УНИВЕРСИТЕТ ЛОБАЧЕВСКОГО

История курса по параллельному программированию: от ручного тестирования к автоматизации

Нестеров Александр,
ИИТММ, каф. МОСТ

Курс по параллельному программированию

Осенний семестр

MPI

Весенний семестр

OpenMP
TBB
std::thread

Задача: сохранить качество проверки работ при увеличении числа студентов

Причины ввода автоматизации

- Увеличение количества студентов
- Ухудшение качества проверки работ
    - Для параллельного программирование особая проблема
- Нужна система, не зависящая от преподавателя
    - Требование круглосуточной автономности
- Программа должна являться гарантией работоспособности

Начальная идея жизнеспособности проекта:

- Промышленный проект, библиотеки, программа считаются живыми пока есть в них нужда и имеет смысл продолжать разработку.
    - Также: идеи берутся из задач бизнеса, динамическое развитие проекта

Начальная идея жизнеспособности проекта:

- Промышленный проект, библиотеки, программа считаются живыми пока есть в них нужда и имеет смысл продолжать разработку.
    - Также : идеи берутся из задач бизнеса, динамическое развитие
- Студенческий проект, библиотека, программа живы, пока студентам это нужно.
    - Также : идеи дает научный руководитель, сданный диплом, заморозка проекта, стартап

Начальная идея жизнеспособности проекта:

- Промышленный проект, библиотека, программа считаются живыми пока есть в них нужда и имеет смысл продолжать разработку.
    - Также : идеи берутся из задач бизнеса, динамическое развитие
- Студенческий проект, библиотека, программа живы, пока студентам это нужно.
    - Также : идеи дает научный руководитель, сданный диплом, заморозка проекта, стартап
- Откуда берутся идеи для учебного проекта и как сделать его динамически развивающимся?

Начальная идея жизнеспособности проекта:

- Промышленный проект, библиотека, программа считаются живыми пока есть в них нужда и имеет смысл продолжать разработку.
    - Также : идеи берутся из задач бизнеса, динамическое развитие
- Студенческий проект, библиотека, программа живы, пока студентам это нужно.
    - Также : идеи дает научный руководитель, сданный диплом, заморозка проекта, стартап
- Откуда берутся идеи для учебного проекта и как сделать его динамически развивающимся?

Основа получения идей и изменения проекта:

1) Попытки обмануть систему

2) Простота проекта для простого порога вхождения (для всех)

Начальная база проекта

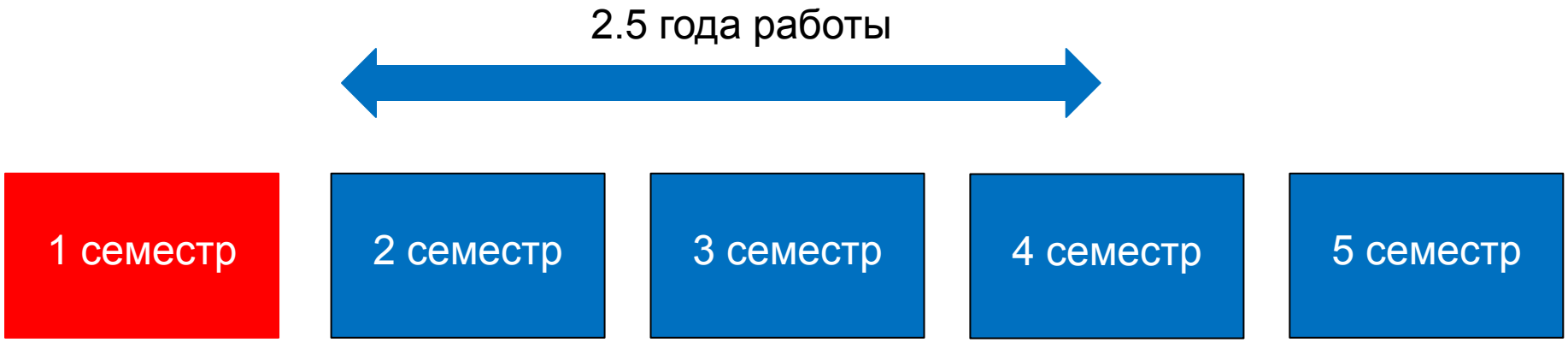
- Непрерывная интеграция (CI)
- Кроссплатформенная разработка
- Документирование и отчетность

