Socio-Technical Challenges and Recommendations for Mitigation when Building ML-Enabled Systems

Alina Mailach 1 and Norbert Siegmund 1

Abstract: Despite numerous success stories about ML-enabled software systems, there are consistent reports from industry struggling to bring ML models into production. To understand these challenges, we examined 66 hours of video content recorded by the MLOps community, offering insights into socio-technical anti-patterns observed within and across teams, along with practitioners’ recommendations to target their root causes.

Keywords: Socio-Technical Anti-Patterns, Machine Learning (ML), Software Systems, Production Challenges

1 Introduction

The original paper “Socio-Technical Anti-Patterns in Building ML-enabled Software: Insights from Leaders on the Forefront” has been published at the International Conference on Software Engineering 2023 [MS23]. In just a few years, machine learning (ML)-enabled software systems became ubiquitous in our daily lives. The huge number of existing applications empowered with cognitive and visual AI capabilities and the emergence of entirely new business domains based on ML draw an indisputable success story. Yet, we see only the fraction of products that made it from development to production. While there is a large body of work targeting technical aspects of developing ML-enabled software, only few papers have targeted the socio-technical aspects. This is surprising because anecdotal evidence from blog posts, tweets, and (video) podcasts, hint that not solely tools and technologies are main causes for failed projects, but management, organization, and social aspects.

In our study, we set out to explore what experts and leaders in industry report and discuss in community meetups on socio-technical challenges when building ML-enabled software. Specifically, we state our research question around the organization and management of teams and persons: What socio-technical challenges and what kind of solutions do practitioners discuss in regard to productionizing machine learning? We performed a large-scale qualitative study using reflexive thematic analysis to identify socio-technical anti-patterns, their causes, and mitigation strategies from practitioners’ discussions. To this end, we analyzed 73 Meetup and Coffee Session recordings from the MLOps community, a group of over 11,000 professionals focused on ML production experiences.
2 Results

We group our findings into three main areas: organizational silos, communication within an organization, and organizational leadership vacuum. The areas are further structured into 7 contexts or development activities in which, in total, 17 anti-patterns can occur.

**Organizational Silos** often manifest when multiple teams work on disjoint tasks, challenging data and model transfer. The *model to production integration* is notably tedious and error-prone, involving transitioning a developed model to production. To address the gap between data scientists and software engineers, practitioners suggest using tools, such as model registries and feature stores. Additionally, forming cross-functional teams, translation between the professions, and pairing them are advised to ease these challenges. Another challenging intersection occurs at the *data producer to consumer integration*, which involves handing over data between teams with different use cases. Practitioners note tight technical coupling, data access, and availability issues. These might be mitigated by increasing data producers’ awareness of downstream dependencies or using a centralized platform as an inter-team contract.

Missing **Communication** has been reported to be a major problem for causing a *redundant development* of features, tools, and infrastructure in organizations. As teams lack natural intersections, organizational knowledge and tools are hard to access. Practitioners recommend centralization of tools and features to efficiently allow for reusability. Furthermore, showcase meetings can help teams get closer together and share what they are working on to find natural intersections with other teams. A lack of communication further leads to *tension between management and data science*, which might be mitigated by strong processes, and documentation and reporting, although this, at least partially, diverges from the low documentation approach of agile practices.

The third area’s issues share a **Leadership Vacuum**, where a lack of knowledgeable authority leads to *headless chicken hiring* – recruiting staff with inadequate or incorrect skills, who end up without a defined product to work on. As unclear roles and titles are reinforcing the skill shortage of the market, it is recommended to hire for skills and potential rather than roles. Tools and models misaligned with team or product goals often result from *Résumé-Driven Development*. This can be mitigated by better utilizing existing organizational knowledge. Finally, successful proof-of-concepts that never make it to production, a perception of ‘everything can benefit from ML’, and infeasible products might be signs of *Hype-Driven Development*, which practitioners recommend to counter with education for management, clear product discovery processes, and a close feedback loop with customers.

**Bibliography**