Teaching Machine Learning as Part of Agile Software Engineering

Steve Chenoweth, *Member, IEEE*, and Panagiotis K. Linos, *Member, IEEE*

***Abstract*—*Contribution:* A novel undergraduate course design at the intersection of software engineering (SE) and machine learning (ML) based on industry-reported challenges.**

***Background:* ML professionals report that building ML systems is different enough that we need new knowledge about how to infuse ML into software production. For instance, various experts need to be deeply involved with these SE projects, such as business analysts, data scientists, and statisticians.**

***Intended outcomes:* The creation of a table detailing and matching industry challenges with course learning objectives, course topics, and related activities.**

***Application design:* Course content was derived from interviewing industry professionals with related experience as well as surveying undergraduate SE students. The proposed course style is designed to emulate real-world ML-based SE.**

***Findings:* Industry-derived content for a pilot undergraduate course has been successfully crafted at the intersection of SE and ML.**

***Index Terms*—Agile software development, computer science, computer science education, course design, design process, instruction, machine learning (ML), software engineering (SE).**

I. Introduction

THE AGILE software development movement [1] teaches that the best approach depends on the task, people, and technologies involved. In Scrum [2], a retrospective after each Sprint allows developers to adjust their approach based on new learnings. Agile teams can rapidly develop features if they have a suitable platform and knowledge of the project's domain and customer. However, unfamiliarity with tools or domains causes risks to remain and hinder feature generation. Lack of expertise in machine learning (ML) poses these threats when ML-based features are included. Integrating ML into agile processes also requires addressing differences in development cycles and collaboration with data scientists.

Moreover, developing artificial intelligence (AI)-based systems, including ML, poses unique challenges due to their nebulous nature and the need for integration with existing systems. ML-based systems often deviate from the assumptions of agile development, such as having client expertise, thus requiring a solid understanding of underlying forces and expectations. The uncertainty in ML projects and reliance on ever-changing data creates inherent gambles and can hinder the use of agile methods [3].

Consequently, important areas of inquiry arise in undergraduate education, regarding effectively teaching students to utilize ML in agile software development projects. One question is whether an undergraduate course can incorporate current ML best practices, such as ML as a Service (MLaaS) or autoML [4], and whether this more specific learning would adequately prepare students for the workforce. The effectiveness of a more canned approach and the challenges it presents are key aspects to consider in further curriculum development.

For this research, it was hypothesized that training students in ML as part of agile generally can mitigate these challenges. More specifically, this article explores the issues and presents directions for an undergraduate course at the intersection of ML and agile software engineering (SE) that is supported by an empirical study and industry feedback. The authors aimed to adequately prepare students for the workforce to provide a model for other institutions.

This article is organized into sections discussing the literature review, the synergy between ML and agile SE, the empirical study, proposed course content, and conclusion with next steps.

This research is granted formal IRB approval for contacts with participants, and they signed the appropriate consent agreements.

II. Literature Review

Industry reports various challenges when infusing ML within an agile SE environment. For instance, Rahman et al. found that "poor quality of requirements" led to many issues—in general, it is easy for a client to describe what they want from an ML project in sweeping generality, but what that means more precisely has a huge impact on the success of the project. Model training and maintenance had lumps. Using the ML model in a production system required complex interfaces [5].

Schleier-Smith [6] attempted to use an agile development style to create real-time ML applications using a standard architecture. Their architecture had a goal of allowing rapid experiments with updating in real-time. Among the issues the author encountered were "Long development cycles, Limited feature transformations, Difficulty in generating training data," and tight coupling between the ML functions and the rest of the online system.

Manuscript received 30 May 2023; revised 6 November 2023; accepted 17 November 2023. Date of publication 21 December 2023; date of current version 4 June 2024. *(Corresponding author: Steve Chenoweth.)*

Steve Chenoweth is with the Computer Science and Software Engineering Department, Rose-Hulman Institute of Technology, Terre Haute, IN 47804 USA (e-mail: chenowet@rose-hulman.edu).

Panagiotis K. Linos is with the Computer Science and Software Engineering Department, Butler University, Indianapolis, IN 46208 USA.

Digital Object Identifier 10.1109/TE.2023.3337343

In 2012, Langford [7] noted that building a system which includes a running ML function, as opposed to just using its model, likely results in performance problems because these algorithms tend not to run in linear time. Langford proposed that agile teams would be good for tackling the novelty of ML projects. However, he pointed out the "balancing acts" they must do, for example, acting in a self-organizing way in areas where they have expertise, but not in other areas. The self-organizing goal recommends that data scientists be an integral part of the development team, something that could involve a culture clash.

A close analogy to developing agile systems incorporating ML is the creation of product lines using agile. The ways of doing this effectively are described especially well in Chapter 25 of the Third Edition of Bass et al.'s [8] *Software Architecture in Practice*. The similarity of these endeavors lies in the separation of the core activity from the final product development. For product lines, a common central core is separated out from each of the included products, so that this part need be done only once. However, this also creates issues, including having all the product development groups needing to wait for new versions of the common core. The pace of development, of a final product and the core, is clearly different because the core work relies on dependencies to all the products in the product line.

What is common for ML, with developing an agile product and an ML "core," is that major components have their own separate issues and tend to get out of synch. Like with product lines, there may also be entirely different people working on the agile final product and the ML core.

In the creation of product lines, close attention is paid to "variation points," places where configurations and features of final products will differ, which are all based on the common core. An equivalent for agile ML systems would be points of interface between the ML model and the rest of the system where one or the other is expected to change. For example, how do you create the product so that the type of ML model can be changed out from under it? Or even the underlying algorithm? Moreover, the pace of change of the AI core of a larger system may resemble the pace for change in the core of a product line—often with slower, longer release cycles than for the products themselves.

Amershi et al.'s [9] investigation of multiple ML projects at Microsoft revealed three key distinctions from traditional SE.

1) ML revolves around data, which introduces greater complexity in terms of discovery, sourcing, management, and versioning compared to software code.
2) Developing models for customizability and extensibility requires not only SE skills but also a deep understanding of ML to construct, evaluate, and fine-tune models from scratch.
3) Finally, maintaining strict module boundaries can be more challenging in ML components compared to SE modules.

ML models can become intertwined in complex ways during training and tuning, even if the intention was to keep them isolated.