Введение

УНИВЕРСИТЕТ ЛОБАЧЕВСКОГО
НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

2.5 года работы

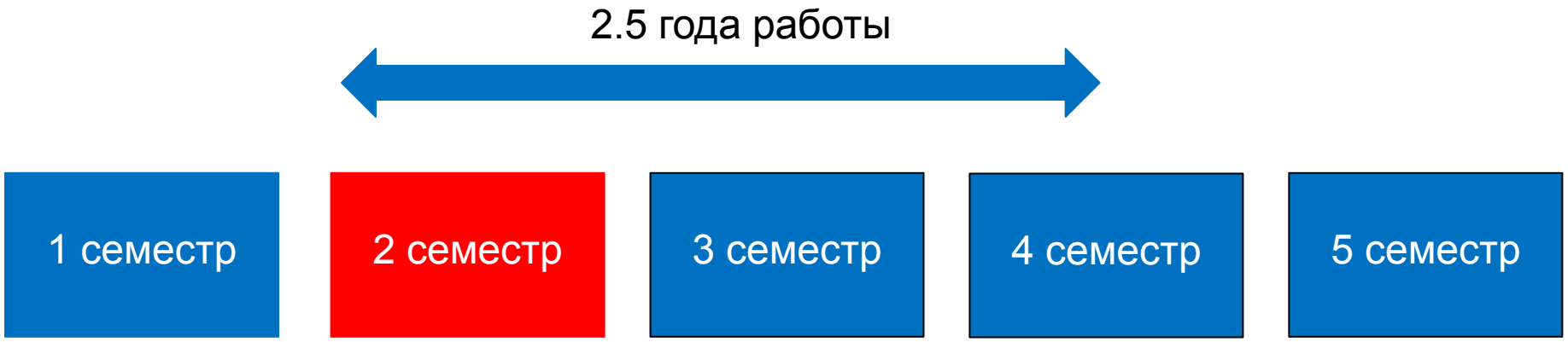
1 семестр	2 семестр	3 семестр	4 семестр	5 семестр



- Создать свой main
- Создать свой CMakeLists.txt, указать везде его существование
- Все положить в нужное место
- Указать изменения в конфигурациях CI
- Собрать проект

Введение

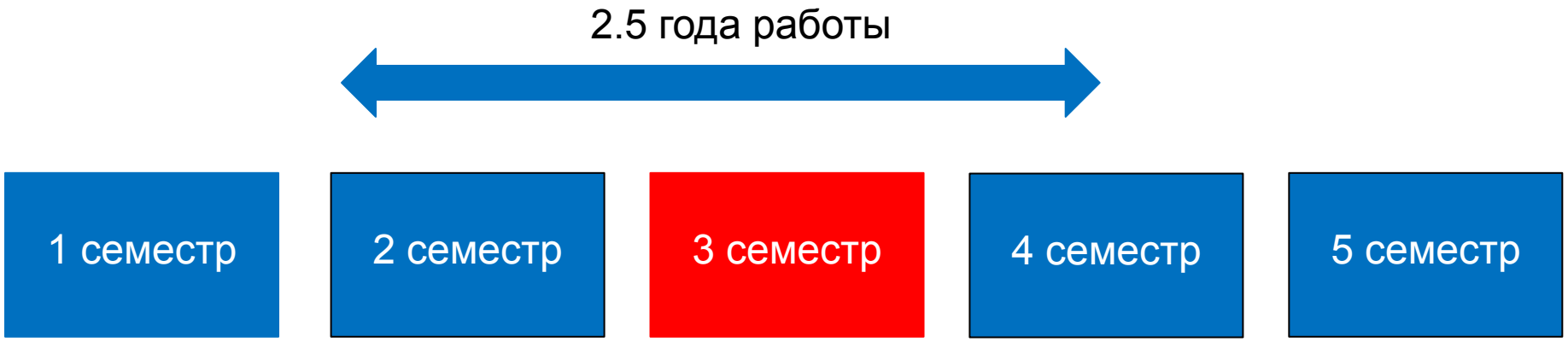
УНИВЕРСИТЕТ ЛОБАЧЕВСКОГО
НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ



