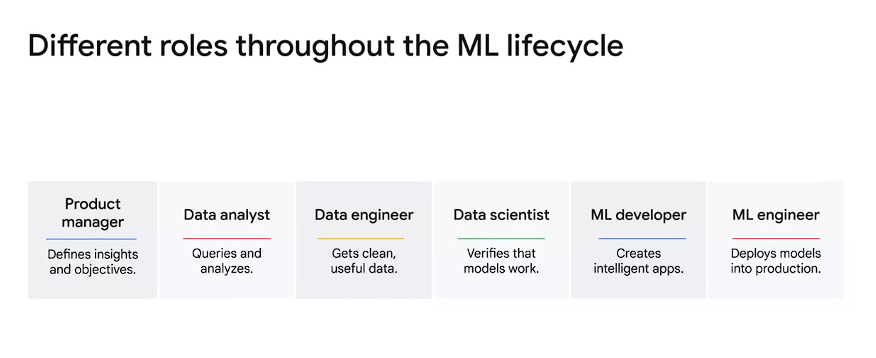
Absolutely, here are the key points from Esra Erdogan's introduction:

1. **Introduction to MLOps**:
   * Machine Learning Operations (MLOps) is a practice aiming to unify ML system development (Dev) and ML system operations (Ops).
   * Focuses on addressing challenges related to reproducibility of machine learning models.
   * Shares principles and tools with DevOps, aiming for shortened development lifecycles and maintaining high-quality software in production.
2. **Target Audience**:
   * Aimed at machine learning data scientists, engineers, and analysts.
   * Assumes proficiency in Python and foundational knowledge of machine learning concepts and building solutions on Google Cloud.
3. **Course Structure**:
   * Divided into two parts.
     + **Part 1: Employing Machine Learning Operations**:
       - Examines challenges faced by ML practitioners.
       - Introduces DevOps in the context of machine learning.
       - Explores the machine learning lifecycle.
     + **Part 2: Understanding Vertex AI**:
       - Explains the importance of Vertex AI as a unified platform.
       - Details the MLOps capabilities within Vertex AI.
       - Explores how Vertex AI facilitates MLOps workflows through hands-on labs and real-world use cases.
4. **Hands-on Learning**:
   * Focuses on a high-value real-world use case: Predicted Customer Life Value (CLV).
   * Progresses from a local workflow with BigQuery and TensorFlow to training and deploying models on Google Cloud using Vertex AI.
   * Includes graded assessments to test knowledge.
5. **Course Outcome**:
   * Equips learners to understand ML projects from an operational perspective.
   * Highlights the necessity of operationalizing ML models in unified AI platforms like Vertex AI.

]

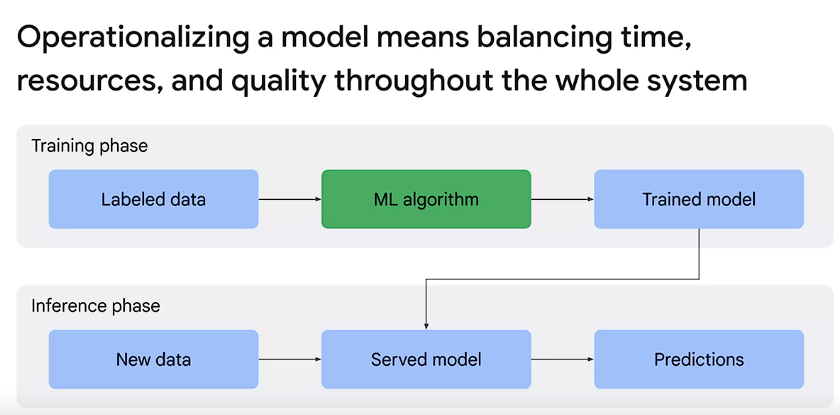
overview of the challenges ML practitioners face when operationalizing and making their models available for production. Here's a breakdown of these challenges:



