

Автоматизация разработки и эксплуатации программного обеспечения (осень 2022 года)



ИУ-5, бакалавриат, курс по выбору



DevOps для задач обработки данных и машинного обучения

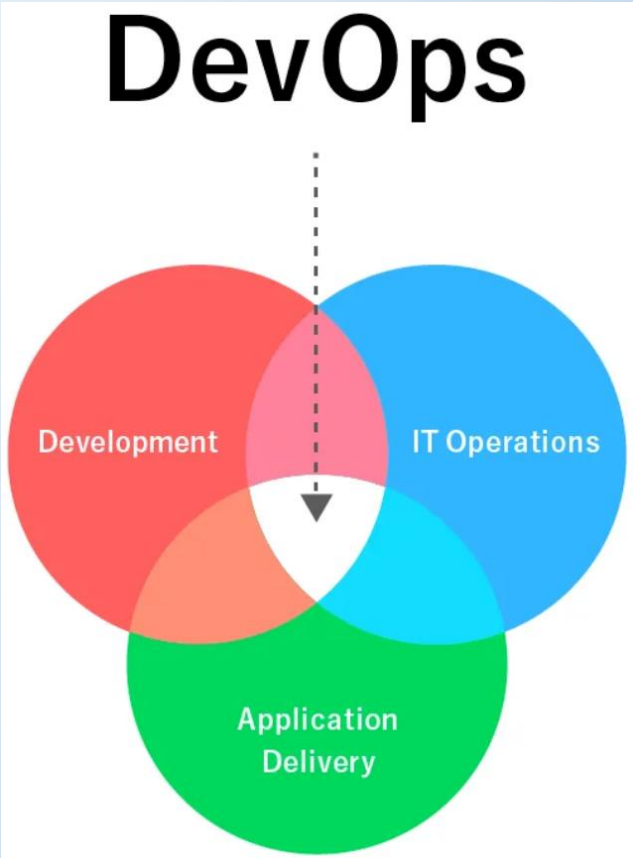
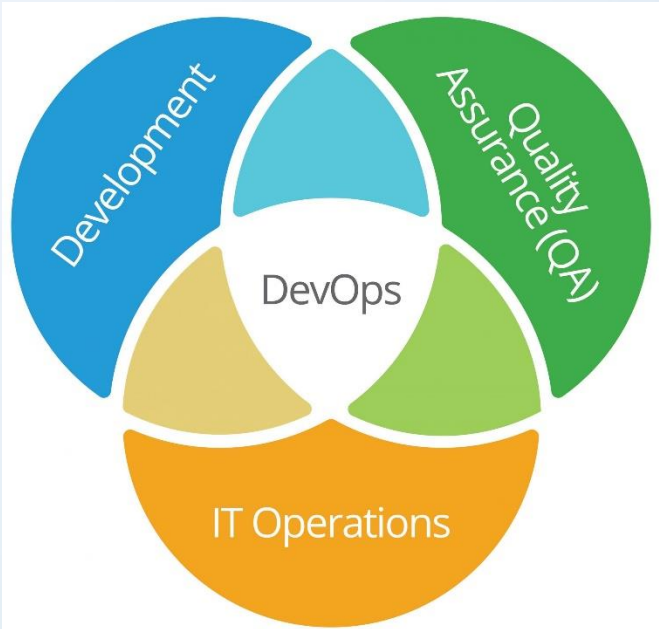
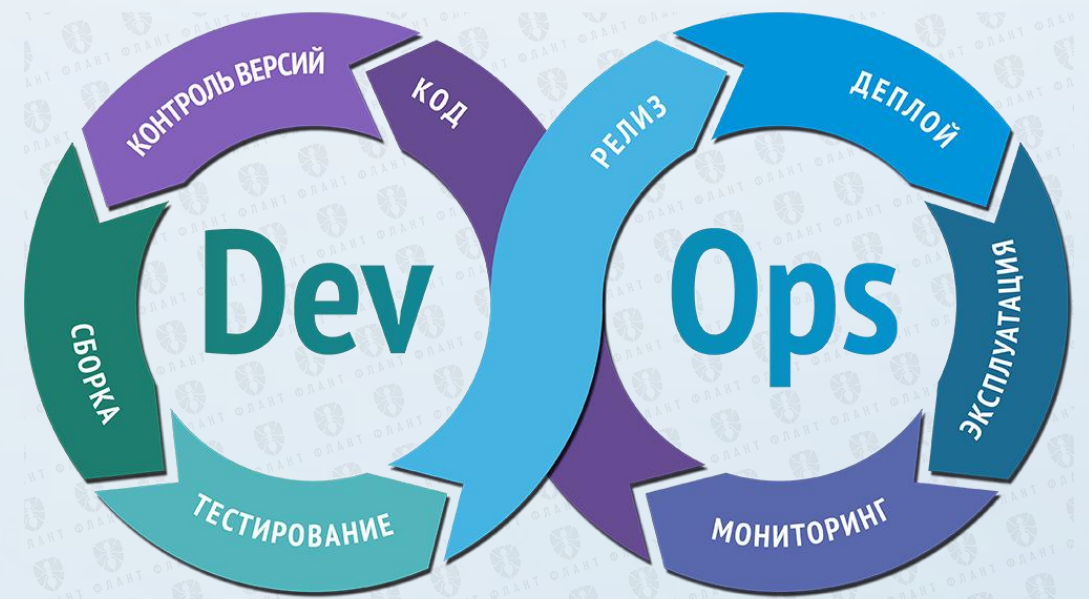
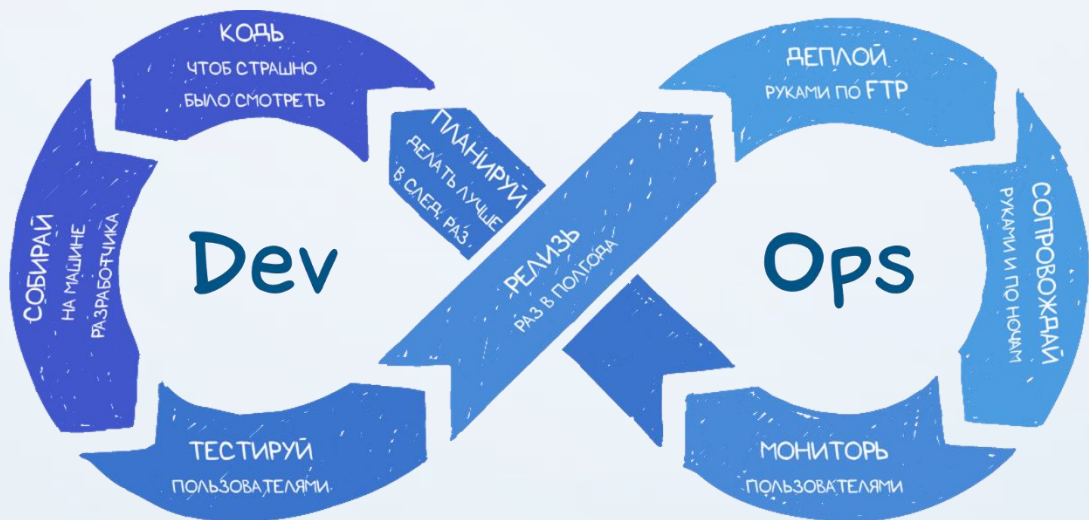
Содержание

1. Разновидности *-Ops: TechOps, DevOps, NoOps, DevSecOps.
2. Разновидности *-Ops для обработки данных и машинного обучения: DataOps, ModelOps, MLOps.
3. Инструменты MLOps и примеры их использования.

Разновидности *-Ops

- Что означают термины TechOps, DevOps, NoOps, DevSecOps?
- Специализации *-ops специалистов.
- TechOps
 - Как правило это наименование должности, роли. Сотрудники на этой должности помогают команде разработчиков уменьшить нагрузку, чтобы те могли сфокусироваться на разработке ПО.
 - TechOps не включает в себя разработку приложений, систем или программного обеспечения. Основная задача TechOps – это помощь всем должностям в IT во всем, кроме разработки ПО.
- DevOps
 - В отличие от TechOps, DevOps это не должность, а в большей степени культура, набор практик и способ работы для повышения эффективности обслуживания IT, особенно развертывания и разработки ПО. Основная задача DevOps – гарантировать выпуск качественного ПО в максимально сжатые сроки и максимально эффективным способом.

