how important a word is to a document in a collection of terms.

The team started an exploratory analysis of outliers based on the number of tokens, allowing them to further clean the provided dataset. It resulted in new parameterized functions to remove outliers. In parallel, the team defined activities to study the application of unit tests in the project's context, promoting new discussions within the group.

[3] https://spacy.io.

Sprint 5 had the goal of delivering the data pipeline. The focus was to study and apply feature engineering, dimensionality reduction, and other techniques to improve the results achieved so far. Meanwhile, the team started studying Latent Dirichlet Allocation models [8]. Following the XP Lab course requirements, the team also promoted a refactoring day, which consisted of a Mob Programming session to define the project's architecture and organize the repository.

The remaining sprints focused on applying and improving the LDA-based model, besides studying patterns to design a library that could assist Fiocruz researchers using it. After this research, the team applied the Faade design pattern [7] to create an API to access the implemented functions. Furthermore, as required by the course and in agreement with the researchers, the team created a documentation for the project, including context, architecture diagrams, and details about the architectural decisions. All artifacts can be found in the project's repository, with an OSS license and a guide for contributions.[4]

### 3.2 Adapting Agile Practices

The Fiocruz project team started their development using practices from three agile methodologies: XP, Scrum, and Kanban. The items below summarize the main adaptations made in different practices to better accommodate the particularities of an applied machine learning project:

- **Data Understanding.** Being intimate with the data and knowing what it can offer is essential for creating machine learning models [1]. Therefore, the team focused its first sprint on this task and continuously reviewed its assumptions and knowledge about the data.
- **Spikes to study techniques before using them.** As the team was inexperienced with the necessary tools and techniques for the project, they created spikes to study them and then discuss solutions before coding. Only after debating and verifying the feasibility of applying different models and libraries, they started development.
- **Sprint boundaries.** As many user stories were experimental in nature, most could not be finished in the same sprint. Even after reducing their scope, it was unattainable to fit them within a single sprint. Therefore, the team gave up trying to reduce user stories and focused on collecting feedback about their progress and course correct even if tasks were unfinished. In the end, this worked well, since the results obtained met the expectations of the Fiocruz researchers.
- **Not only working software: useful data and insights.** For a data science project, discovering new tools, gathering information, and finding insights about data was just as important as developing software with quality. Therefore, the team delivered these reports to the Fiocruz researchers.
- **Mob Programming for exploration.** Following XP Lab course recommendations, the team started using Mob Programming for team building.

[4] Documentation (PT-BR): https://gitlab.com/labxp_fiocruz/documentation.

However, they continued applying the technique weekly to share knowledge, discuss solutions, and plan activities.
- **Pair Programming for implementation.** Pair Programming was essential to share knowledge during development. On each sprint, the priority was to form pairs that had never worked together, but also help each other with their coding skills.
- **Notebooks for experimentation, scripts for production.** All experiments began on Jupyter notebooks to validate solutions and present insights to the Fiocruz researchers. After results were deemed satisfactory, the code was reimplemented with functions in Python scripts. This provided the opportunity to apply Test-Driven Development (TDD), since the team could plan their tests while prototyping in the notebooks, and then start the script reimplementation with them.
- **Applying a different test pyramid.** Inspired by the ideas of Continuous Delivery for Machine Learning [5], the team focused on understanding different types of tests related to machine learning, particularly regarding how to test data and training pipelines.

### 3.3 Challenges

Throughout the project, the Fiocruz project team had to deal with many challenges related to the machine learning product development, such as:

- **Reference Architecture.** The team did not have a reference architecture to solve the problem proposed by the Fiocruz researchers. While there are well-documented architectural patterns in more traditional domains such as web development, the team had difficulties finding a proven way to implement their solution. The team architected a library using Object-Oriented software patterns [7], considering the project context and the single responsibility principle.
- **Team Insecurities.** Given the problem proposed by the Fiocruz researchers, the team was always unsure whether results were adequate. This is a characteristic of using unsupervised learning, since there was no objective way to assert the quality of proposed models. Nevertheless, the constant interaction with the researchers helped to validate the results.
- **Sprint scope.** During the first three sprints, the team tried to increase the granularity of user stories and tasks to finish them within a single sprint. However, as the tasks were experimental by nature, it was hard to predict the necessary work time. In the end, the team chose to prioritize quality. Sprints were used to maintain continuous feedback with the researchers to ensure satisfaction and reduce the risk of not delivering what was expected.