1. **Complex Management**:
   * **Data, Models, and Architectures**: Keeping track of various elements like data, model architectures, hyperparameters, and experiments.
   * **Version Control**: Managing different versions of models and their corresponding code.
   * **Experimentation Control**: Handling different training procedure parameters, hyperparameter values, and performance metrics across trials.
2. **Monitoring and Benchmarking**:
   * **Iterative Monitoring**: Constantly tracking changes, implemented ideas, their success rates, and performance metrics to identify the best model for a specific use case.
   * **Benchmarking Models**: Evaluating **Best models** against each other to pinpoint the best-performing one.
   * **Best models refer to one that delivers the ideal result for specific use case**
3. **Collaboration and Teamwork**:
   * **Cross-functional Collaboration**: Working with various roles including data scientists, data engineers, ML engineers, application developers, site reliability engineers, business analysts, and users to operationalize ml models.
4. **Reproducibility**:
   * **Deployment Challenges**: Difficulty deploying a model to production if **it cannot be reproduced.**
   * **Policy Compliance**: Addressing policies or regulations that discourage or disallow bypassing reproducibility.
   * **Rerun and Parameter Sweep**: Desire to rerun the best model with a more comprehensive parameter sweep for improved results.
5. **Performance and Agility**:
   * **Streamlined Deployment**: Achieving streamlined training and making models production-ready for enhanced performance and agility.
   * **Automation Benefits**: Reducing errors through automation, even with a manual review step in the pipeline.
6. **Regular Updates and Traceability**:
   * **Regular Model Updates**: Need for regular model updates as new data arrives for a production application.
   * **Traceability Importance**: Ensuring traceability for maintaining and tracking changes effectively.

These challenges collectively underscore the complexity and critical considerations involved in operationalizing machine learning models for real-world production environments

**Overview of Mitigating Challenges Faced by ML Practitioners:**

****

**Think of MLOps as a lifecycle management discipline for machine learning. Its goal is a balanced process-driven approach to the management of resources, data, code, time, and quality to achieve business objectives and meet regulatory concerns**

Operational Lens Approach:

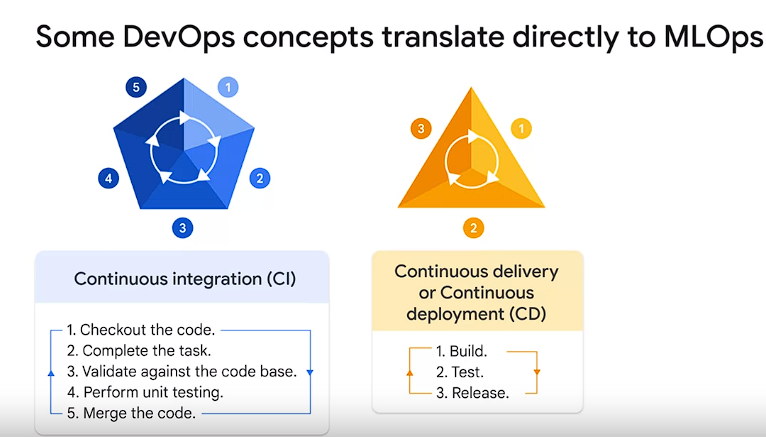
* Balancing time, resources, and quality in the entire system during operationalization.
* MLOps mirrors DevOps in **streamlining AI and ML projects**.