DevOps



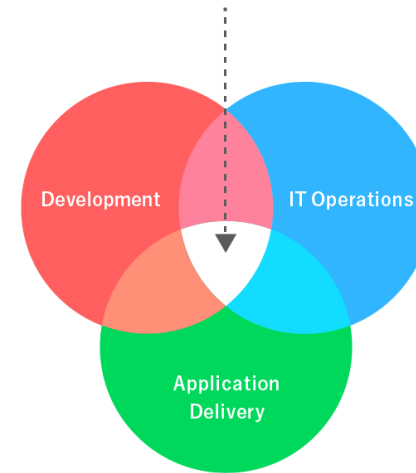
NoOps

- NoOps (No Operations) – это подход, при котором некоторые процессы эксплуатации отсутствуют, например, процессы развертывания и подготовки к использованию.
- Основная задача NoOps – улучшение процесса развертывания приложений, при котором исключается взаимодействие между отделами разработки и эксплуатации.
- Реализация такого подхода возможна при следующих условиях (*):
 - Разработчики сами выполняют операции по миграции окружения разработки с тестового сервера на производственный (в production) без привлечения дополнительного звена в виде DevOps-инженеров.
 - В качестве production-сервера используются облачные решения: PaaS (Platform as a Service) и IaaS (Infrastructure as a Service), в которых, благодаря технологиям виртуализации и типовым вариантам рабочих сред, разработчик может самостоятельно развернуть нужное рабочее окружение за несколько минут.
- Таким образом, NoOps пропагандирует отказ от DevOps-инженеров, как от дополнительных участников Agile-команды по разработке ПО, при сохранении всех принципов и положений самой концепции DevOps.
- Преимущества подхода NoOps:
 - Отсутствие расходов на DevOps-инженеров.
 - Повышение вовлеченности и ответственности за итоговый результат специалистов из отдела разработки.
 - Более строгое соответствие методологии Agile (скорость, адаптивность, частые релизы, возможности оперативно реагировать на запросы и не бояться ошибок, так как всегда возможен возврат к предыдущей версии).
 - Повышение скорости и качества операционных процессов благодаря использованию инструментов автоматизации и внедрению облачных решений.
- Поскольку условия (*) являются достаточно серьезными ограничениями, то подход DevOps продолжает активно развиваться. NoOps можно рассматривать как достаточно узкий подход, который годится лишь для некоторых команд.

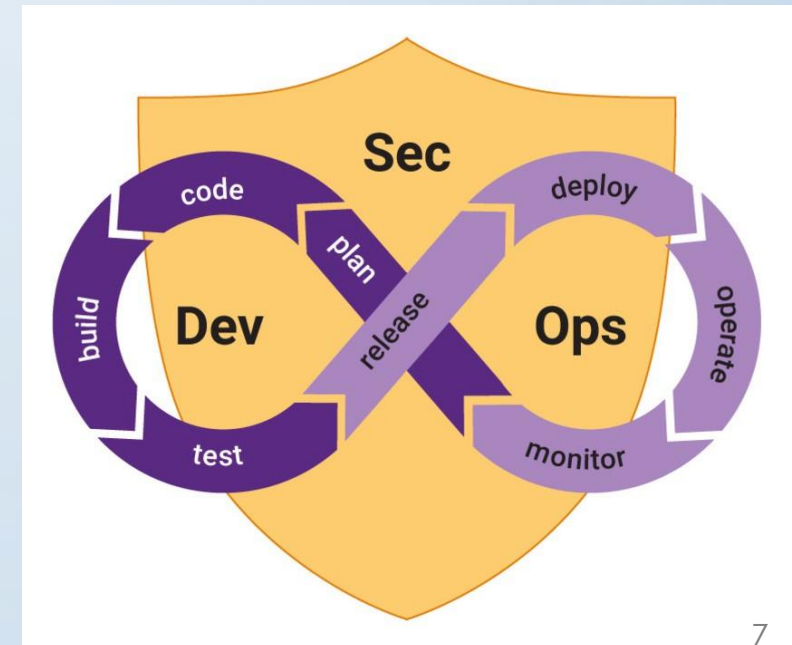
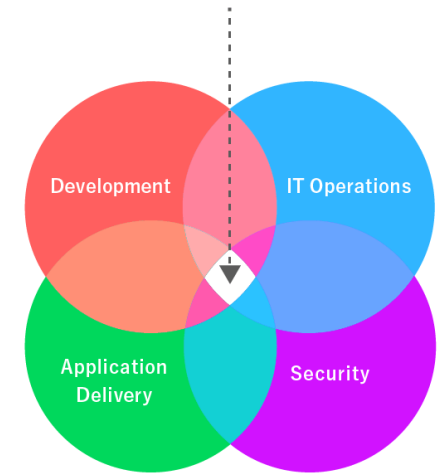
DevSecOps-1

- Инженер DevSecOps автоматизирует обеспечение информационной безопасности на всех этапах жизненного цикла разработки программных продуктов, от первоначального проектирования до интеграции, тестирования, развертывания и доставки.
- До появления DevOps проверка безопасности продуктов выполнялась на заключительных этапах жизненного цикла разработки ПО. Поскольку основное внимание уделялось разработке, проверка безопасности считалась менее важной чем другие этапы.
- Этот подход работал, пока обновления ПО выпускались всего один или два раза в год. Когда разработчики начали применять методики Agile и DevOps, циклы разработки ПО сократились до нескольких недель или даже дней, а традиционный подход к безопасности стал неактуальным.
- DevSecOps решает проблемы на том этапе, когда ошибки легче, быстрее и дешевле исправлять (до того, как код будет запущен на серверах). Кроме того, DevSecOps делает безопасность приложений и инфраструктуры общей ответственностью групп разработки, безопасности и эксплуатации, а не единоличной ответственностью подразделений безопасности.

