

DevOps 2021:

Вводные данные.

Определение DevOps.

Жизненный цикл программных продуктов.

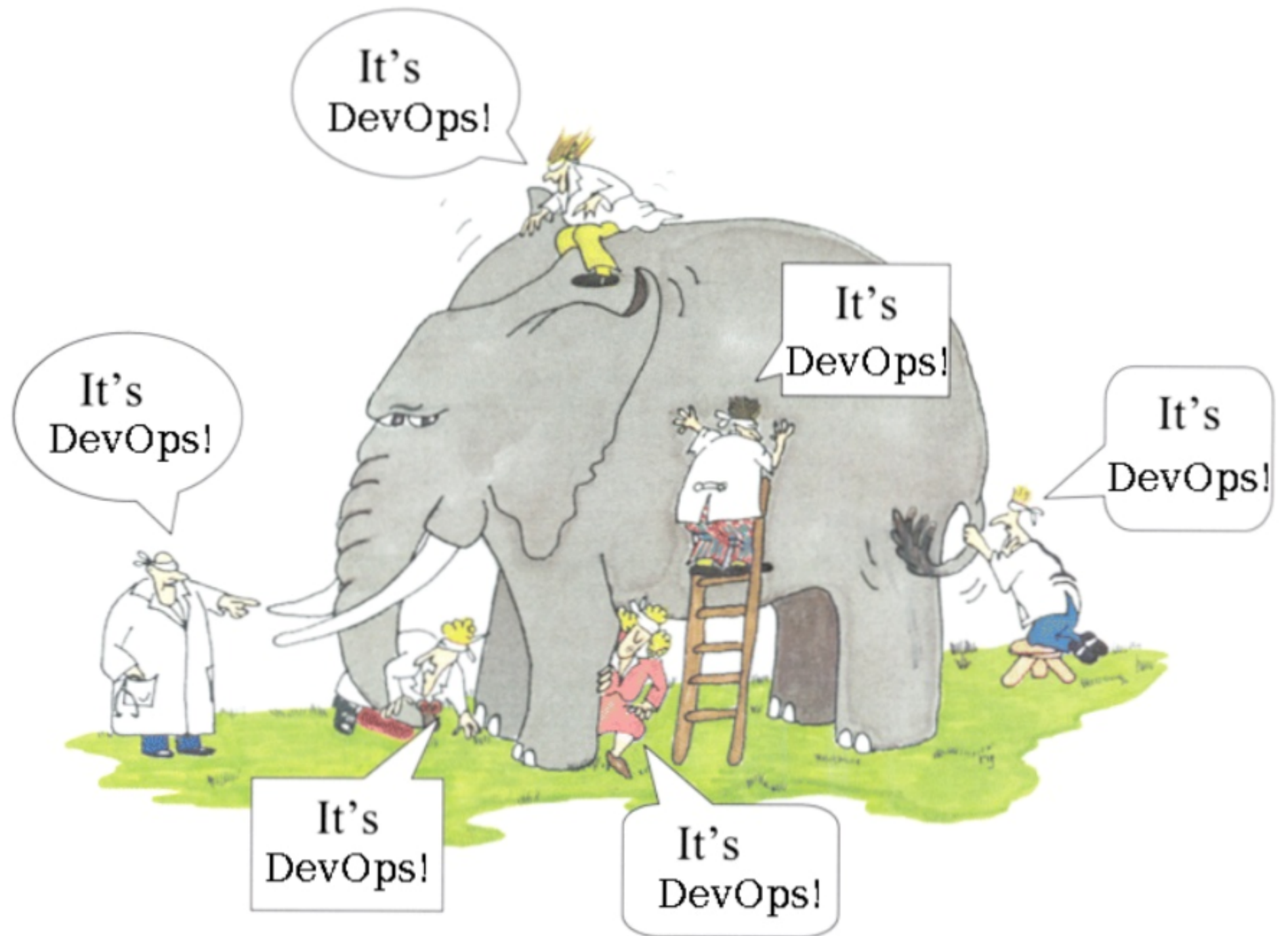
DevOps в цикле разработки.