In their 2019 study, Baier et al. [10] did a structured literature review and an interview study of professionals. They identified the key ML needs for "automated strategies for data drift detection and handling, standardization of ML infrastructure, and appropriate communication and expectation management."

In 2020, Paleyes et al. [11] summarized surveys of the deployment of ML systems. They describe the many skills required for successful ML projects. The authors used their list in analyzing the data for the study. Paleyes also recommended the survey method that was used, for gathering data about ML deployment.

Serban et al. [12] described a trend toward adopting ML-specific SE techniques for developing these systems. An example is "Automated Model Deployment." Another is "Enforce Fairness and Privacy." They found some important practices still had a low-adoption rate, such as feature management and hyperparameter optimization.

Heseenius et al. proposed a development model for systems, including data-driven applications such as ML. They identified roles more specific than those often assigned in software projects, including data scientist, data domain expert, and domain expert in addition to software engineer. In their structured engineering process, people in these roles played different parts during phases of a data-driven project [13].

These authors offered an "Engineering Data-Driven Application" process which shows separate tracks for the data engineering from the main SE of a project. The data engineering includes feasibility of ML for a project exploring the data, establishing model requirements, model development, model integration with the main part of the system, and operational mode. Additional points where the SE and data engineering come together are at the specification and design phase, and at the ending with operational use of the system. The authors believed this process is adequate for many kinds of systems, and they have had some experience using it. They mentioned agile software development as being amenable to this process, but they did not show a model specifically describing how agile would fit. They did allude to an agile process, but it is one of discovery of data features and capabilities, not one of feature creation.

In addition, [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26] support the thesis that development processes for creating successful ML systems are significantly different from processes for normal business systems. ML tools used in industry have become more identified with process-related steps for ML model creation.

Thus, the need for education and best practices is clear, to support the rapidly growing type of ML engineering which ties ML to other software. In the field of teaching AI generally, as with ML, the focus usually is on teaching what algorithms do. Much published research is about using AI to teach AI, e.g., "Teaching Introductory AI with Pac-Man" [27]. Or "A Robot Laboratory for Teaching AI" [28]. Or teaching AI using "agents" as a theme [29], a methodology pioneered by Russell and Norvig [30]. Kastner and Kang's 2020 paper [31] is a step forward in trying to integrate AI generally with production systems, using SE principles.

Existing undergraduate and professional courses in ML often focus on building pure ML models and algorithms and lack integration with traditional SE work. While some courses touch on topics like continuous integration, testing, and AI integration, there is a need for more comprehensive curriculum development based on industry feedback. Due to time constraints and the need to cover basics, undergraduate courses typically also lack advanced ML material. However, universities like the University of Southampton [32] and the University of Alberta [33] include topics, such as applying ML methods, analyzing algorithm limitations, and considering ethical and societal contexts. Advanced courses at institutions like MIT [34], UC Berkeley [35], and CMU [36] delve into topics like analyzing tradeoffs, designing production systems with AI components, and SE methodologies.

It is worth mentioning that one of the coauthors of the current article has created a course on SE for ML, focusing on transforming ML models into production software systems and addressing system decomposition and feedback incorporation. It followed guidelines from Kästner and Kang [31] and thus is like CMU's course [36]. It used Hulten's book [37].

In the above-mentioned courses, the authors believe the methods of extracting issues in industry and applying them to their course were not made explicit, though in some cases, they did list colleagues and professionals who reviewed the course content. In the authors' opinion, such courses still can serve as a foundation for crafting a new undergraduate course at the intersection of ML and SE, which would benefit from professionals' insights and lessons learned.

Finally, the integration of ML and SE into cloud-based systems is a relatively unexplored area with limited published research. Existing materials from vendors primarily focus on building "pure ML" systems for prediction rather than complex systems that require coordination between ML development and other components. Google's "Best Practices for Implementing ML on Google Cloud" offers guidance on preprocessing datasets and incorporating ML into cloud-based systems [38]. Advanced vendor courses are centered around building standalone ML systems using specific tools provided by, say, Google or Microsoft, rather than combining ML with other tools for larger system development [39], [40], [41].

III. Synergies Between Machine Learning and Agile Software Engineering

While crafting the new course, we needed to consider possible ways of combining ML with agile SE with the end goal being the production of an intelligent system (IS). Table I below summarizes selected combinations (or synergies) along with application examples.

Any additional synergies which are not covered here are likely to be explored soon. The remainder of this section describes the synergies included in Table I and some related key challenges for each synergy.

TABLE I
Synergies Between ML and Agile SE

	Type of Synergy	**Application Examples**
1	Use an agile methodology (e.g., Scrum) to manage the process of developing ML models and algorithms including data sampling and curation, model training, model validation, testing and deployment.	Exploration of a new market opportunity using a known ML algorithm.
2	Use Agile (e.g., Scrum) to develop an Intelligent System (IS) while at the same time researching appropriate AI-based ML models for that system.	IT help desk IS that learns from feedback of successful instructions.
3	Use ML as a cloud-based Service (MLaaS) during a Scrum project to build data science based ISs.	A banking web application automating a loan application and approval process.

A. Synergy 1 Challenges

Agile processes like Scrum are employed to build ML models and related algorithms. However, the nature of many ML systems, which involve data exploration and experimentation, poses challenges in fitting them into the agile model. These developments, typically led by data scientists, involve multiple iterations with different algorithms, result visualizations, and eventual reporting to the client. Unlike traditional agile development, where incremental deliverables are expected, these ML developments often deliver even an incomplete solution only at the end. Existing academic courses addressing the development of such systems are further discussed later in this article.

B. Synergy 2 Challenges

Here, Scrum is utilized to develop an IS while simultaneously conducting research on suitable ML models for that system. There are similar problems to Synergy 1, regarding making the ML part of the development fit the agile process. Next are detailed expected concerns for this Synergy.

1) Education: The authors believe there are no undergraduate courses addressing use of agile methodologies with a system that includes ML as well as other common system features.

2) Pace of Delivery: The agile process is designed to do its rapid cycles, in each one gathering information about the next feature, then developing and testing that feature, then showing it to the customer perhaps 2 to 3 weeks later. This works well when the platform supporting all those features already is decided and significant architectural changes are not needed. In ML systems this is not the case if the ML is being researched at the same time. The AI part of the platform may change significantly at any time, and the level of risk may not be reducible for individual features separately.