DevOps



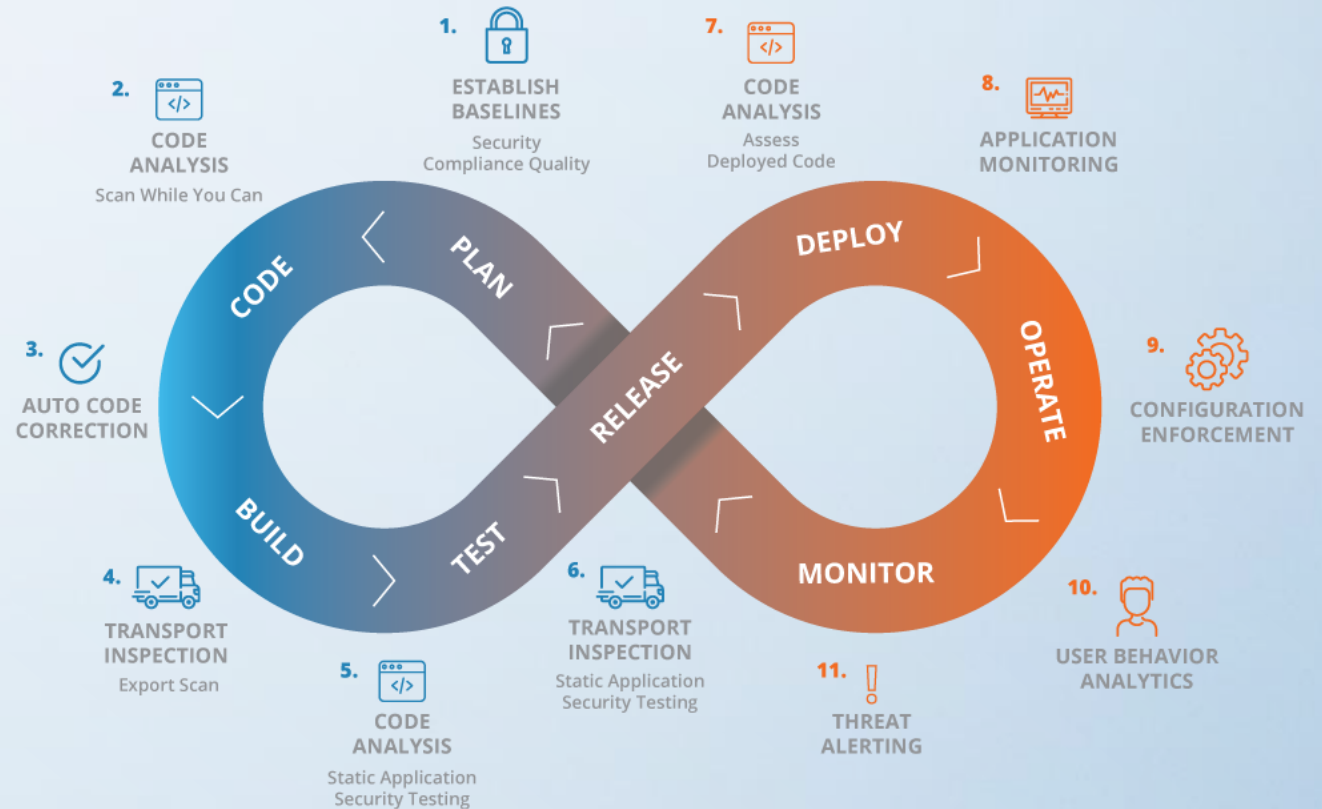
DevSecOps



DevSecOps-2

Типичный рабочий процесс DevOps и DevSecOps:

- Разработчик создает код и вносит его в систему управления контролем версий.
- Другой разработчик извлекает код из системы управления версиями и выполняет статический анализ кода для выявления любых дефектов безопасности или ошибок качества кода.
- Затем создается среда с использованием инструмента «инфраструктура как код». Приложение развернуто, и к системе применены настройки безопасности.
- После этого для вновь развернутого приложения выполняется набор автоматического тестирования, включая серверную часть, пользовательский интерфейс, интеграцию, тесты безопасности и API.
- Если приложение проходит эти тесты, оно развертывается в производственной среде.
- Эта новая производственная среда постоянно контролируется для выявления любых активных угроз безопасности для системы.

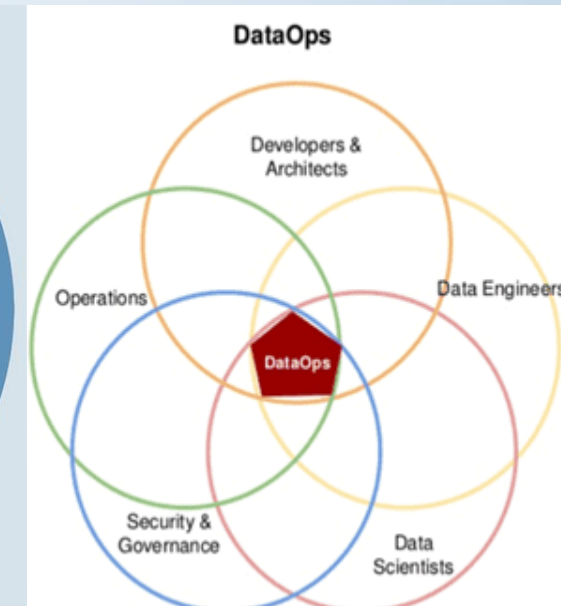
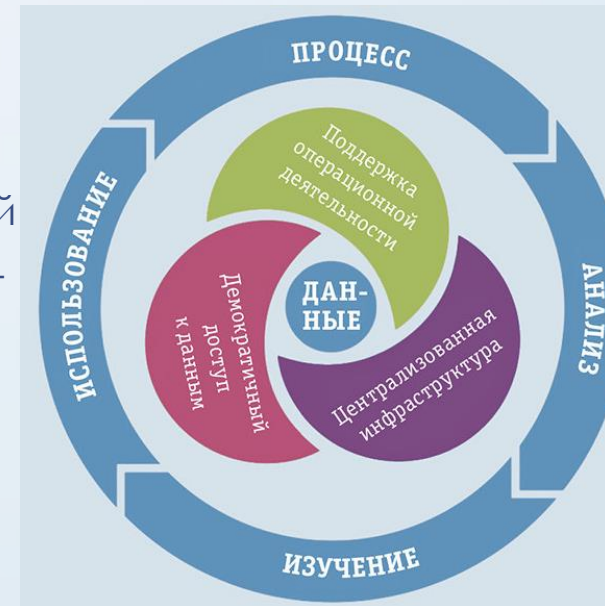


Разновидности *-Ops для обработки данных и машинного обучения

- Что означают термины:
 - DataOps,
 - AIOps,
 - MLOps,
 - ModelOps?

DataOps (BigData+Ops) – 1

- «DataOps – это автоматизированная и ориентированная на процессы методология. Она используется командами дата-аналитиков для повышения качества и сокращения временного цикла в аналитике данных». Википедия.
- DataOps — способ управления данными, обеспечивающий коммуникации и интеграцию уже имеющихся данных, команд и систем, позволяющий получить преимущества от изменения, перестройки оргструктуры и технологий для поддержки взаимодействия между теми, кто собирает и готовит данные, и теми, кто их анализирует и применяет в бизнесе.
- Главная задача DevOps – предоставить бизнесу работающее ПО. Задача DataOps – предоставить предприятию актуальные работающие данные.
- DataOps-инженеры особенно нужны в Big Data команде.
- Три основные идеи, на которых фокусируется DataOps:
 - Работа с данными должна быть воспроизводимой.
 - Аналитика как код (любые действия могут быть описаны кодом).
 - Данные как платформа (данные могут являться основой для разработки ML и AI-приложений; данные очищены, предобработаны, к ним обеспечен доступ и т.д.).
 - [Портал открытых данных.](#)



DataOps (BigData+Ops) – 2

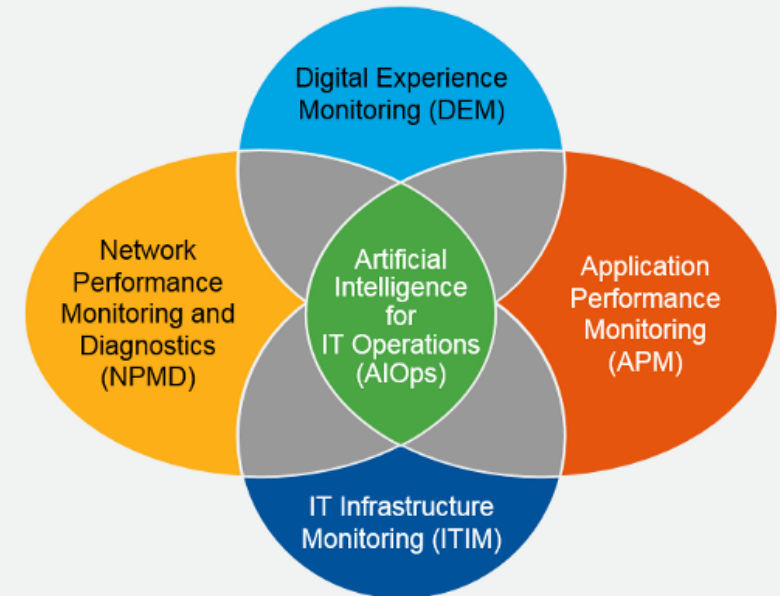
Критерий	DataOps-инженер	DevOps-инженер
сфера ответственности	Автоматизация и мониторинг управления данными в течение всего их жизненного цикла	Автоматизация и мониторинг разработки и развертывания программного обеспечения
направления деятельности	<ul style="list-style-type: none"> – инженерия данных – интеграция данных – повышение качества данных – обеспечение целостности данных и политики безопасного доступа к ним 	<ul style="list-style-type: none"> – разработка ПО – тестирование – развертывание и поддержка эксплуатации

прикладные задачи	<ul style="list-style-type: none"> – автоматизация процессов загрузки данных в хранилища – мониторинг операционных потоков данных – оптимизация аналитики данных – создание инфраструктуры для корректного хранения, движения и использования данных 	<ul style="list-style-type: none"> – разработка ПО в ускоренном режиме – частая поставка и развертывание ПО – быстрое переключение от задач разработки к эксплуатации через тестирование
взаимодействие с другими членами команды Big Data	<ul style="list-style-type: none"> – инженеры данных (Data Engineers) – исследователи данных (Data Scientists) – аналитики данных (Data Analysts) 	<ul style="list-style-type: none"> – программисты, разработчики ПО (Software Engineers, Developers) – системные администраторы и техническая поддержка (System Administrators, Operational Team) – тестировщики (Testers, QA-Engineers)