Чего мы ждём от участников курса:

- Участия xD
- Ответственного отношения к заданиям
- Культурной коммуникации
- Вопросов к преподавателям
- Обратной связи по качеству курса/материалов

Что мы предлагаем:

- Обзор ключевых компетенций DevOps-специалистов
- Коварные домашние задания
- Два месяца доступа к знаниям преподавателей-экспертов
- Social networking для начинающих специалистов
- Приглашения на работу в Andersen лучшим курсантам



DevOps (Development + Operations)

- DevOps - набор практик, нацеленных на активное взаимодействие специалистов по разработке со специалистами по информационно-технологическому обслуживанию и взаимную интеграцию их рабочих процессов друг в друга. Базируется на идее о тесной взаимозависимости разработки и эксплуатации программного обеспечения и нацелен на то, чтобы помогать организациям быстрее создавать и обновлять программные продукты и услуги. (Википедия)
- DevOps - это набор методик, с помощью которых можно автоматизировать процессы между командами разработчиков и ИТ-специалистов, чтобы они могли быстрее и надежнее собирать, тестировать и выпускать релизы программного обеспечения. (Atlassian)
- DevOps – это сочетание культурных принципов, подходов и средств, которое повышает способность компаний создавать приложения и сервисы на высокой скорости. С DevOps разработка и оптимизация продуктов происходит быстрее, чем при использовании традиционных процессов разработки программного обеспечения и управления инфраструктурой. Благодаря такой скорости компании могут повысить уровень обслуживания клиентов и более эффективно конкурировать на рынке. (AWS)

« DevOps отличается от обычного сисадмина тем, что не может сказать "это задача программистов" »

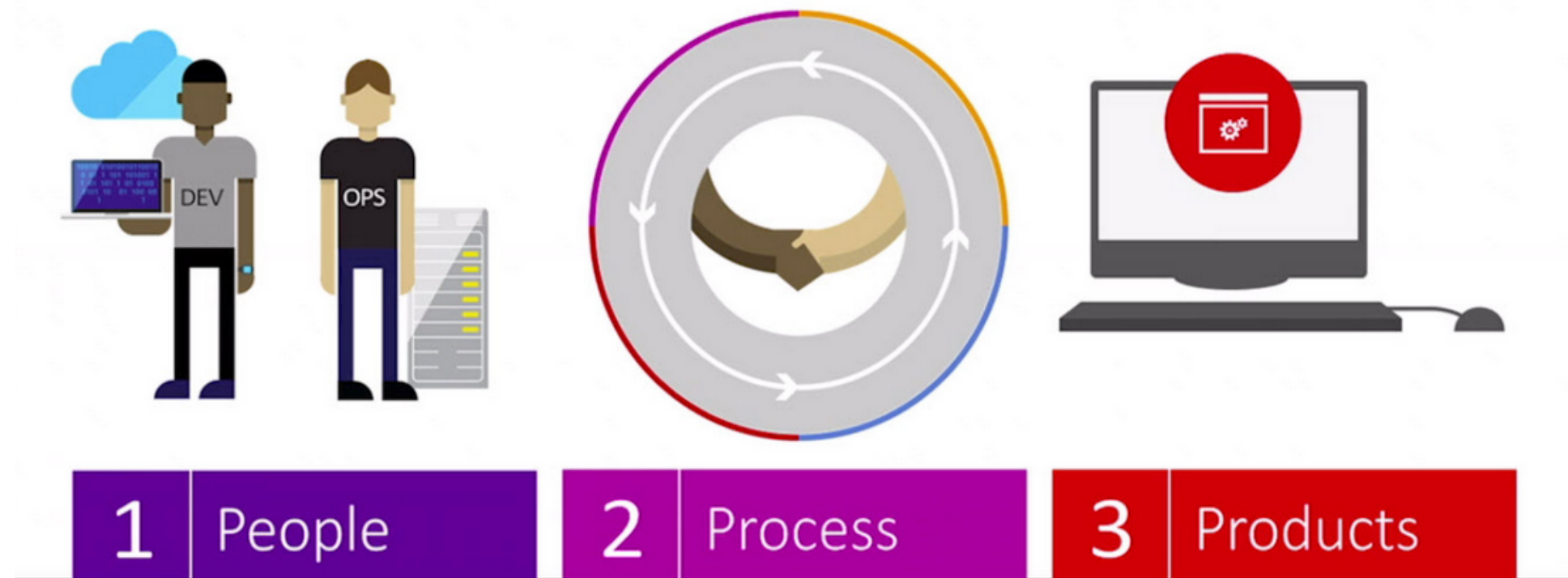
DevOps инженер не занимается:

























- Организационными мероприятиями, перестройкой структуры разработки компании.
- Программированием.
- Тестированием ПО.
- Поиском багов, багфиксингом.
- Разработкой и написанием технической документации.
- Разработкой архитектуры проектов.