Due to the COVID-19 pandemic context, the XP Lab course was held remotely. Although this might seem a challenge, students explored how to build interpersonal relationships through team-building dynamics and slack time.

### 3.4 Results

At the beginning of the project, the Fiocruz team mapped tools and practices they expected to use during the development. Then, the team created a table compiling their self-assessed familiarity with those items. Figure 1a shows their knowledge at the beginning of the project. After the initial evaluation, the team defined pairings and conducted workshops to share knowledge. Figure 1b shows their knowledge by the end of the course. Fortunately, there was a significant improvement, indicating that the team learned with the experience.

| | Knowledge Board (Beginning) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | XP | Scrum | Kanban | Docker | Git | Python | Pandas | NLTK | SpaCy | Notion | Unit tests | E2E tests |
| Student #1 | 😊 | 😁 | 😁 | 😊 | 😁 | 😐 | 😐 | 😭 | 😭 | 🤔 | 😐 | 😐 |
| Student #2 | 😭 | 😐 | 😭 | 😭 | 😭 | 😐 | 😐 | 🤔 | 😐 | 😭 | 🤔 | 🤔 |
| Student #3 | 😭 | 😐 | 😁 | 😐 | 😊 | 😊 | 😁 | 🤔 | 😐 | 😭 | 😭 | 😭 |
| Student #4 | 🤔 | 🤔 | 😐 | 🤔 | 😭 | 😭 | 🤔 | 🤔 | 🤔 | 😁 | 🤔 | 🤔 |
| Student #5 | 🤔 | 🤔 | 🤔 | 😭 | 😭 | 😊 | 😐 | 🤔 | 😐 | 🤔 | 🤔 | 🤔 |
| Student #6 | 😐 | 😊 | 😁 | 😭 | 😊 | 😐 | 😐 | 🤔 | 🤔 | 🤔 | 🤔 | 🤔 |

(a) Self-assessed knowledge collected at the beginning of the project.

| | Knowledge Board (End) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | XP | Scrum | Kanban | Docker | Git | Python | Pandas | NLTK | SpaCy | Notion | Unit tests | E2E tests |
| Student #1 | 😁 | 😁 | 😁 | 😁 | 😁 | 😊 | 😁 | 😁 | 😁 | 😁 | 😁 | 😁 |
| Student #2 | 😁 | 😁 | 😁 | 😭 | 😐 | 😊 | 😊 | 😭 | 😐 | 😁 | 😁 | 😐 |
| Student #3 | 😁 | 😁 | 😁 | 😭 | 😁 | 😁 | 😁 | 😐 | 😊 | 😁 | 😊 | 😐 |
| Student #4 | 😁 | 😁 | 😁 | 😭 | 😐 | 😊 | 😐 | 😐 | 😐 | 😁 | 😊 | 😭 |
| Student #5 | 😁 | 😁 | 😁 | 😁 | 😁 | 😁 | 😁 | 😁 | 😁 | 😁 | 😁 | 😁 |
| Student #6 | 😁 | 😁 | 😁 | 😭 | 😊 | 😊 | 😐 | 😐 | 😐 | 😊 | 😭 | 😭 |

(b) Self-assessed knowledge collected at the end of the course.

**Fig. 1.** Knowledge boards comparing the Fiocruz team knowledge in different methodologies, technologies, and concepts.

The weekly meetings between the team and the Fiocruz researchers allowed continuous feedback and review of results. During these meetings, the researchers focused on guiding the team's actions towards the project goals, while giving them the freedom to experiment with different techniques and do their research. On the other hand, the team always prepared for the meetings by bringing rich insights and making technical questions about machine learning tools.

In the end, the project was delivered with a complete product that included a Python open-source library that can be integrated in the Fiocruz researchers' routine, with documentation that will allow the project's continuation. All code can be found in the project's repository, with an OSS license and a guide for future contributions[5].

## 4 Suggestions for Data Science Projects

Based on the experiences described in Sects. 2 and 3, here follows a set of suggestions for educators attempting to bring data science to their agile courses (or

[5] Code (MIT License): https://gitlab.com/labxp_fiocruz/experimentation.

agility to their data science courses). These tips may also be useful for practitioners who wish to improve the agility of their own data science projects.