3) Development Tools: For existing systems to which ML is being added, there are different tools and platforms for the AI part of the system. Typically, a model is created by training on a dataset. That model is then used with "test data" in the production system. But there are inherently two separate efforts—the main system and the AI part being added, and likely two separate groups with different expertise working on these.

4) Feedback and Retraining: Consider interactive ML systems, where the user activity generates more data feeding back into the pool of data for learning. For example, when the goal is to optimize the operation of a help desk, and each instruction to the operators for a given step then gives feedback to the AI data about the success of that instruction. In these systems there is a somewhat unstable 2-way dependency between the AI system generating models and the production system. Changes to either side can alter the interface between the two parts of the system. For instance, we discover that we need to track the timing of each step in a help operation, as well as rate its success.

5) Client Knowledge: Success of an agile project depends on getting correct directions from a client who works closely with the team. This direction gathers an additional layer of complexity when the client is a customer or product manager who is not an expert on data science, and the product software team is getting additional direction from a person who is expert on that, who is discovering simultaneously what is possible to do with the data and the kinds of algorithms likely to be successful.

C. Synergy 3 Challenges

In this scenario, MLaaS is employed within a Scrum project to develop an IS. Here, vendors are trying to supply education and tools to make development easier. But there are additional issues, beyond those listed for Synergy 2. The following list of challenges is composed from the feedback received while interviewing ML professionals (see next section entitled Empirical Study). It also considers selective suggestions from the AI Chatbot known as ChatGPT [42].

1) Data Availability and Quality: MLaaS depends on data for ML training and inference. Ensuring the availability, quality, currency, and diversity of data is challenging in an agile development environment where pace of development has priority. Data collection, labeling, and preprocessing requires significant time and effort, at odds with the iterative and fast-paced nature of agile development.

2) Uncertain Requirements and Scope: MLaaS systems involve complex algorithms and models that require multiple iterations and experimentation to achieve satisfactory results. This uncertainty in requirements and scope makes it difficult to plan and estimate development efforts accurately within the fixed timeframes of agile sprints.

3) Model Training and Iteration Time: Training ML models is computationally intensive and so time-consuming. Agile development emphasizes short iterations and frequent releases, which do not align well with the time required for training and tuning ML models. It can be challenging to strike a balance between delivering working software incrementally and allowing time for model development and validation.

4) Collaboration/Cross-Functional Expertise: Successful MLaaS development requires collaboration between data scientists, software engineers, domain experts, and stakeholders. Agile engineering methodologies emphasize cross-functional teams, close collaboration, and continuous communication. However, achieving effective collaboration across diverse skill sets and integrating ML-specific knowledge into agile practices is a challenge.

5) Continuous Integration and Deployment: Continuous integration and deployment (CI/CD) practices are important for agile development, allowing for rapid feedback and frequent releases. However, integrating ML models into CI/CD pipelines is complex due to their resource-intensive nature, dependencies on specific libraries and frameworks, and the need for version control and reproducibility.

6) Testing and Validation: Testing ML models is not as straightforward as testing traditional software systems. It involves evaluating model performance, generalization, and robustness across various data scenarios. Incorporating testing and validation practices into agile development, such as unit testing, integration testing, and validation with real-world data, is challenging and requires specialized expertise.

IV. Empirical Study

In the 2022–2023 academic year, the authors initiated an empirical study to gather qualitative data from students and ML professionals to inform the content of the new course. This involved conducting two separate online surveys, one for students and another for professionals. Additionally, they conducted in-depth virtual interviews with ML professionals who generously volunteered their time to meet with them. Below is provided a summary of the collected responses in separate sections.

A. Students Survey

The authors distributed an online questionnaire to a total of 349 students, including 232 former students and 117 current students. From this, they received a total of 67 student responses. The questionnaire aimed to assess students' interest and expectations for the new course. The respondents represented a diverse range of disciplines, including computer science, SE, data science, mechanical engineering, mathematics, and statistics. While most students had taken introductory courses on ML, AI, and SE, only a few had advanced ML or SE coursework. Next, the authors organized and described all student responses in different themes.

1) Student Interest: More than 90% of the students surveyed expressed a keen interest in the ML and SE combined course. They highlighted the need for practical implementation of ML algorithms within software, as existing ML courses often focus heavily on theory. Students desired the knowledge and skills to effectively utilize, share, and deploy ML models in real-world applications. They sought a course that bridged the gap between ML theory and its practical implementation in SE. One student remarked, *"Absolutely [I am interested in such a course]. A big disconnect in these [ML] classes: I am getting results and I am building cool models, but I do not know how to utilize, share, and deploy these models."*

2) Student Expectations and Concerns: Furthermore, 100% of the students who took the survey expressed high expectations for the course. They wanted to explore the applications of ML and gain insights into integrating ML systems with software. Emphasizing the importance of SE, reference architectures, design patterns, and ML-specific workflows, students aimed to develop practical skills in deploying and sharing ML models, working on real projects, understanding the integration of ML with agile SE methodologies, and becoming job-ready.

When asked about concerns, some students (i.e., less than 50%) mentioned potential difficulties in interpreting ML model outputs and uncertainty about the course project. More than 80% of the students also expressed a desire for more hands-on experience. As for suggestions, students recommended making ML and SE courses as prerequisites to avoid redundancy and to ensure a strong foundation. Moreover, they suggested structuring the course as a workshop-style format to enhance practical learning.

B. Professionals Survey

The online survey was completed by 47 professionals. The overall goal was twofold.

1) To find out about their challenges and listen to their stories (successes and failures) about deploying ML in an agile environment.
2) If they had any recommendations about crafting a course that can help future graduates with using ML effectively in their jobs.

1) Professional Background and Experience: Professionals from various backgrounds and roles responded with their current and/or former job titles, including technical consultants, software developers, heads of product, agile portfolio managers, scrum masters, project managers, system engineers, researchers, AI practice leads, IT consultants, bioinformaticians, engineering managers, software engineers, validation engineers, directors, architects, application developers, DevOps engineers, Web developers, instructional technologists, and systems analysts. Approximately, 50% of the professionals had more than 11 years of experience.