AIOps-1

- Искусственный интеллект для ИТ-эксплуатации (AIOps, Artificial Intelligence for IT Operations) включает в себя использование технологий искусственного интеллекта и машинного обучения наряду с большими данными, интеграцией данных и технологиями автоматизации, чтобы помочь сделать ИТ-эксплуатацию интеллектуальной и построенной на прогностических моделях. AIOps дополняет ручные операции решениями на основе ИИ и машинного обучения.
- Обычно AIOps-система состоит из трех основных подсистем:
 - Аналитическая подсистема использует инструменты искусственного интеллекта для сбора данных о состоянии ИТ-среды.
 - Подсистема машинного обучения применяет алгоритмы для анализа этих данных и автоматического создания прогнозов о том, как они изменятся в будущем.
 - Подсистема автоматизации использует существующие процессы, политики и шаблоны для автоматизации задач, которые часто выполняются вручную. Это можно сделать либо путем создания сценариев для выполнения людьми, либо путем их непосредственного выполнения без вмешательства человека.
- AIOps использует большие данные из различных источников, которые могут быть в любом формате, например:
 - Системные журналы и метрики.
 - Данные о событиях в реальном времени.
 - Состояние сети и данные о трафике.
 - Заявки и данные об инцидентах.
 - Накопленные знания о предыдущих инцидентах.

AIOps Will Provide Consolidated Analysis of Monitoring Data



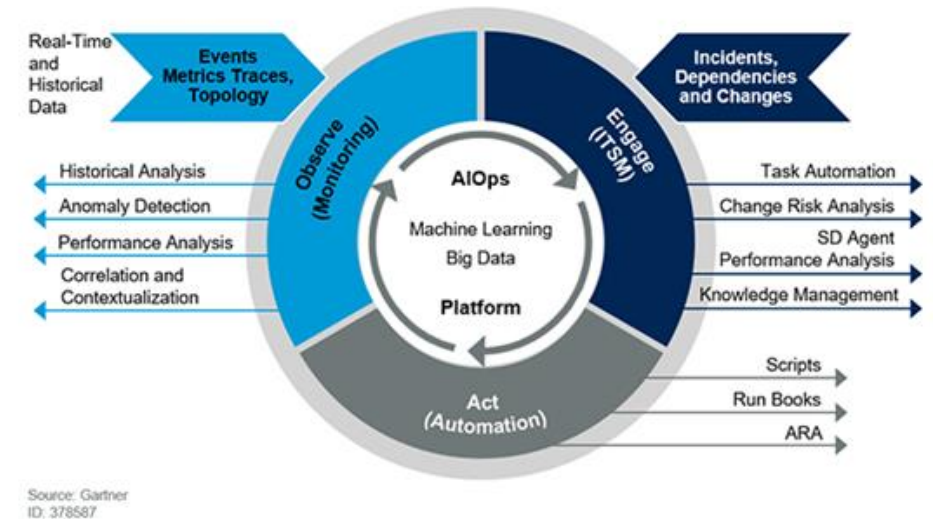
ID: 352799

© 2018 Gartner, Inc.

AIOps-2

- Обширные и разнообразные данные ИТ (обозначены черным и синим шевронами). AIOps основан на объединении различных данных как из управления ИТ-операциями (ITOM) (показатели, события и т. д.), так и из управления ИТ-услугами (ITSM) (инциденты, изменения и т. д.). Это явление называют «разрушением разрозненных хранилищ данных» - объединением данных из разрозненных источников, чтобы они могли «общаться» друг с другом и ускорить выявление первопричин или использовать автоматизацию.
- Extensive and diverse IT data. В основе платформы, в центре рисунка, лежат большие данные. Поскольку данные выделяются из разрозненных инструментов, их необходимо объединить для поддержки аналитики следующего уровня. Это должно происходить не только в автономном режиме, но и в реальном времени по мере поступления данных.
- Machine learning. Большие данные позволяют применять машинное обучение для анализа огромных объемов разнообразных данных. Это невозможно ни до объединения данных, ни вручную. Машинное обучение автоматизирует существующую ручную аналитику и позволяет использовать новую аналитику для новых данных - и все в таком масштабе и скорости, которые недоступны без AIOps.
- Observe. Это эволюция традиционного домена ITOM, который объединяет данные о разработке (трассировки) и другие данные, не относящиеся к ITOM (топология, бизнес-метрики), чтобы обеспечить новые способы корреляции и формирования контекста. В сочетании с обработкой в реальном времени идентификация вероятной причины становится одновременной с генерацией проблемы.
- Engage. Развитие традиционного домена ITSM включает двунаправленную связь с данными ITOM для поддержки вышеуказанного анализа и автоматического создания документации для аудита и соответствия / нормативных требований. AI / ML выражается здесь в когнитивной классификации плюс маршрутизации и интеллекте в точке взаимодействия с пользователем, например, чат-ботами.
- Act. Это последнее звено цепочки создания ценности AIOps. Автоматизация анализа, рабочего процесса и документации будет напрасной, если ответственность за действия вернется в руки людей. Act включает в себя кодификацию человеческих знаний в области автоматизации и согласования исправлений и реагирования.

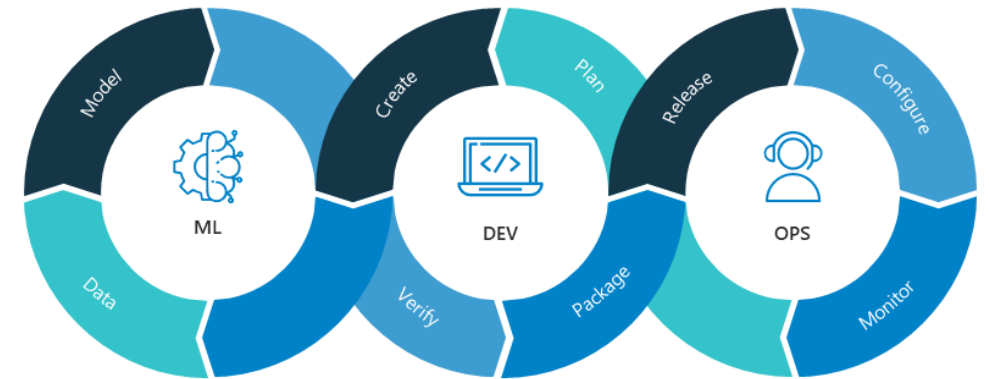
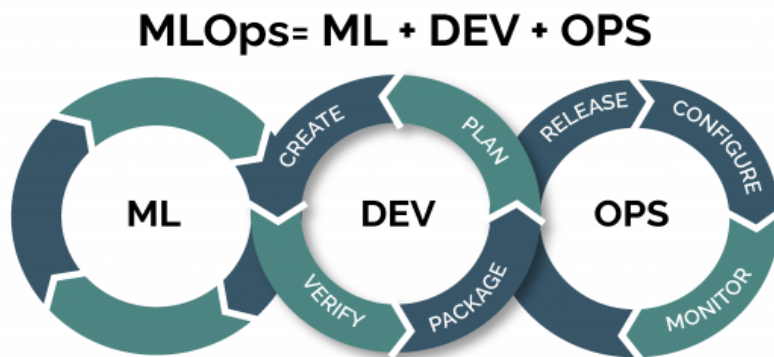
AIOps Platform Enabling Continuous Insights Across IT Operations Monitoring (ITOM)



[Детальный пример из статьи «Что такое AiOps, как это работает?»](#)