- Создать свой main
- Создать свой CMakeLists.txt, указать везде его существование
- Все положить в нужное место
- Указать изменения в конфигурациях CI
- Собрать проект

Введение

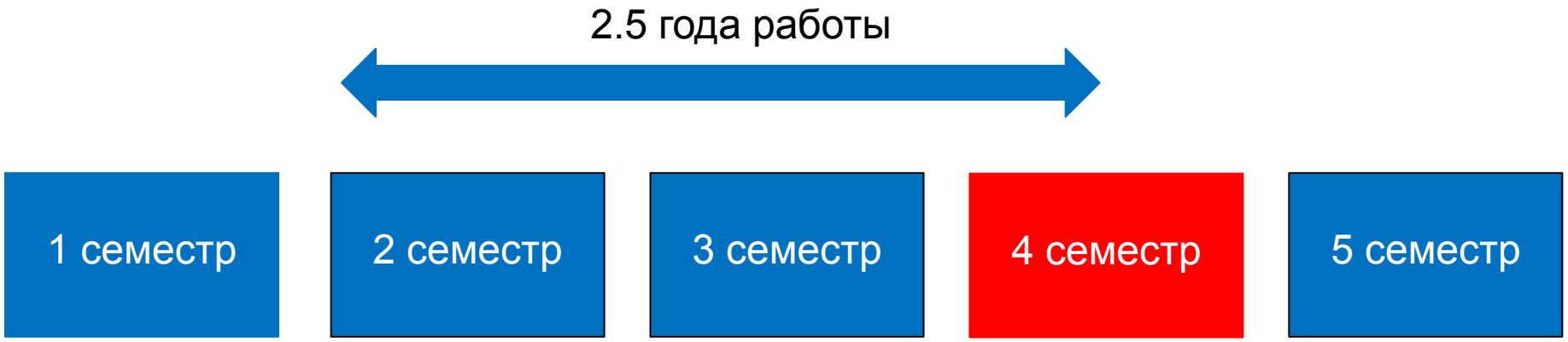
УНИВЕРСИТЕТ ЛОБАЧЕВСКОГО
НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ



- Создать свои google тесты в main
- Найти нужный образец CMakeLists.txt
- Все положить в нужное место, правильно назвать работу
- Собрать проект
- Не списывать!

Введение

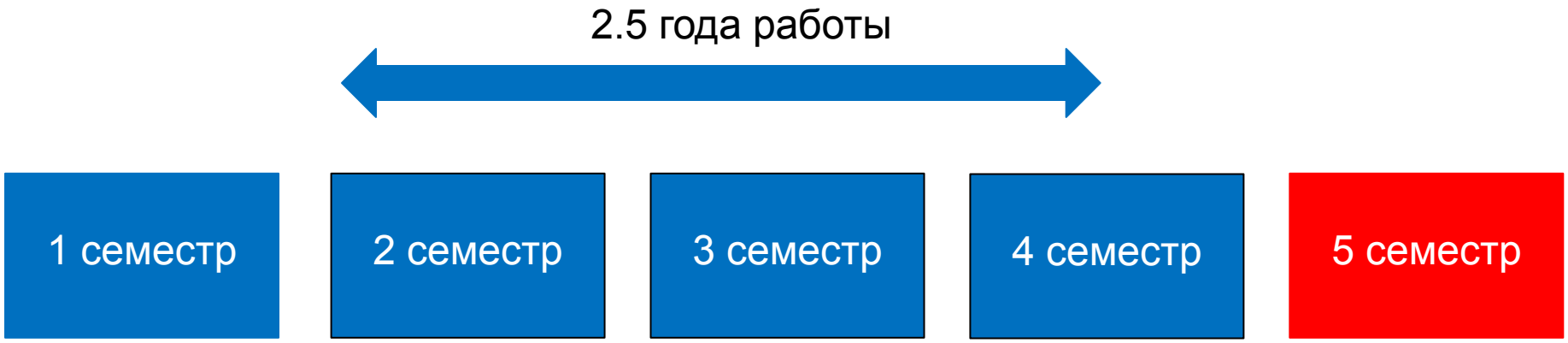
УНИВЕРСИТЕТ ЛОБАЧЕВСКОГО
НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ



- Создать свои google тесты в main
- Найти нужный образец CMakeLists.txt
- Все положить в нужное место, правильно назвать работу
- Собрать проект, пройти все проверки на качество
- Не списывать!