More than 90% of the professionals surveyed indicated that they have experience working on agile projects. They have worked as scrum masters, team members, and individual contributors in various capacities. Some professionals mentioned working in small teams, while others highlighted the utilization of agile practices in their organizations. There were also mentions of using a combination of waterfall and agile processes, as well as the adoption of scaled agile frameworks like SAFe [44]. Overall, professionals expressed familiarity with agile methodologies and their application in different contexts. For instance, one professional mentioned: *"Yes, [my] company's business unit widely follows agile. My team (8–10 devs) works in weekly sprints on tickets that cover the full stack spread of Web development. We host daily stand ups, weekly planning and refinement, and biweekly retrospectives."*

Most of the professionals shared their experiences with ML, indicating various levels of involvement. Some mentioned brief exploration during academic courses, while others mentioned ML being used in product increments or specific tools. More than 80% of the professionals mentioned working on projects that involved ML, such as sentiment analysis, automatic grading tools, predictive models for health decision making, and image processing/computer vision. Some professionals also mentioned utilizing tools like ChatGPT and GitHub Copilot. Overall, professionals had diverse experiences with ML, ranging from academic coursework to practical applications in their work. Here is a response: *"Yes, my team and I utilize ML frequently to develop predictive models to support individual and public health decision making."*

In addition, the authors asked all professionals whether they have used MLaaS in an agile environment. Approximately 40% of the respondents said that they had used MLaaS in some capacity and the rest 60% they had not.

2) Training and Development in ML: Professionals who participated in the survey provided valuable insights into their own experiences and motivations for learning ML. Approximately 40% of the professionals mentioned receiving formal training through graduate school courses, while the rest 60% emphasized being self-taught and augmenting their education with AI-related coursework. Their interest in ML stemmed from its remarkable ability to identify patterns, work toward desired outcomes, and its wide applicability across various domains. Many professionals expressed enthusiasm for the concept of teaching computers to think independently. Some even expressed their pursuit of additional ML classes or degrees in data science and ML to further enhance their knowledge and skills.

When it came to their organizations' SE practices, more than 95% of the professionals revealed that they follow an agile paradigm. They mentioned various agile approaches, such as Kanban, XP, Scrum, and commitment-based management. One professional shared their team's use of a simple storyboard to organize and track tasks, along with weekly ML meetings, within their small team. Overall, agile methodologies were prevalent among the professionals, with a strong emphasis on iterative and collaborative approaches to software development.

Professionals also provided insights into how their organizations train developers in ML. The approaches varied significantly, with some professionals mentioning seminars and formal classes as the primary training method. On-the-job training was highlighted as another crucial avenue for learning ML, allowing professionals to gain hands-on experience while working on projects. A combination of self-guided learning, in-house training programs, external vendor training, and collaborative initiatives were also mentioned. Some organizations emphasized collaborative approaches, such as pairing developers with experts, providing internal training, or bringing in external trainers, to facilitate learning. Individual development plans and self-driven learning were recognized as additional methods for fostering ML expertise. Overall, there was a diversity of approaches to training developers in ML, ranging from formal classes to on-the-job learning and collaborative initiatives.

3) Challenges and Insights in ML Development: More than 80% of the professionals mentioned various challenges they have faced during the development and deployment workflow of ML-based systems. These challenges include keeping the models up to date as data changes, model convergence and performance with real-world data, determining optimal parameters for training, online model updating and versioning, feasibility concerns, the importance of choosing appropriate models for problem domains, and the nonlinear nature of ML development compared to traditional software development processes. They mentioned that the unique characteristics of ML, such as the experimental nature and reliance on trial and error, can pose difficulties in integrating it into conventional software development processes. Here is a related response from a professional: *"ML doesn't follow normal software dev cycles. You do not have features to implement or nice clear work tickets to do. It tends to be experimental, and a good deal of trial and error. That can make it hard to work into normal software development processes."*

In addition, professionals shared their challenges when infusing ML during agile software development. One data scientist working on an ML-based recommendation system for an e-commerce platform faced difficulties in accessing and preprocessing the required data, as well as integrating the ML models into the existing architecture. Delays in these tasks impacted the project schedule. To overcome these challenges, they prioritized data access, established data cleaning protocols, and collaborated with the development team for architectural changes.

Another professional emphasized the barrier of AI-readiness of data for successful ML implementation. Some challenges mentioned were integrating ML into existing systems, organizational reluctance, or lack of seriousness toward ML, and the need for validation of ML/AI output. Despite the challenges, professionals expressed opportunities and the importance of safe validation in ML deployment.

4) Endorsement and Recommendations for New Course: All professionals unanimously supported the idea of offering a supporting course and provided recommendations for specific topics and skills they expected to see from students taking it. These included the ability to integrate ML with other systems, understanding experimental design and product integration, practical training in basic ML models, and applying them in realistic scenarios. Other suggested skills and knowledge encompassed collaboration, cross-functionality, statistics proficiency, a strong foundation in computer science and programming, understanding of ML basics, familiarity with popular ML frameworks, agile software development knowledge, problem-solving abilities, collaboration and communication skills, ethical considerations, SE principles, data understanding and QA, adaptability, knowledge of ML capabilities and applications, creative thinking, and evidence of care in tool creation with ML, including transparency and bias mitigation. The professionals emphasized a comprehensive range of skills, knowledge, and attitudes necessary for successful ML integration in software development.

Professionals provided valuable recommendations for the course topics, including data management techniques for preprocessing and cleaning, model selection and evaluation, deployment considerations for scalability and maintenance, integration of ML models with software development practices, ethical considerations addressing bias and privacy, effective collaboration in interdisciplinary teams, case studies of ML integration in agile projects, understanding APIs, emphasizing validation and testing strategies, and exploring ethical principles inspired by Isaac Asimov's Three Laws of Robotics [45].

5) Engagement and Seminars: More than 80% of the professionals expressed their willingness to engage with the students during the course, albeit with certain constraints and limitations. Some mentioned their limited availability, while others hesitated due to their role not being specifically in ML engineering or lacking expertise in the subject matter. Nonetheless, their willingness to contribute in some capacity demonstrates their support and potential for engagement with the students.

Finally, approximately 70% of the professionals expressed interest in having a seminar on the synergy of ML and agile development delivered to their organizations. Some indicated that they already have a similar course available, while others expressed a positive inclination or mentioned that their department or organization would likely be interested. The overall response indicates a strong potential for delivering such a seminar to meet the demand and interest within these organizations.