MLOps-1

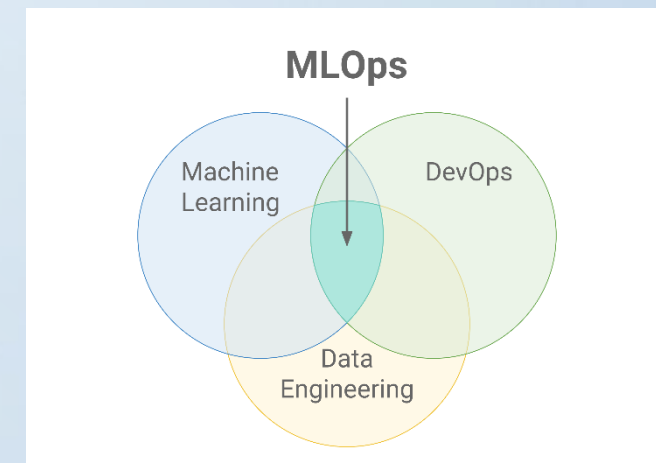
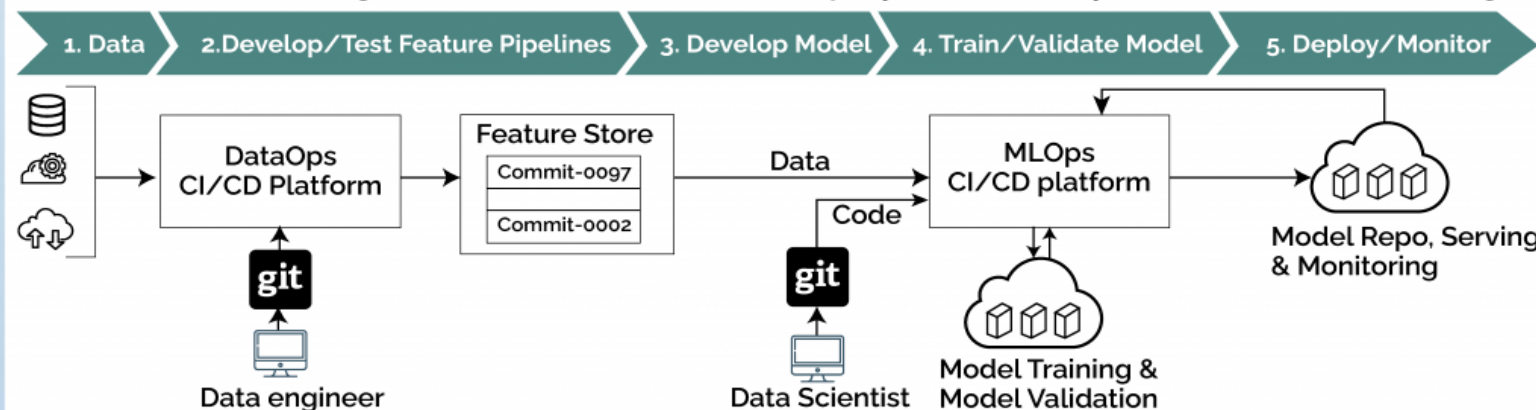
- «MLOps – это набор практик (и инструментов), направленных на надежное и эффективное внедрение и поддержку моделей машинного обучения в производстве». Википедия.



Experiment
Data Acquisition
Business Understanding
Initial Modeling

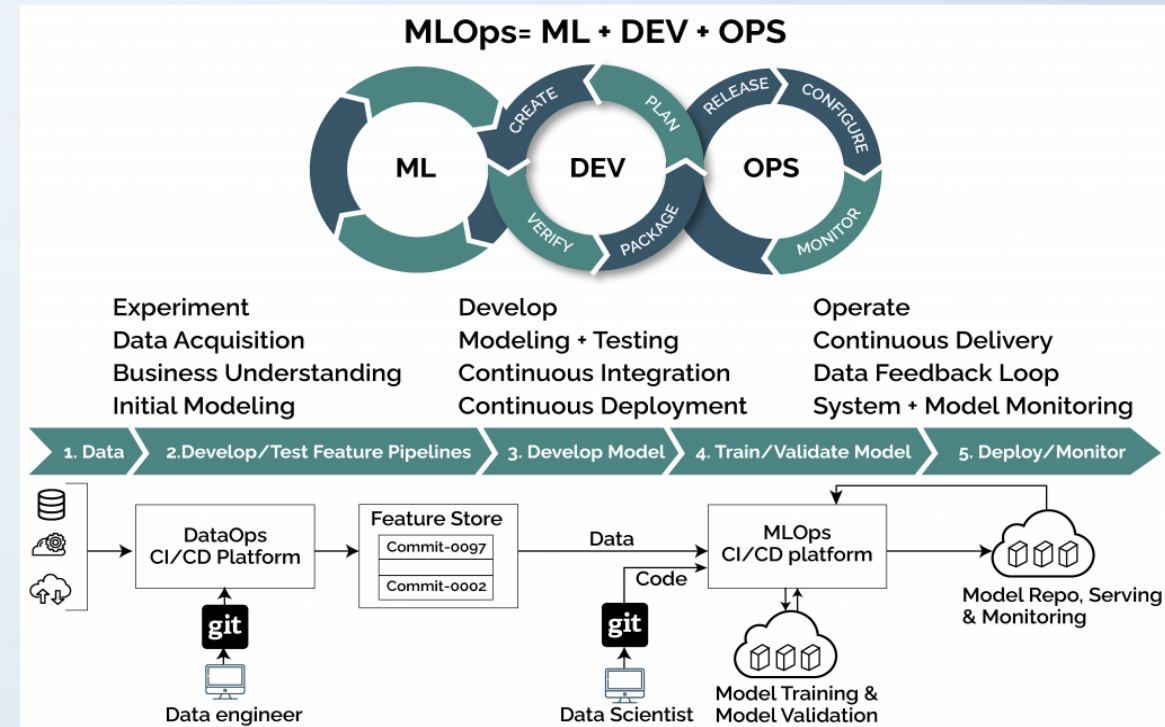
Develop
Modeling + Testing
Continuous Integration
Continuous Deployment

Operate
Continuous Delivery
Data Feedback Loop
System + Model Monitoring

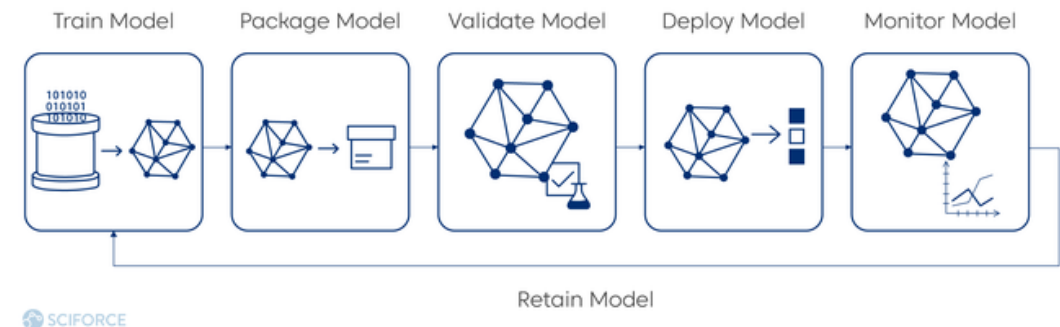


MLOps-2

- Каждый проект машинного обучения направлен на построение статистической модели из данных с применением алгоритма машинного обучения. Следовательно, данные и модель машинного обучения представляют собой два разных артефакта для разработки программного обеспечения в части разработки кода. В целом жизненный цикл машинного обучения состоит из трех элементов:
- Инженерия данных:** предоставление и обучение наборов данных для алгоритмов машинного обучения. Включает в себя прием данных, исследование, проверку, очистку, маркировку и разделение (на набор данных для обучения, проверки и тестирования).
- Проектирование модели:** подготовка окончательной модели. Включает в себя обучение модели, оценку, тестирование и упаковку.
- Развертывание модели:** интеграция обученной модели в бизнес-приложение. Включает обслуживание модели, мониторинг производительности и ведение журнала производительности.