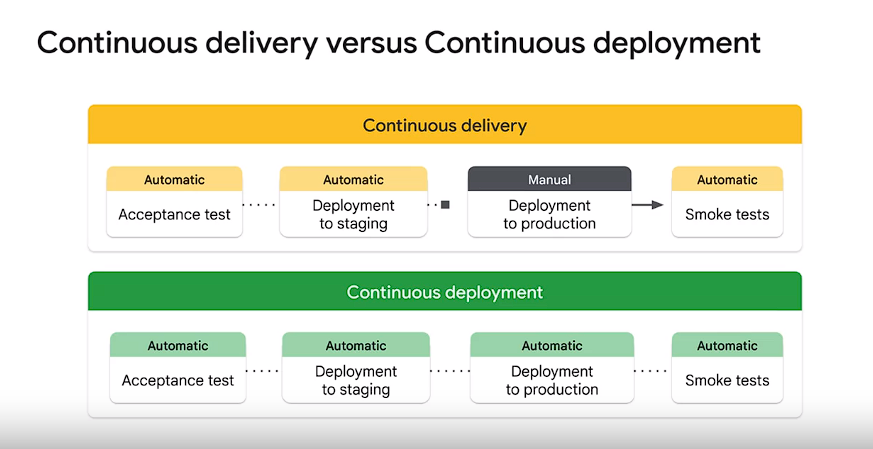
Similarities between DevOps and MLOps:

* Code Repository and Source Control:
  + Developers work on separate tasks in parallel using a branching strategy.
  + Merging efforts after working on code in local directories.
* Handling Code:
  + Downloading code, making changes locally, ensuring changes match the latest version, performing unit tests, and frequent merges.

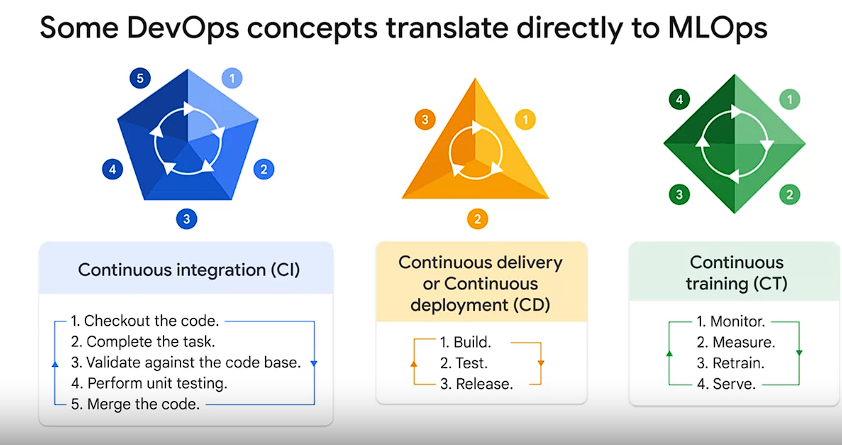
Continuous Integration and Deployment (CI/CD):

* Continuous Integration (CI):
  + Automates building and testing applications, frequently integrating changes with the main code.
* Continuous Delivery (CD):
  + Automates integration, deployment to staging, and smoke tests. Manual deployment to production.
* Continuous Deployment:
  + Automates configuration and deployment to production alongside CI.





Continuous Training (CT) for ML Models:



* Models need continuous training to adapt to changes in data profiles before redeployment.

Focus and Differentiation of MLOps:

* Continuous integration now involves testing and validating data, schemas, and models.
* MLOps incorporates monitoring, retraining, and serving models automatically.

Technical Debt in ML:

* Represents a compromise between faster delivery and high quality.
* Building an ML model isn't the challenge; operating an integrated ML system continuously is.

**Phases of Developing and Publishing a Machine Learning Model:**

Operationalizing a Model:

* Balancing time, resources, and quality throughout the system.
* MLOps as a lifecycle management discipline.

Similarities between DevOps and MLOps in Workflow:

* Code repositories, source control, and branching strategies.
* Continuous integration, delivery, and deployment.

Challenges with Continuous Training and ML Complexity:

* Challenges related to multi-functional teams, experimental nature of ML, and testing complexities.
* Deployment complexities involving multi-step pipelines for retraining and deploying models.

Deployment Challenges:

* Concept drift and model decay due to changing data profiles.
* Challenges with deployment as it involves multi-step pipelines for model deployment.

Strategies to Mitigate Technical Debt in ML Systems:

* Approaches to mitigate the accumulation of technical debt in ML systems.

These points encompass the strategies and challenges involved in operationalizing machine learning mod