6) Professionals Interviews: In addition to completing the survey, eight experienced professionals agreed to participate in one-on-one virtual interviews with the authors. The rest of this section summarizes the recorded discussions with these volunteer professionals. During the interviews, the authors identified some common themes which are described in separate sections below. Below is provided a brief description of the overall background of these professionals without disclosing any personal information.

a) Professionals' background: Professionals are interviewed from the following industry domains: medicine, multivocational short-term development, AI research and development, cybersecurity, computer vision and robotics, bioinformatics, public health, and other IT-consulting industries. More specifically, one professional interviewed is the lead of the AI practice at a software development consulting firm, experienced in grants from DARPA. Another professional holds a Ph.D. in CS and supervises a team working on federal health projects with ML components. They specialize in data analytics, digital engineering, and cybersecurity. A Scrum coach with over 11 years of experience guides agile development at their organization. A scientist conducts AI research for computer vision and robotics, while also participating in AI and data science contests. Finally, a professional with 25+ years of experience develops AI-assisted systems for cloud-based chromatogram analysis.

b) Driving forces: Professionals expressed that the United States (U.S.) federal government is driving the adoption of ML and AI, making it an opportune time for data scientists and AI/ML professionals. The focus has shifted from research to practical deployment of models for everyday use. Regulatory agencies like the U.S. Food and Drug Administration (FDA) are investing in ML to enhance supply chain management. Concerns about explainable AI are diminishing as interest grows in AI Chatbots like ChatGPT, even though their inner workings may be difficult to comprehend. There is a demand for predictive capabilities and solutions to handle real-world data, such as analyzing interview conversations. Some related responses include: *"Federal government is pushing hard to use ML and AI more, so it is a good time to be in this space being a data scientist or an AI/ML practitioner."* and *"FDA and other agencies are investing large sums of money to get a better handle on the supply chain."*

c) ML and SE integration: All interviewed professionals showed strong support for integrating ML and agile software development, emphasizing that there are no inherent conflicts between the two. They believe that combining SE with ML is the future direction. They also highlighted the importance of avoiding waterfall approaches and embracing true agile practices for effective ML integration. To synchronize ML activities within agile slices, they recommended starting with a basic model that can be quickly trained and gradually refining it to the desired level of accuracy and fidelity. Here are a couple of quotes: *"There is nothing antithetical in agile that will preclude ML"* and *"I think that marrying SE with ML is the way of the future."*

d) Real projects with ML and agile development: Professionals shared various examples of ML applications within an agile environment during the interviews. These include projects focused on resilient supply chains for FDA, predictive tools for failure prediction in jet engines and water softeners, planning software for municipal highway funding, banking loan applications with iterative factor selection, deployment and monitoring of models in production at a bank, and time series analysis for predicting part failure in turbine engines. They also raised concerns about validating model performance and scaling up from development tests to production. A professional mentioned that: *"Predictive tools like failure prediction with, for instance, jet engines, or water softeners. We want to predict when they will fail so that we can service it before it fails."*

e) Challenges and lessons learned: Professionals shared valuable insights regarding challenges and lessons learned from their experiences while using ML within an agile environment. These include the challenge of combining SE and data science, difficulties in implementing true agile practices within government procurement, the need to slice functionality in banking loan applications, regulatory constraints in medical and banking applications, coordination and collaboration in agile ML development, data shaping for analytics, effective communication with nontechnical audiences, data quality and creation challenges, the problem of overfitting, iterative testing with imperfect data, extracting useful results from dirty datasets, and the value of "good enough" outcomes from ML. Here is a related quote: *"The combination of SE and DS is a real challenge that many organizations are facing right now. My employer has tried to develop or leverage frameworks that are becoming more of a standard in industry things like MLOps or AIOps which are frameworks that try to wed SE practices with DS practices where you get a deployed well-working model."*

f) Best practices of ML, MLaaS, and generative AI: The interviewees expressed their excitement and familiarity with various ML and generative AI best practices. They discussed MLaaS tools like DataRobot [46], the value and challenges of using MLaaS versus custom development, the benefits of autoML [4] in accelerating the ML process, the availability of pretrained ML models like Amazon Comprehend Medical [47], the potential of MLaaS and autoML in reducing custom development, the limitations of MLaaS in government cloud environments, concerns about the black-box nature of autoML, and the use of algorithm choice systems like Azure ML and Ensemble [48], [49]. These practices offer opportunities to streamline ML development and improve efficiency. Here is what a professional said: *"One of the things that I find really exciting is that thing called autoML, namely, various algorithms that can automate many steps in the ML workflow typically starting from selecting one or more ML algorithms, training the model, and producing an ensemble of models like a regression model with a support vector machine. It is a way to accelerate this ML process."*

g) Diverse teams: According to the professionals, teams handling ML-infused agile software projects should be formed with integrated members including data scientists, subject matter experts, cloud architects, software engineers, data engineers, devops professionals, and project managers. The industry has moved away from the notion that data scientists alone can handle all aspects of the project, recognizing the need for diverse skill sets. Some organizations have developers with data science knowledge who not only develop the ML models but also build the necessary infrastructure. In certain sectors like finance, dedicated data scientists may be involved in handling the data and models. Collaboration and teamwork are emphasized as essential for success in these projects. Here is an example of a related response: *"Another thing that our employer does [for ML & SE] is building teams that have a diverse array of skill sets, including data scientists, cloud architects, software engineers, data engineers, devops professionals, and a project manager in order to develop, deploy, and then monitor a (ML) model in production successfully."*

h) Training: Professionals stated that their organizations train people in ML through various methods. Apprenticeships are implemented where new individuals work closely with senior developers on projects. Familiarity with agile terminology and statistical fluency are sought in new recruits. In-house training is facilitated through the development of tutorials and the establishment of internal paper-reading groups. Additionally, external training resources provided by platforms like Azure are utilized. Salespeople are also trained in ML concepts. The organization may hire individuals with ML expertise to provide training, and external training programs offered by platforms are utilized as well.

i) Ethics, bias, and privacy: During the discussions, ethics, fairness, bias, privacy, and regulations were identified as crucial considerations in ML projects. Some organizations address these issues contractually, relying on external services for tasks like resume scanning to mitigate unfair outcomes. Nongovernmental organizations play a role in vetting groups, datasets, and ML models for bias. While some organizations invest significantly in ethical and responsible AI practices, it was noted that clients often allocate a small portion of their budget toward these aspects compared to model development. When using MLaaS, responsibility lies with the user to ensure ethical behavior. Assessing the technology, evaluating equitable training data, and implementing monitoring approaches to detect drift were highlighted as important