ML Lifecycle, according to Microsoft, looks like:



MLOps-3

- Для MLOps характерно следующее понимание CI/CD:
 - **Непрерывная интеграция (CI)** применяется к тестированию и проверке данных, схем и моделей, а не только к коду и компонентам.
 - **Непрерывное развертывание (CD)** относится ко всей системе, которая предназначена для развертывания другой услуги, предоставляемой машинным обучением, но не к отдельному программному обеспечению или услуге.
 - **Непрерывное обучение (CT)** уникально для моделей машинного обучения и означает обслуживание и переподготовку моделей.
- Существует три уровня MLOps, разделенных на категории и основанных на уровне автоматизации процессов:
 - **Уровень 0 MLOps:** процесс создания и развертывания модели ML выполняется полностью вручную. Этого достаточно для моделей, которые редко меняются или обучаются.
 - **Уровень 1 MLOps:** непрерывное обучение модели путем автоматизации конвейера машинного обучения, хорошо подходит для моделей, основанных на новых данных, но не для новых идей машинного обучения.
 - **Уровень 2 MLOps:** автоматизация CI/CD позволяет работать с новыми идеями проектирования функций, архитектуры модели и гиперпараметров.
- MLOps нуждается в постоянном мониторинге и проверке точности:
 - Мониторинг памяти: мониторинг использования памяти при построении прогнозов.
 - Мониторинг производительности модели: когда необходимо производить переобучение моделей? Данные могут изменяться, и это может повлиять на результаты прогнозов.
 - Мониторинг инфраструктуры: постоянный сбор и анализ соответствующих данных об инфраструктуре.

Различие между MLOps и DataOps

- На предприятии может быть DataOps без MLOps, потому что возможно извлекать и преобразовывать данные без машинного обучения. Обратное едва ли верно.
- DataOps применим на протяжении всего жизненного цикла приложений данных. MLOps в первую очередь предназначен для упрощения управления и развертывания моделей машинного обучения.
- Целью DataOps является оптимизация циклов управления данными, ускорение выхода на рынок и получение высококачественных результатов. Цель MLOps – облегчение развертывание моделей машинного обучения в производственных средах.



Жизненный цикл модели-1

- ЖЦМ или жизненный цикл модели – это совокупность всех этапов существования ML-модели: с момента осознания ее необходимости до вывода из эксплуатации.
- Можно выделить следующие стадии ЖЦМ:
 - Постановка задачи DS/ML: сведение и формализация решаемой бизнес-задачи к задаче построения модели машинного обучения или применения методов продвинутой аналитики;
 - Подготовка данных: проектирование переменных и разработка витрины данных для обучения моделей ML и применения методов Data Science;
 - Обучение модели: подбор архитектуры и параметров модели решающей поставленную задачу;
 - Внедрение модели: постановка модели в промышленную эксплуатацию (как говорится, в PROD);
 - Эксплуатация и мониторинг: применение модели для принятия решений в рассматриваемой бизнес-задачи и контроль качества этих решений;
 - Изменение или вывод: пересмотр модели в связи с изменением условий ее применения или вывод её из эксплуатации.

1. Отсутствие инфраструктуры ЖЦМ



Жизненный цикл модели-2

- При этом подходе все еще есть вероятность упустить какие-либо события, происходящие с моделью. Отдельно взятые ML/MLOps-сервисы плохо синхронизированы и согласованы между собой. Процесс перевода модели из одного сервиса в другой долгий, сложный и ведется вручную.
- Например, специалисты Data Science обучают модель и сохраняют артефакты в mlflow так, как им удобно, а разработчики бэкэнда осуществляют перенос модели в production в каком-то своем формате, к примеру, не поддерживающем работу композитной модели, то есть использующей результаты других подмоделей как признаки при обучении. В итоге им нужно каждый раз договариваться заново, а ведь каждая команда продолжает развитие в рамках своих технологий, и там постоянно что-то меняется.

2. Управление отдельными этапами ЖЦМ



Жизненный цикл модели-3(1)

- **Model Registry или библиотека моделей**

- Реестр моделей необходим для учета каждого разработанного алгоритма. Из реестра мы узнаем о хронологии появления моделей, версию модели, этап, на котором она сейчас находится. В реестре хранятся ссылки на всю связанную информацию и артефакты, такие как документация, датасеты, другие модели, и в целом все метаданные, например, модели могут иметь теги для фильтрации, поиска, объединения похожих алгоритмов в одну категорию.

- **Управление бизнес-процессами ЖЦМ**

- Верхнеуровневые этапы пайплайна можно разделить на детальные бизнес-процессы, которыми можно управлять из системы в BPMN-подобной нотации. Например, бизнес-процесс разработки модели описывает что и в каком порядке должны сделать участники команды разработки модели, автоматизирует процесс передачи релевантных данных внутри команды, позволяет зафиксировать по шагам все принятые решения, а бизнес-процесс эксплуатации модели описывает что делать, если качество модели сильно ухудшилось.

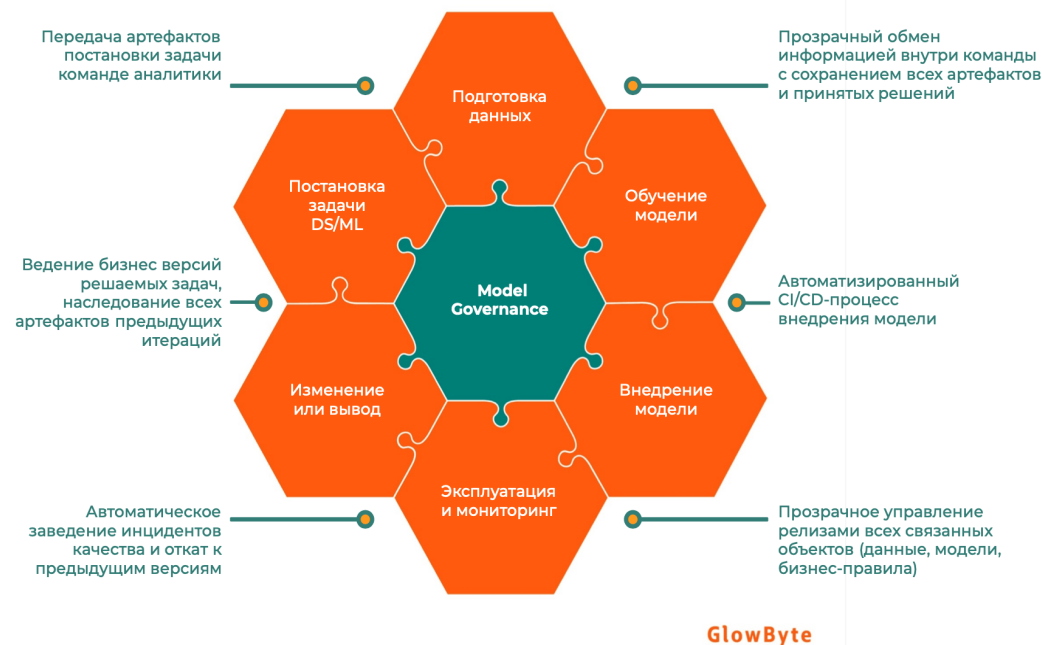
- **Прозрачная ролевая система**

- Чтобы сделать ЖЦМ более прозрачным, нужна четкая ролевая система, с помощью которой команда будет знать, кто за что отвечает. При таком подходе Teamlead может назначать ответственных за ту или иную задачу, отслеживать статус выполнения задач и обращаться к нужному сотруднику за отчетом. Каждый член команды будет знать о своих задачах, которые могут быть собраны в одном месте.
- Например, при переходе на этап подготовки данных для модели эта задача назначается сотруднику из команды с ролью Data Engineer. Сотрудник в свою очередь получает уведомление о новом назначении, он увидит новую задачу в списке задач, там же у него будет возможность заполнить необходимые артефакты и перевести задачу в статус «in progress/done».
- В больших командах и организациях становится важным вопрос разграничения прав доступа. Не каждый участник команды должен видеть информацию чувствительную для бизнеса и не каждый участник команды должен иметь право на внесение изменений в определенные артефакты и атрибуты модели.

- **Хранилище артефактов моделей**

- В процессе разработки моделей появляется большое количество артефактов. ЖЦМ решает задачу организации единого места хранения разнородных артефактов — от бизнес-решений (согласования документации, прохождение code review, принятия решений о внедрении и т.д.) и заканчивая техническими артефактами (pickle-файл модели, sql-скрипт сборки витрины), которые обычно хранятся в специализированных системах (Jira, git, DVC, простые файловые хранилища, локальные папки разработчиков).
- Единый интерфейс для артефактов позволяет получить информацию о модели для любой роли-участника разработки в одном окне.
- Наличие хранилища также позволяет разграничивать права доступа.