DevOps инженер занимается:

- Админской работой.
- Автоматизацией.
- Выпуском релизов.
- Мониторингом.
- Созданием и поддержкой окружений (тестовых, боевых, сборочных и т.п.).

DevOps: the three stage conversation



Configuration Management	Continuous integration	Microservices	Collaboration	Monitoring	Development
					
					
					
					

Терминология

- Релиз (Release) - выпуск готового продукта или готовый продукт.
- Релиз кандидат (RC) - практически готовый продукт. Завершающий этап тестирования.
- Бета версия (Beta) - версия продукта, предназначенная для тестирования.
- Альфа версия (Alpha) - ранняя стадия готовности продукта. Тестирование.
- Разработчик (Developer) - программист.
- QA (Tester) - тестировщик.
- Тимлид (Team Leader) - глава команды разработки.
- Product owner - отвечает за разработку продукта.
- Ветка (Branch) - термин из СКВ.
- Запустить (Push) - термин из СКВ.
- Коммит (Commit) - применить изменения. Термин из СКВ.

Терминология

- Продакшн, прод (Production) - боевое окружение.
- Стэйдж (Stage) - одно из промежуточных окружений, например тестовое.
- Сборка (Build) - подготовленный для использования продукт, обычно с п. номером.
- Задеплоить, залить, выложить (Deployment) - установить сборку на стэйдж или прод.
- Баг (Bug) - ошибка в ПО.
- Фича (Feature) - возможность, особенность продукта
- Код ревью (Code review) - инспекция кода на наличие ошибок.
- Компилировать, билдить (Compile) - собирать продукт из исходного кода.

Типовой проект включает в себя следующие этапы разработки программного обеспечения:

- Анализ требований к проекту.
- Проектирование.
- Реализация.
- Тестирование продукта.
- Внедрение и поддержка.

Традиционно используют следующие методологии разработки ПО.

- Waterfall — классический подход.
- RUP (Rational Unified Process) — рациональный подход.
- Agile — общая методология гибкой разработки.
- Spiral — спиральный метод.
- Incremental model — инкрементальная модель разработки.
- JAD (Joint Application Development) — подход, ориентированный на пользователя.
- RAD (Rapid Application Development) — модель быстрой разработки.
- V-модель.

Waterfall. Каскадная модель или «водопад»