TABLE II
MATCHING INDUSTRY CHALLENGES WITH ACADEMIC LEARNING OBJECTIVES AND COURSE CONTENT

Industry Challenges	**Learning Objectives**	**Course Topics & Activities**
Data discovery, curation, dispersion, labeling, visualization.	Students learn how to manage big data.	Wrangle real data from industrial databases, applications, and devices.
Model convergence and performance with real-world data.	Learn how to use tools for tuning of models.	Tuning stock models for performance, using meta-learning.
Online model updating as data changes. Versioning, feasibility concerns, the importance of choosing appropriate models for problem domains.	Learn how to build systems so that different models and underlying algorithms can be switched-out.	Try making a generic interface to accept multiple models.
The non-linear nature of ML development compared to traditional software development processes.	Learn how to merge the flow of ML capabilities with other features whose availability is on a different schedule.	Practice situations with systems already built. Play the user in varying feedback to the model.
Determining optimal parameters for training.	Learn how to relate features to model training needs.	Set up a sequence of experiments to detect optimal models.
Merging ML models with other software that uses different tool sets.	Learn how to create interfaces between ML tools and models, and non-ML systems.	Add an ML model to an existing system.
Model complexity and interpretation, managing resources.	Learn how to train and test models on realistic data.	Try different algorithms, hyperparameters, etc. on the same data and compare results.
Applying metrics, regulation, and simulation to models.	Learn how to verify model predictions with users and for other considerations.	Interact with real users of the data to decide the value of ML results.
Operational support, deployment patterns, team dynamics, feedback, and continuous delivery.	Learn how to deploy models in realistic environments with model updating.	Integrate an ML system with an existing system that is designed to enhance.
Biases, fairness, authorship, implications of decisions, user involvement, explainability, security challenges.	Learn how to manage security, trust, and ethics issues.	Establish related criteria ahead of time, then test the ML system against these. Infuse checks into ML systems that show performance related to possible biases.
Infusing ML and data science within agile SE.	Learn how to coordinate ML and data science activities during agile software development.	Provide project opportunities that combine ML and data science with agile development.
Having teams with experts of diverse backgrounds including ML, SE, and DS.	Learn how to work with other experts and non-technical people effectively.	Form diverse teams with SE, ML, Math, and DS majors.

practices in ensuring responsible and ethical ML deployment. Here is a related response from a professional: *"There are nongovernmental organizations that vet groups or data sets or ML models for bias."*

j) Recommended student skills: Professionals provided valuable suggestions for preparing undergraduate students with the necessary skills for their future jobs. They emphasized the importance of exposing students to real-world data science problems, as they often lack experience with messy and unprocessed data. Data engineering was highlighted as a crucial aspect of data science, comprising a significant portion of the work. The professionals also emphasized the need for students to develop the ability to learn independently and avoid getting distracted by superficial aspects, encouraging them to gain a broader perspective on their learning journey. Here is a related quote: *"One of the things that I have observed over* *the years is the need that the data scientists who come out of school especially undergrads need to know how to deal with messy data."*

k) Suggested course topics: During the interviews the authors collected helpful suggestions for specific topics to include in our course content. These suggestions involve guiding students through the entire data cleaning and data engineering process, utilizing MLaaS solutions, such as pre-trained models or autoML, and incorporating agile practices. Additionally, it was recommended to focus on mathematical fluency and statistical analysis, addressing gaps in formal knowledge and providing explanations for algorithmic choices. A related publication titled *"Enterprise AIOps: A framework for enabling AI"* by Justin Neroda, Steve Escaravage, and Aaron Peters was also recommended as a valuable resource [50].

V. Proposed Course Content

Based on the collected empirical data and other industry challenges reported, for instance in [51], the authors tried to match such challenges with related academic learning objectives, course content, tools, techniques, methods etc., as shown in Table II. The next steps include, among others, preparation of a detailed course description, the syllabus, selection of a textbook and deciding on prerequisites.

As shown in Table II, industry challenges, such as data discovery, curation, dispersion, labeling, and visualization, will be addressed in the course with learning objectives focused on managing big data. Students can wrangle real data from industrial databases, applications, and devices. Other challenges include model convergence and performance with real-world data, online model updating, nonlinear nature of ML development, model complexity and interpretation, applying metrics and regulations, operational support and deployment patterns, biases and fairness, and infusing ML and DS within agile SE.

Moreover, the proposed course will cover topics, such as tuning models, building systems for model switching, merging ML capabilities with other features, training, and testing models on realistic data, verifying model predictions with users, deploying models in realistic environments, managing security, trust, and ethics, coordinating ML and data science activities in agile software development, and effective teamwork with diverse backgrounds.

Additional activities include tuning stock models, creating a generic interface for multiple models, practicing with already-built systems, interacting with real users, integrating ML systems with existing ones, establishing criteria, and testing for biases, and forming teams with various expertise.

One author gained teaching experience with a pilot SE for ML course in the 2020–2021 academic year, prior to the current survey. They find this experience valuable for shaping the desired course design. The required course project involved taking a previously designed ML algorithm from a prior AI course the student had taken and adapting it for a production system. The course covered most of the points listed in Table II below, and it was completed by twenty students successfully.

VI. Conclusion and Next Steps

In conclusion, the authors' experience in crafting a new course at the intersection of ML and agile SE indicates its potential as a valuable and appealing option for undergraduate students. The authors gathered qualitative results from surveys conducted with interested students and software professionals, supporting the claim of its usefulness. By aligning challenges faced by ML professionals with academic learning objectives, they designed the course content. Additionally, they incorporated student feedback, including their expectations and concerns, to shape the course.

In the future, the authors will consider further untapped synergies between ML and agile processes. For instance, ML can enhance agile practices by detecting bugs and issues in ongoing projects, integrating ML-based tools with existing project tools, and providing effective training on their utilization. ML also contributes to improving software systems' quality attributes, such as performance and reliability. ML's role in user interaction is another specialized area, where intelligent applications learn user behavior, collaborate with users, and adapt ML models based on user actions. ML systems facilitate collaboration among team members by notifying relevant changes and help clients prioritize tasks based on dependencies in the backlog.