Введение

УНИВЕРСИТЕТ ЛОБАЧЕВСКОГО
НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

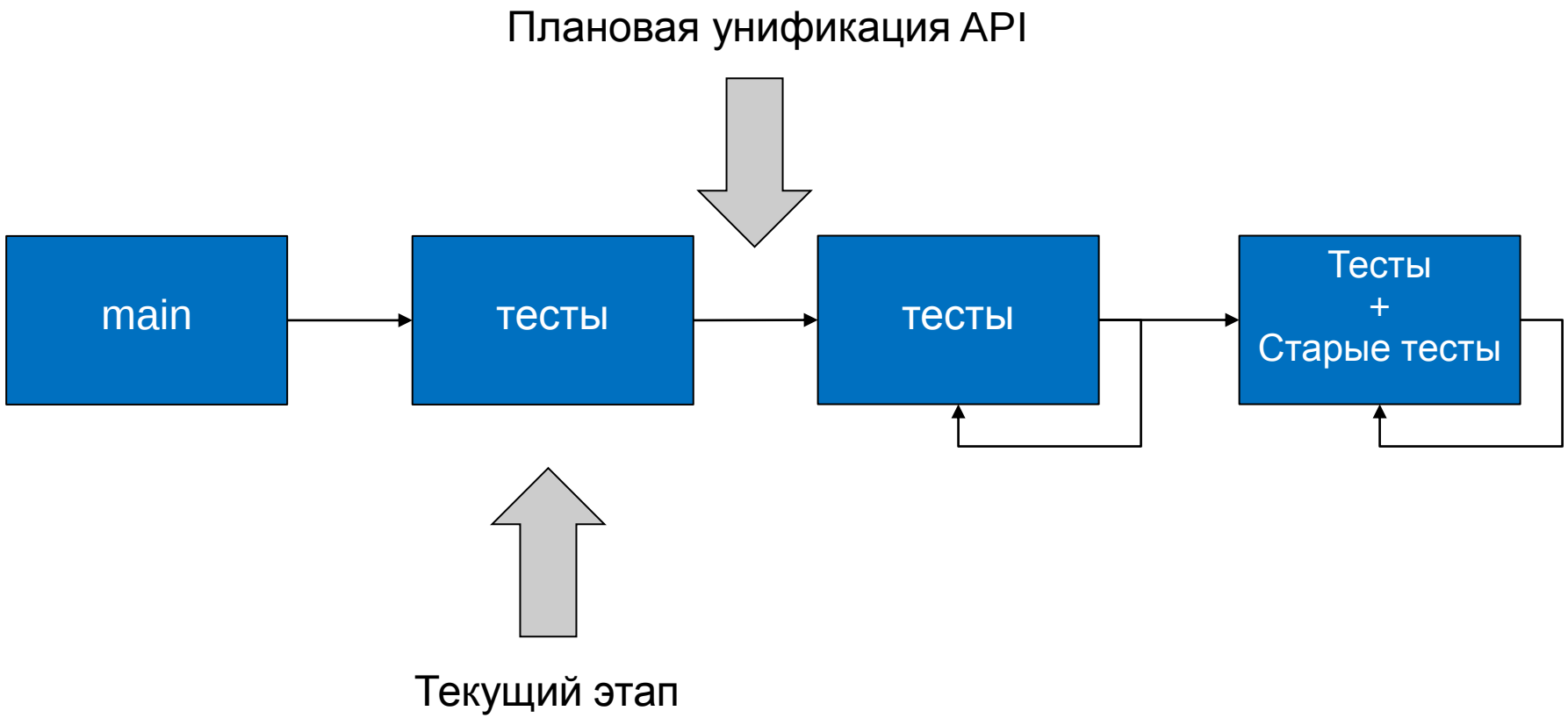


- Создать свои google тесты в main
- Найти нужный образец CMakeLists.txt
- Все положить в нужное место, правильно назвать работу
- Собрать проект, пройти все проверки на качество
- Не списывать!
- Следить постоянно за работой.