3. Решение для ЖЦМ класса Model Governance



Жизненный цикл модели-3(2)

- **Интеграция со сторонними сервисами**

- Хорошим решением для системы управления ЖЦМ станет разработка синхронизационного модуля, который будет отвечать за интеграцию с Jira, Wiki, Git, почтой, календарем и другими сервисами. Например, автоматическое создание и заполнение страницы на Wiki метаданной о модели или автоматическое заведение задач в Jira и дедлайнов в календаре. При этом синхронизация может работать и в обратную сторону, то есть обновления в специализированных системах также могут интегрироваться с мастер-системой ЖЦМ.

- **Интеграция с MLOps-решениями**

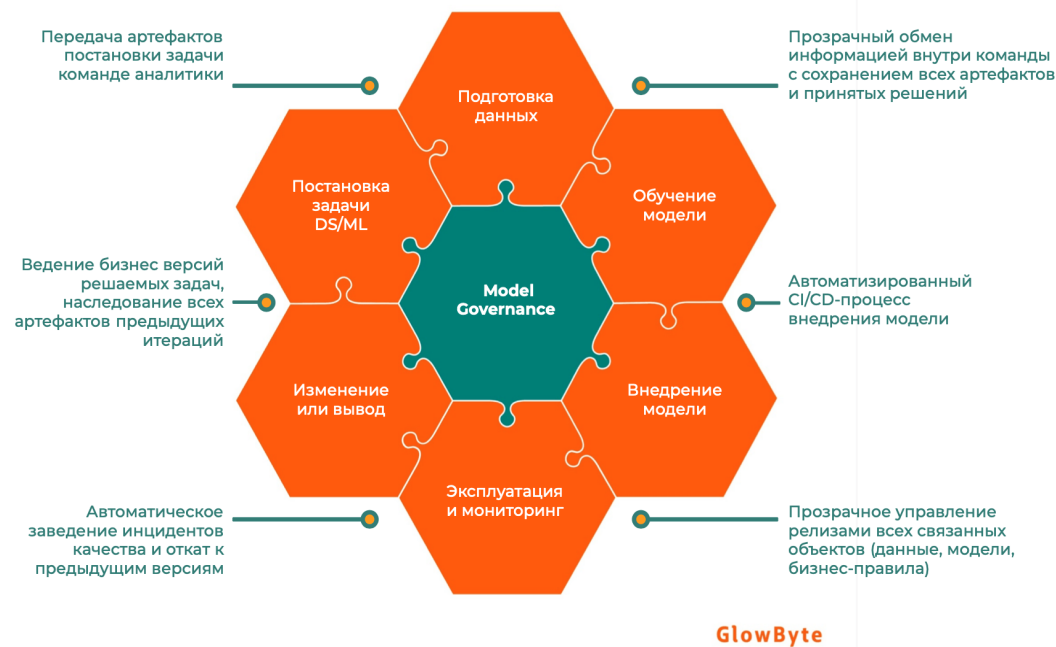
- Это сделает процесс работы с ЖЦМ проще и удобнее. Например, запуск процессов CI/CD по кнопке, отображение экспериментов из mlflow, отображение дашбордов мониторинга.

- **Мониторинг качества бизнес-метрик**

- Важно отслеживать не только статистическое качество самой модели, но и бизнес-метрики применения конкретной модели в конкретных бизнес-задачах. Дашборды разработанного сервиса показывают успешность применения каждой модели на основных бизнес-показателях.
- С помощью такой системы легко отслеживать каждый этап работы над моделью, понимать, на какой стадии она находится и кто в настоящий момент за нее отвечает. Помимо этого, появляется возможность считать время от постановки задачи до вывода в production, находить самые длительные места и оптимизировать их.

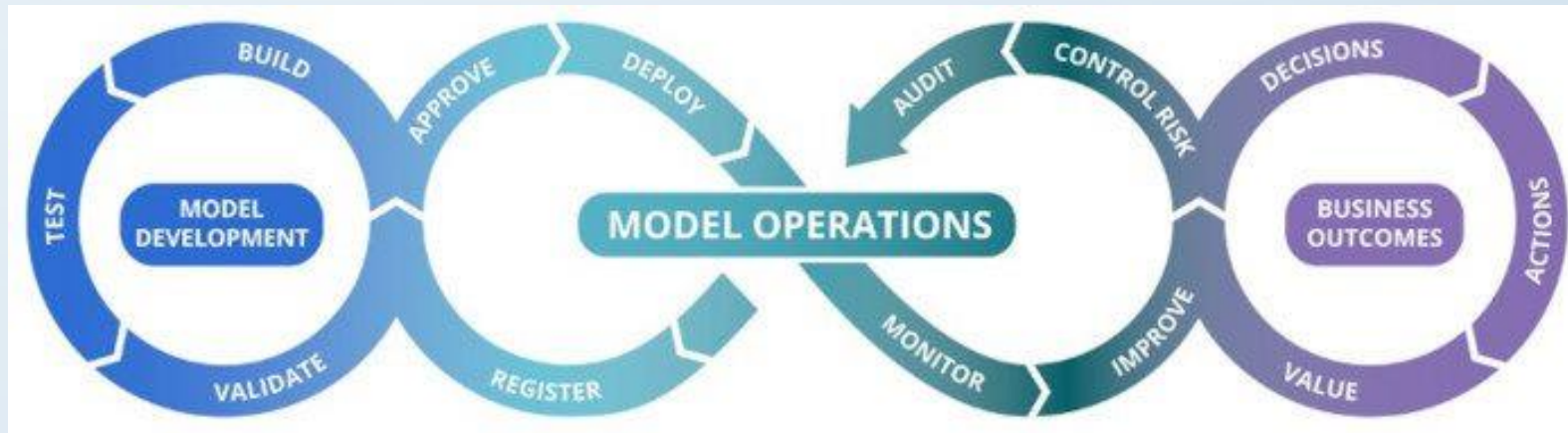
- Также описанная система ЖЦМ позволяет существенно упростить работу и передачу информации внутри команд, например, новым сотрудникам легко погружаться в проект, имея доступ ко всем артефактам модели, а про работающую в PROD модель никогда не возникнет вопросов кто и зачем ее сделал.
- Система станет не только единой точкой входа во все ML/MLOps-сервисы, но и удобным инструментом для всех ролей от разработчиков до менеджеров.

3. Решение для ЖЦМ класса Model Governance



ModelOps

- ModelOps – это система операций с ML-моделями, ориентированная на руководство и управление жизненным циклом широкого спектра операционализированных моделей искусственного интеллекта (ИИ) и методов принятия решений, включая машинное обучение, графы знаний, правила, оптимизацию, лингвистические и агентные модели.
- ModelOps лежит в основе любой корпоративной стратегии искусственного интеллекта, управляя жизненными циклами всех моделей в производстве на всем предприятии, от запуска в production до оценки и обновления приложения согласно набору управляющих правил. Это позволяет бизнес-экспертам самостоятельно оценивать работу ИИ-модели с экономической точки зрения, независимо от специалистов по обработке и анализу данных.
- Цель ModelOps в устранении разрыва между развертыванием модели и ее управлением в производственной среде с учетом бизнес-показателей, технических ограничений и рисков. Технически это реализуется с помощью повторно используемых компонентов, которые позволят версиям модели соответствовать бизнес-приложениям и включают MLOps-практики, как мониторинг модели, обнаружение дрейфа и активное обучение.
- ModelOps – это расширение MLOps для масштабируемого и управляемого объединения нескольких объектов, решений и платформ ИИ. ModelOps требует тех же специалистов, что и MLOps, а также компетенций, связанных с ИТ-операциями, управлением рисками и общим бизнес-менеджментом. Таким образом, в отличие от MLOps, ModelOps фокусируется не только на моделях машинного обучения, а нацелено на операционализацию всех ИИ-решений и используется ИТ-командой и бизнес-стейкхолдерами. ModelOps фокусируется на управлении моделями и комплексном управлении жизненным циклом, где нужно убедиться, что прогнозируемая ценность для бизнеса, операционная эффективность, а также уровни рисков соответствуют требованиям.