Cloud providers already implement Continuous Integration/Continuous Delivery (CI/CD) of ML models, and agile metrics can be employed to assess overall project progress through intelligent burndown charts. ML models also aid in managing scope creep and offer valuable insights.

Moving forward, the authors will continuously refine and enhance the content of the proposed new course. It will be offered at the authors' institutions, with knowledge-sharing to improve the learning experience. They are exploring possibilities of cross-listing and co-teaching the course to enable collaboration among teams from different universities. Involving ML professionals as mentors for these teams and projects is also under consideration.

Other plans involve integrating ML as a black box within an existing Scrum-based SE course, incorporating it into a semester-long project. Additionally, the authors are considering a senior capstone course that combines ML with an agile SE project. Finally, they envision a future curriculum track or concentration that encompasses multiple related courses, covering agile SE, AI, ML, as well as cloud-based MLaaS topics.

References

[1] P. Gregory and K. Taylor, "Defining agile culture: A collaborative and practitioner-led approach," in *Proc. IEEE/ACM 12th Int. Workshop Cooperative Human Aspects Softw. Eng. (CHASE)*, 2019, pp. 37–38.

[2] "What is scrum? What is the definition of scrum?" Mountain Goat Software. Accessed: Sep. 25, 2023. [Online]. Available: https://www.mountaingoatsoftware.com/agile/scrum

[3] E. Hochmüller and R. T. Mittermeir, "Agile process myths," in *Proc. APSO*, Leipzig, Germany, May 2008, pp. 5–8.

[4] "What is automated machine learning (Auto ML)?" Microsoft Azure. Accessed: Sep. 9, 2023. [Online]. Available: https://learn.microsoft.com/en-us/azure/machine-learning/concept-automated-ml?view= azureml-api-2

[5] M. S. Rahman, F. Khomh, E. Rivera, Y.-G. Guéhéneuc, and B. Lehnert, "Challenges in machine learning application development: An Industrial experience report," in *Proc. SE4RAI*, Pittsburgh, PA, USA, May 2022, pp. 21–28.

[6] J. Schleier-Smith, "An architecture for agile machine learning in real-time applications," in *Proc. KDD*, Sydney, NSW, Australia, Aug. 2015, pp. 2059–2068.

[7] J. Langford, "Machine learning and algorithms; agile development," *Commun. ACM*, vol. 55, no. 8, pp. 11–12, Aug. 2012.

[8] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice*, 3^{rd} ed. Boston, MA, USA: Addison-Wesley, 2013.

[9] S. Amershi et al., "Software engineering for machine learning: A case study," in *Proc. IEEE/ACM 41st Int. Conf. Softw. Eng. Softw. Eng. Pract. (ICSE-SEIP)*, Montreal, QC, Canada, 2019, pp. 291–300, doi: 10.1109/ICSE-SEIP.2019.00042.

[10] L. Baier, F. Jöhren, and S. Seebacher, "Challenges in the deployment and operation of machine learning in practice," in *Proc. 27th Eur. Conf. Inf. Syst.*, Stockholm, Sweden, 2019, p. 136.

[11] A. Paleyes, R.-G. Urma, and N. D. Lawrence, "Challenges in deploying machine learning: A survey of case studies," *ACM Comput. Surveys*, vol. 5, no. 6, p. 114, 2023. [Online]. Available: https://doi.org/10.1145/3533378

[12] A. Serban, K. van der Blom, H. Hoos, and J. Visser, "Adoption and effects of software engineering best practices in machine learning," in *Proc. 14th ACM/IEEE Int. Symp. Empir. Softw. Eng. Meas.*, Bari, Italy, p. 3, pp. 1–12.

[13] M. Hesenius, N. Schwenzfeier, O. Meyer, W. Koop, and V. Gruhn, "Towards a software engineering process for developing data-driven applications," in *Proc. IEEE/ACM 7th Int. Workshop Realizing Artif. Intell. Synergies Softw. Eng. (RAISE)*, 2019, pp. 35–41.

[14] P. Y. Simard et al., "Machine teaching—A new paradigm for building machine learning systems patrice," Aug. 2017, *arXiv:1707.06742v3*.

[15] Q. Guo et al., "An empirical study towards characterizing deep learning development and deployment across different frameworks and platforms," in *Proc. 34th IEEE/ACM Int. Conf. Autom. Softw. Eng. (ASE)*, 2019, pp. 1–13.

[16] Z. Chen et al., "An empirical study on deployment faults of deep learning based mobile applications," in *Proc. IEEE/ACM 43rd Int. Conf. Softw. Eng. (ICSE)*, 2021, pp. 674–685.

[17] L. Oakden-Rayner, J. Dunnmon, G. Carneiro, and C. Re, "Hidden stratification causes clinically meaningful failures in machine learning for medical imaging," in *Proc. ACM CHIL*, Toronto, ON, Canada, Apr. 2020, pp. 151–159.

[18] L. Rivero et al., "Deployment of a machine learning system for predicting lawsuits against power companies: Lessons learned from an agile testing experience for improving software quality," in *Proc. 19th Braz. Symp. Softw. Qual.*, São Luís, Brazil, Dec. 2020, pp. 1–10.

[19] T. Chandra, "Regressing towards simpler prediction systems," in *Proc. 8th ACM Int. Conf. Web Search Data Min.*, Shanghai, China, Feb. 2015, pp. 335–336.

[20] D. Zhang and J. J. P. Tsai, *Advances in Machine Learning Applications in Software Engineering*. Hershey, PA, USA: Idea Group Publ., 2007.