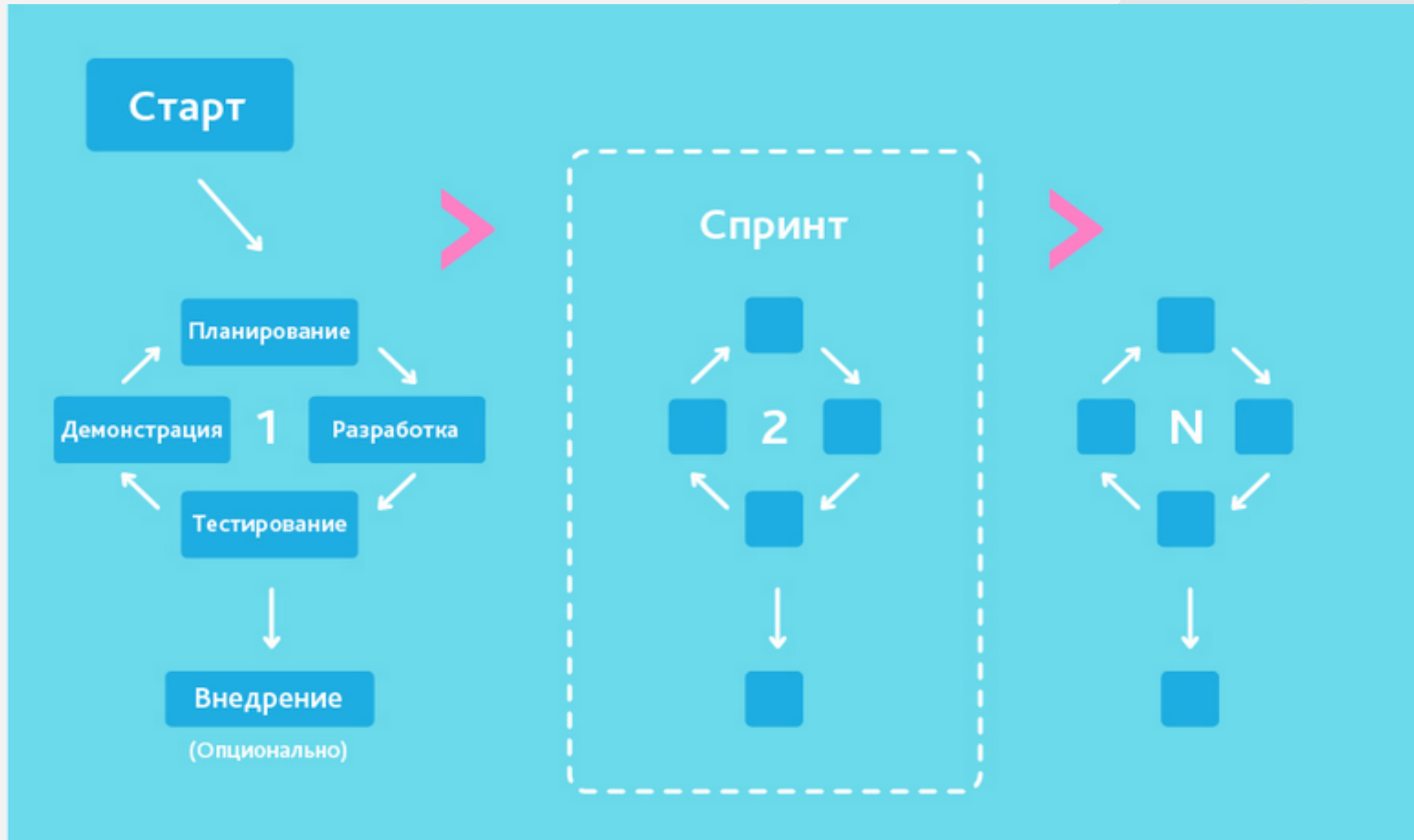
Incremental model



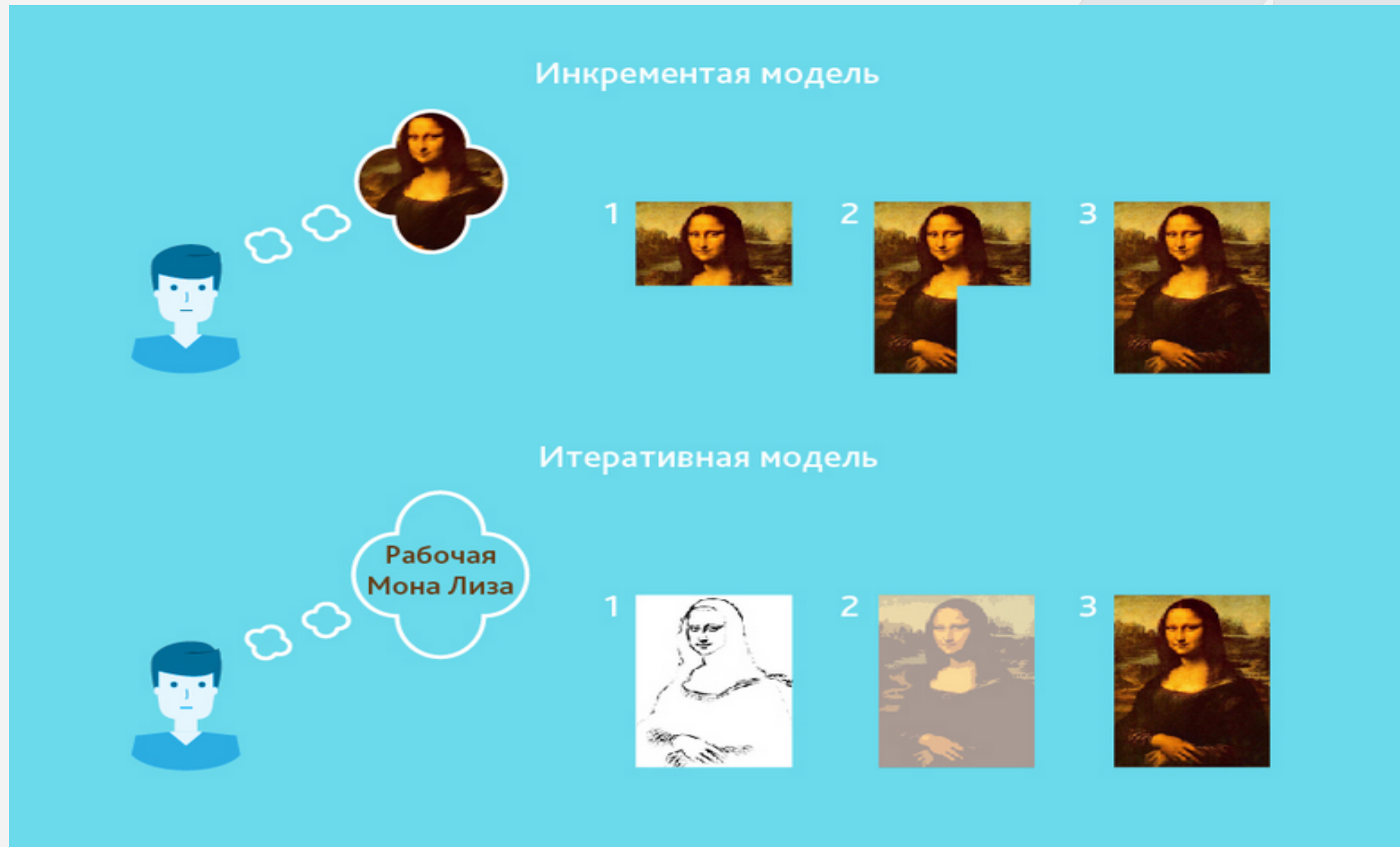
RAD model

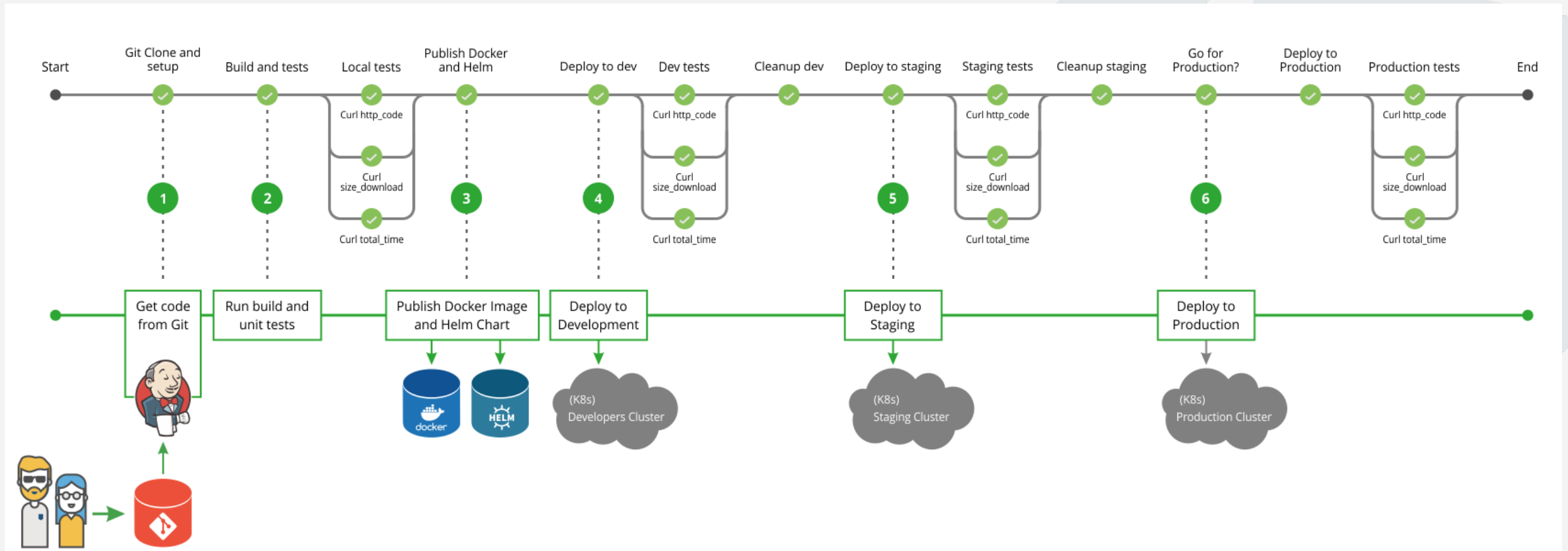


Agile model



Iterative Model





Домашнее задание:

- Завести репозиторий на github.com
 - один репозиторий
 - директория под каждое домашнее задание
 - README.md в каждой директории!
- README.md driven development:
 - создать Readme.md с markdown разметкой
 - написать кратко о себе
 - указать 3 пункта, почему вам интересен курс
 - указать 3 пункта, чего вы сами ожидаете от курса
 - commit-messages & readmes in English :-)
- Устроим блог из репозитория, ежедневное задание: TIL
 - *TIL == Today I've Learned