Инструменты MLOps

- [Awesome MLOps](#)
- [The Best Open-Source MLOps Tools You Should Know](#)
- [Top 10 Open Source MLOps Tools](#)
- <https://habr.com/ru/company/vk/blog/694482/>

Cookiecutter

- Шаблонизатор для создания проектов.
- Официальный сайт - <https://github.com/cookiecutter/cookiecutter>
- Создание проекта машинного обучения - <https://drivendata.github.io/cookiecutter-data-science/>

Snakemake

- Система сборки Data Science – проектов.
- Официальный сайт - <https://snakemake.github.io/>
- Статья с описанием продукта - <https://f1000research.com/articles/10-33/v1>

DVC - Data Version Control

- Статьи с пояснением принципов работы:
 - <https://habr.com/ru/company/raiffeisenbank/blog/461803/>
 - <https://habr.com/ru/post/535274/>
- Официальный сайт - <https://dvc.org/doc>
- Ключевые разработчики из ods.ai – <https://ods.ai/projects/dvc>
- Компания занимается разработкой инфраструктуры DS-проектов – <https://iterative.ai/>

Apache AirFlow

- Статьи с пояснением принципов работы:
 - https://ru.wikipedia.org/wiki/Apache_Airflow
 - <https://habr.com/ru/company/vk/blog/339392/>
- Официальный сайт - <https://airflow.apache.org/>

MLflow

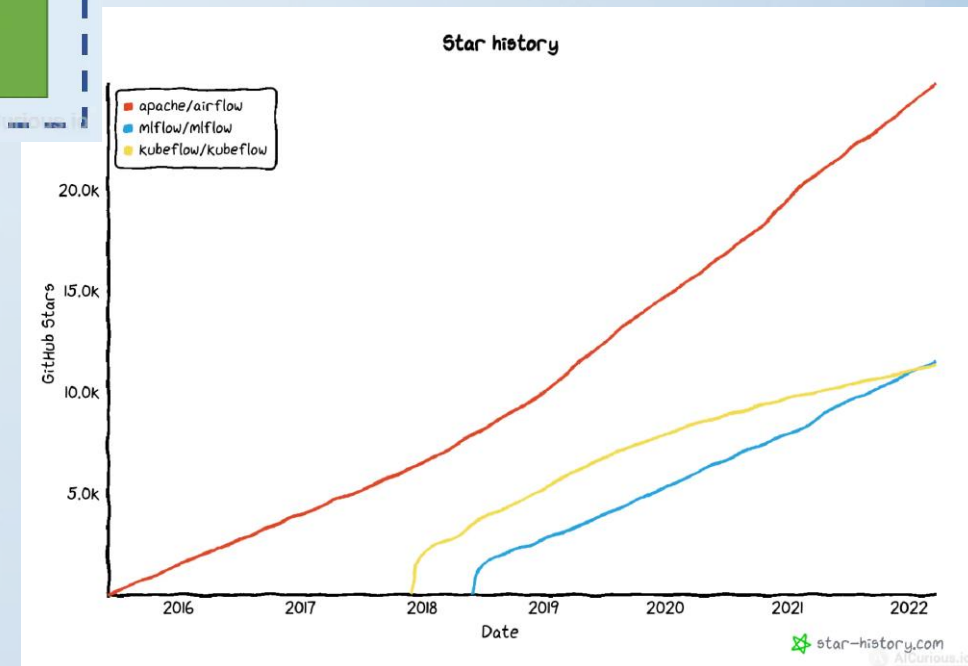
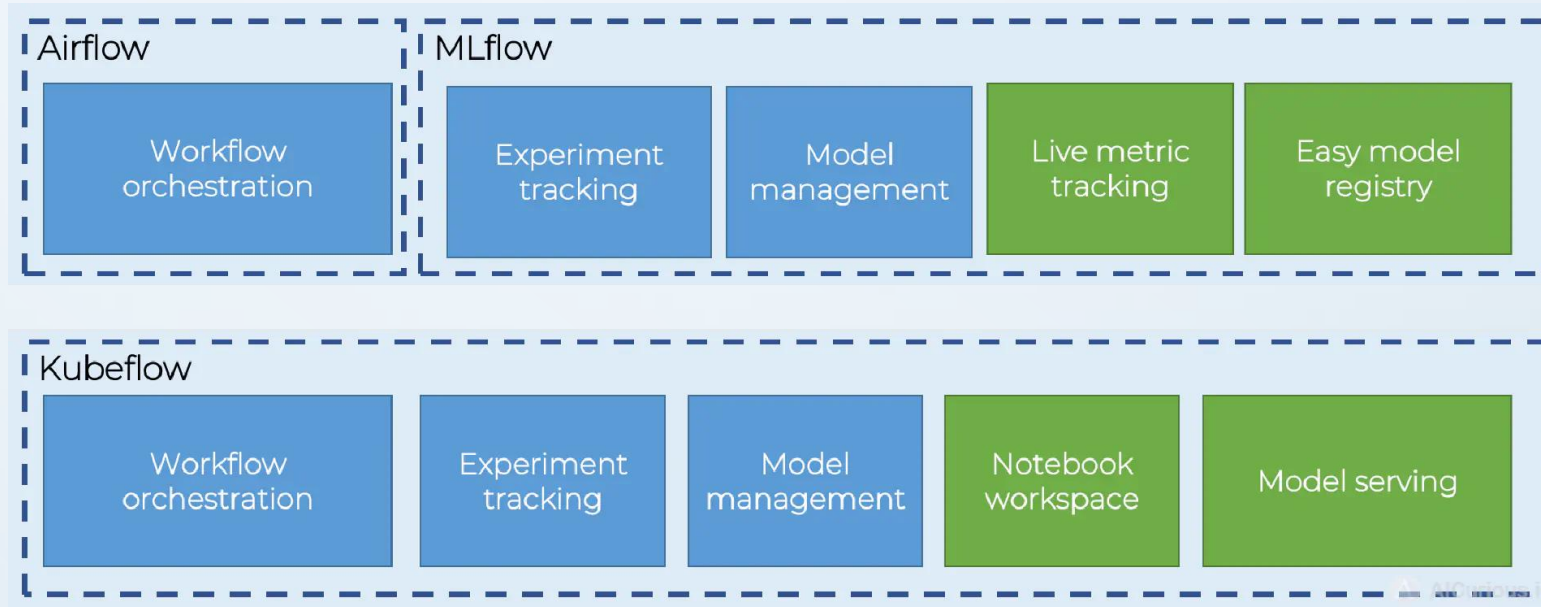
- Статьи с пояснением принципов работы:
 - <https://habr.com/ru/company/X5Tech/blog/593263/>
 - <https://habr.com/ru/company/vk/blog/565022/>
- Официальный сайт - <https://mlflow.org/>

Kubeflow

- Статьи с пояснением принципов работы:
 - <https://habr.com/ru/company/vk/blog/547066/>
 - <https://habr.com/ru/company/nixys/blog/578880/>
- Официальный сайт - <https://www.kubeflow.org/>

Сравнение продуктов

- Статья с пояснениями - <https://aicurious.io/posts/airflow-mflow-or-kubeflow-for-mlops/>



Источники

- [Курс «MLOps и production подход к ML исследованиям» \(ods.ai\)](https://ods.ai)
- Статьи:
 - [Роль TechOps, DevOps, and NoOps в жизненном цикле программного обеспечения](#)
 - [Что такое NoOps и зачем тут облака](#)
 - [NoOps: Новый Agile в облаках](#)
 - [Как и зачем изучать методики DevSecOps в 2021 году?](#)
 - [Что такое AIOps или искусственный интеллект для ИТ-эксплуатации? Топ 10 случаев использования AIOps](#)
 - [Что такое AIOps, как это работает?](#)
 - [Манифест DataOps](#)
 - [5 столпов MLOps](#)
 - [Как и зачем управлять ML-моделями?](#)
 - [Познакомьтесь с ModelOps: новый расширенный MLOps для бизнеса](#)
 - [DataOps : The new DevOps for Analytics](#)
 - [MLOps and ModelOps: What's the Difference and Why it Matters](#)
 - [Awesome MLOps](#)