[21] "Dotscience: The state of development and operations of AI." 2019. Accessed: Sep. 25, 2023. [Online]. Available: https://dotscience.com/assets/downloads/Dotscience_Survey-Report-2019.pdf

[22] (Alegion, Dublin, Ireland). *Dimensional Research: Artificial Intelligence and Machine Learning Projects Are Obstructed by Data Issues Global Survey of Data Scientists, AI Experts and Stakeholders*. May 2019. Accessed: Sep. 25, 2023. [Online]. Available: https://cdn2.hubspot.net/hubfs/3971219/Surv%20Asse%201905/Dimension%20Resear%20Machi%20Learni%20P%20Repo%20FINAL.pdf

[23] K. Wiggers. "For 1 in 4 companies, half of all AI projects fail." IDC. Jul. 2019. Accessed: Sep. 25, 2023. [Online]. Available: https://venturebeat.com/2019/07/08/idc-for-1-in-4-companies-half-of-all-ai-projects-fail/

[24] (IDC, Needham, MA, USA). *Investment in Artificial Intelligence Solutions Will Accelerate as Businesses Seek Insights, Efficiency, and Innovation, According to a New IDC Spending Guide*. (Aug. 30, 2021). Accessed: Sep. 9, 2023. [Online]. Available: https://www.idc.com/getdoc.jsp?containerId= prUS48191221

[25] K. Wiggers. "The machine: Making sense of AI." Dec. 11, 2019. Accessed: Sep. 25, 2023. [Online]. Available: https://venturebeat.com/2019/12/11/algorithmia-50-of-companies-spend-upwards-of-three-months-deploying-a-single-ai-model/

[26] F. Ishikawa and Y. Nobukazu, "How do engineers perceive difficulties in engineering of machine-learning systems? Questionnaire survey," in *Proc. IEEE/ACM Joint 7th Int. Workshop Conducting Empir. Stud. Ind. (CESI) 6th Int. Workshop Softw. Eng. Res. Ind. Pract. (SER&IP)*, 2019, pp. 2–9.

[27] J. DeNero and D. Klein, "Teaching introductory artificial intelligence with Pac-Man," in *Proc. AAAI Conf. Artif. Intell.*, vol. 24, 2010, pp. 1885–1889.

[28] D. Kumar and L. Meeden, "A robot laboratory for teaching artificial intelligence," *ACM SIGCSE Bull.*, vol. 30,, no. 1, pp. 341–344, 1998.

[29] M. Pantic, R. Zwitserloot, and R. J. Grootjans, "Teaching introductory artificial intelligence using a simple agent framework," *IEEE Trans. Educ.*, vol. 48, no. 3, pp. 382–390, Aug. 2005.

[30] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. London, U.K.: Pearson, 2020.

[31] C. Kästner and E. Kang, "teaching software engineering for AI-enabled systems," in *Proc. Int. Conf. Softw. Eng. Educ. Train. Track (ICSE-SEET)*, May 2020, pp. 45–48.

[32] Southampton Course, Introduction to Machine Learning. Accessed: Sep. 25, 2023. [Online]. Available: https://www.southampton.ac.uk/courses/modules/feeg6042#overview

[33] University of Alberta Course, Dept. Comput. Sci., Univ. Alberta, CMPUT 466—Introduction to Machine Learning. Accessed: Sep. 25, 2023. [Online]. Available: https://www.ualberta.ca/computing-science/undergraduate-studies/course-directory/courses/introduction-to-machine-learning

[34] MIT Course, MIT, Machine Learning for Business, Engineering, and Science: Examples, Applications, and Case Studies. Accessed: Sep. 25, 2023. [Online]. Available: https://sl-onlinetraining.xpro.mit.edu/ice9/pdfs/agenda/MIT/MitxP%20EL%20V6-1712.pdf

[35] UC Berkeley Course, UC Berkeley, CS 253: Software Engineering and Artificial Intelligence. Accessed: Sep. 25, 2023. [Online]. Available: https://www2.eecs.berkeley.edu/Courses/CS253/

[36] CMU Course, Machine Learning in Production/AI Engineering. Accessed: Sep. 25, 2023. [Online]. Available: https://ckaestne.github.io/seai/

[37] G. Hulten, *Building Intelligent Systems: A Guide to Machine Learning Engineering*. New York, NY, USA: Apress, 2019.

[38] "Best practices for implementing machine learning on Google cloud." Cloud Architecture Center. Accessed: Sep. 25, 2023. [Online]. Available: https://cloud.google.com/architecture/ml-on-gcp-best-practices

[39] Microsoft Advanced Courses, The Era of AI is Now, Build the Future Faster With Microsoft AI. Microsoft Build, May 2023. Accessed: Sep. 25, 2023. [Online]. Available: https://learn.microsoft.com/en-us/training/paths/build-ai-solutions-with-azure-ml-service/

[40] Azure ML Ops Course, Udemy, Azure machine learning & MLOps: Beginner to advance. [Online]. Available: https://www.udemy.com/course/azure-machine-learning-mlops

[41] Azure Cloud Courses, Microsoft, Student Developer Resources—Gain Skills to Jump-Start Your Career and Make a Positive Impact on the World. Accessed: Sep. 25, 2023. [Online]. Available: https://azure.microsoft.com/en-us/resources/students/?activetab= pivot:githubtab

[42] "Introducing ChatGPT." [Online]. Available: https://openai.com/blog/chatgpt

[43] "GPT 3.5." Accessed: Oct. 5, 2023. [Online]. Available: https://chat.openai.com/#

[44] "SAFe, provided by scaled agile." Accessed: Sep. 9, 2023. [Online]. Available: https://scaledagileframework.com

[45] "ISAAC Asimov's three laws of robotics." Accessed: Sep. 25, 2023. [Online]. Available: https://en.wikipedia.org/wiki/Three_Laws_of_Robotics

[46] "Experience the impact of value-driven AI." DataRobot. Accessed: Sep. 25, 2023. [Online]. Available: https://www.datarobot.com

[47] "Amazon comprehend medical: Extract information from unstructured medical text accurately and quickly." AmazonComprehendMedical. Accessed: Sep. 25, 2023. [Online]. Available: https://aws.amazon.com/comprehend/medical/

[48] "Azure machine learning: Use an enterprise-grade AI service for the end-to-end machine learning lifecycle." Accessed: Sep. 25, 2023. [Online]. Available: https://azure.microsoft.com/en-us/products/machine-learning

[49] E. Lutins. "Ensemble methods in machine learning: What are they and why use them?" Accessed: Sep. 25, 2023. [Online]. Available: https://towardsdatascience.com/ensemble-methods-in-machine-learning-what-are-they-and-why-use-them-68ec3f9fef5f

[50] J. Neroda, S. Escaravage, and A. Peters, *Enterprise AIOps: A Framework for Enabling Artificial Intelligence*, 1st ed. Sebastopol, CA, USA: O'Reily, 2021.

[51] S. Chenoweth and P. K. Linos, "Qualitative findings from an online course on machine learning," in *Proc. IEEE Front. Educ. Conf.*, San Jose, CA, USA, 2018, pp. 1–9. [Online]. Available: https://doi.org/10.1109/FIE.2018.865