This research paper tries to pinpoint that how the software industry changes with the advancement of disruptive technologies in the market. The latest trend to hit the software industry is around integrating artificial intelligence (AI) capabilities based on advances in machine learning. Microsoft is using this AI and ML based concepts to design applications like Bing search or cortana the virtual assistant. It discusses how Microsoft software teams build applications with customer focused AI features. For that, Microsoft has integrated existing Agile software engineering process with AI-specified workflows

They found out 3 fundamental differences to building applications and platforms for training and fielding applications and platforms for training and fielding machine-learning models:

1. ML is all about data.
2. Building customizing and extensibility of model doesn’t only require SE skills but also deep knowledge of ML to build and evaluate model from scratch.
3. Difficult to maintain strict module boundaries between machine learning components than for SE modules

They found out nine stages of ML workflow that are both data and model oriented. In the model requirements stage, designers decide which features are feasible to implement with machine learning and which can be useful for a given existing product or for a new one. During data collection, teams look for and integrate available datasets (e.g., internal or open source) or collect their own. Data cleaning involves removing inaccurate or noisy records from the dataset, a common activity to all forms of data science. Data labelling assigns ground truth labels to each record. Feature engineering refers to all activities that are performed to extract and select informative features for machine learning models. During model training, the chosen models (using the selected features) are trained and tuned on the clean, collected data and their respective labels. In model evaluation, the engineers evaluate the output model on tested or safeguard datasets using pre-defined metrics.

They conducted interviews with snowball sampling strategy to know their familiarization with AI. They interviewed with 14 software engineers and specialized interviews according to their role. They also used questionnaire based on the interview Respondents used a broad spectrum of ML approaches to build their applications, from classification, clustering, dynamic programming, and statistics, to user behavior modeling, social networking analysis, and collaborative filtering. However, achieving this level of integration can be challenging because of the different characteristics of ML modules compared with traditional software components.

Since many machine learning techniques are centered on learning from large datasets, the success of ML-centric projects often heavily depended on data availability, quality and management. They found that Microsoft teams have found it necessary to blend data management tools with their ML frameworks to avoid the fragmentation of data and model management activities.

The integration of machine learning continues to become more ubiquitous in customer-facing products. Thus, engineers with traditional software engineering backgrounds need to learn how to work alongside of the ML specialists. Debugging activities for components that learn from data not only focus on programming bugs, but also focus on inherent issues that arise from model errors and uncertainty. A number of teams have found it important to employ rigorous and agile techniques to evaluate their experiments. To ensure that system deployment goes smoothly, several engineers recommend not only to automate the training and deployment pipeline, but also to integrate model building with the rest of the software, use common versioning repositories for both ML and non-ML codebases, and tightly couple the ML and non-ML development sprints and stand-ups.

Selecting the right ML model is essential for achieving desired performance. The authors discuss techniques for comparing and evaluating different models, including cross-validation and hyper parameter tuning. Effective documentation and collaboration are essential for ensuring the maintainability and scalability of ML projects. The paper outlines best practices for documenting ML workflows and fostering collaboration among team members.

An extensive portfolio of AI applications and platforms was made by integrating machine learning into existing software engineering processes and cultivating and growing ML talent. In this paper, they described the results of a study to learn more about the process and practice changes undertaken by a number of Microsoft teams in recent years. From these findings, they synthesized a set of best practices to address issues fundamental to the large-scale development and deployment of ML-based applications.

The paper concludes by summarizing the key findings and highlighting the importance of applying software engineering principles to ML projects. It emphasizes the need for interdisciplinary collaboration between software engineers and ML researchers to address the unique challenges of building ML-powered software systems.

Overall, the paper provides valuable insights and practical recommendations for integrating ML into real-world software engineering practices.