- **Understand the data and what it can offer.** As recommended by CRISP-DM [4], the first step in a data science project should focus on understanding the business requirements and the available data. Having a well-scoped problem and real data was paramount for the success of the Fiocruz project in comparison with the Civil Police project.
- **Use notebooks for experimentation, scripts for production.** Jupyter notebooks are a great tool for experimentation, since they promote rapid iteration during development. However, they are not ideal for production code since they complicate applying good practices such as code versioning and testing. After using them to gain insights and collect client's feedback, code should be reimplemented in scripts using proper traditional software engineering techniques.
- **Make tests, lots of them.** Self-tested code enables refactoring and debugging. Test-driven development further improves code quality by encouraging thinking about functionality first. Data science code may go untested because the development environment does not facilitate it (see the previous item) and because it relies on external libraries. However, automated testing is a proven software engineering technique that can and should be applied in as much data science code as possible. There is emerging literature such as CD4ML [5] that provide guidance for testing different parts of applied machine learning software.
- **Use mobbing for brainstorming, use pair for coding.** Mob and Pair Programming promote joining multiple developers in a single computer to develop. Mob Programming proved itself very useful for discussing solutions and techniques, given the whole team could share their ideas. On the other hand, Pair Programming showed itself more efficient in executing coding tasks, since it allows teams to further parallelize their work.
- **Focus on quality, not deadlines.** Dividing work into sprints (as described by Scrum) did not benefit the predictability of delivery. Many tasks, experimental in nature, leaked beyond the expected sprint boundaries. Rather than viewing sprints as deadlines for the tasks, it is better to focus on the quality of results and use sprint reviews as an opportunity for continuous feedback with clients.
- **Iterate with stakeholders to collect feedback.** Customer collaboration is one of the four values of the Agile Manifesto. This interaction is even more important for data science projects, given their dependency on data. Sharing insights with clients and further understanding business requirements and data particularities allows creating better models to solve the problem proposed.

## 5 Conclusion

This paper showed two experiences with data science projects in the XP Lab course offered by the Institute of Mathematics and Statistics at University of

São Paulo. It summarized the challenges and lessons learned from adapting Agile practices – particularly from XP and Scrum – for data science. These adaptations were summarized in a set of suggestions to help educators and practitioners to be agile in their data science initiatives.

There are some factors that may make it difficult to reproduce the experiences described in this research, notably the positive results from the Fiocruz project described in Sect. 3. First, the Fiocruz researchers prepared a dataset for the team. Second, the researchers were technical clients that could support the team with tools and techniques. This might not be possible for all projects, as shown by the Civil Police project described in Sect. 2.

Given the successful results and the popularity of data science, the XP Lab course organizers hope to bring other data science projects to the course, and continue compiling good practices for succeeding in them.

**Acknowledgments.** We would like to thank the members of the Civil Police and the Fiocruz project teams, as well as thank the Intelligence Department of São Paulo Civil Police and the Cellular Communication Laboratory of the Oswaldo Cruz Institute. This research would not be possible without them.

## References

1. Isaque, A., Leonardo, L., Paulo, M., Carla, R.: Product engineering for machine learning: a gray literature review. In: 2021 IEEE/ACM 43rd International Conference on Software Engineering: WAIN 2021 - 1st Workshop on AI Engineering - Software Engineering for AI
2. de Oliveira Rosa, T., Goldman, A.: Is it possible to apply agile methods to contribute to the Linux kernel?. Agile Processes Softw. Eng. Extreme Program. Workshops **396**, 291–297 (2020)
3. Goldman, A., Almeida Santos, V.: Continuous Improvement of an XP Laboratory Course: An 18 year History, Experience Report, In: Agile (2019)
4. Wirth, R., Hipp, J.: CRISP-DM: towards a standard process model for data mining. In: Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining (2000)
5. Sato, D., Wilder, A., Windheuser, C.: Continuous Delivery for Machine Learning (2019). https://martinfowler.com/articles/cd4ml.html
6. Beck, K., Andres, C.: Extreme Programming Explained: Embrace Change (2nd Edition). Addison-Wesley Professional (2004)
7. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Longman Publishing Co., Inc, (1995)
8. Blei, D.M., Andrew, Y.N.G., Jordan, M.I.: Latent Dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)