Текущая база проекта

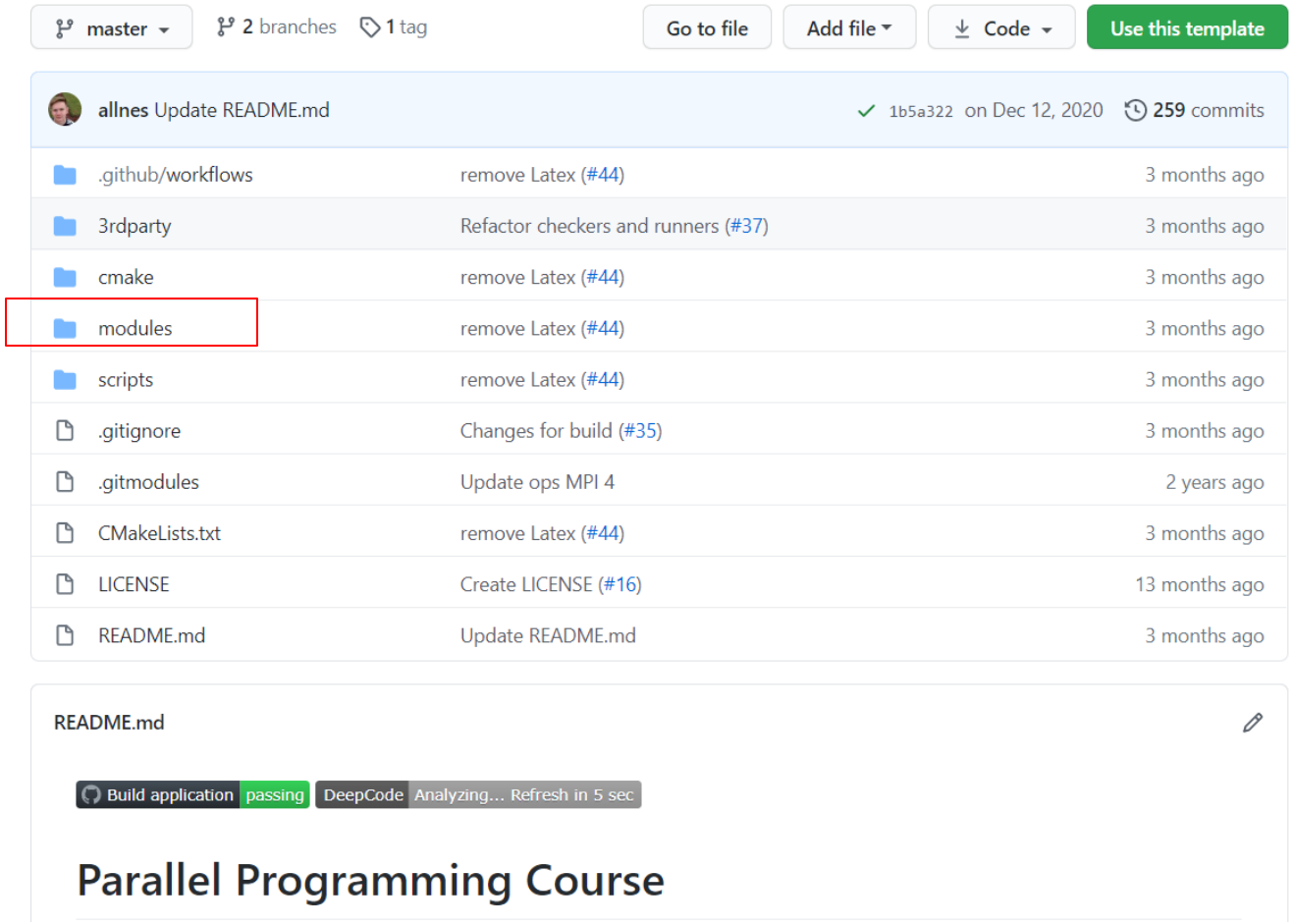
- Google Test Framework
  - Google Test MPI Listener (https://github.com/LLNL/gtest-mpi-listener)
  - Базовый Google Test Framework (для OpenMP и TBB)
- Непрерывная интеграция (CI)
  - Travis CI
  - AppVeyor
  - Github Actions
- Кроссплатформенная разработка
  - CMake и его особенности (TBB)
- Документирование и отчетность
  - LaTeX
- Проверка списывания исходного кода
  - JPlag

Идея автоматизации



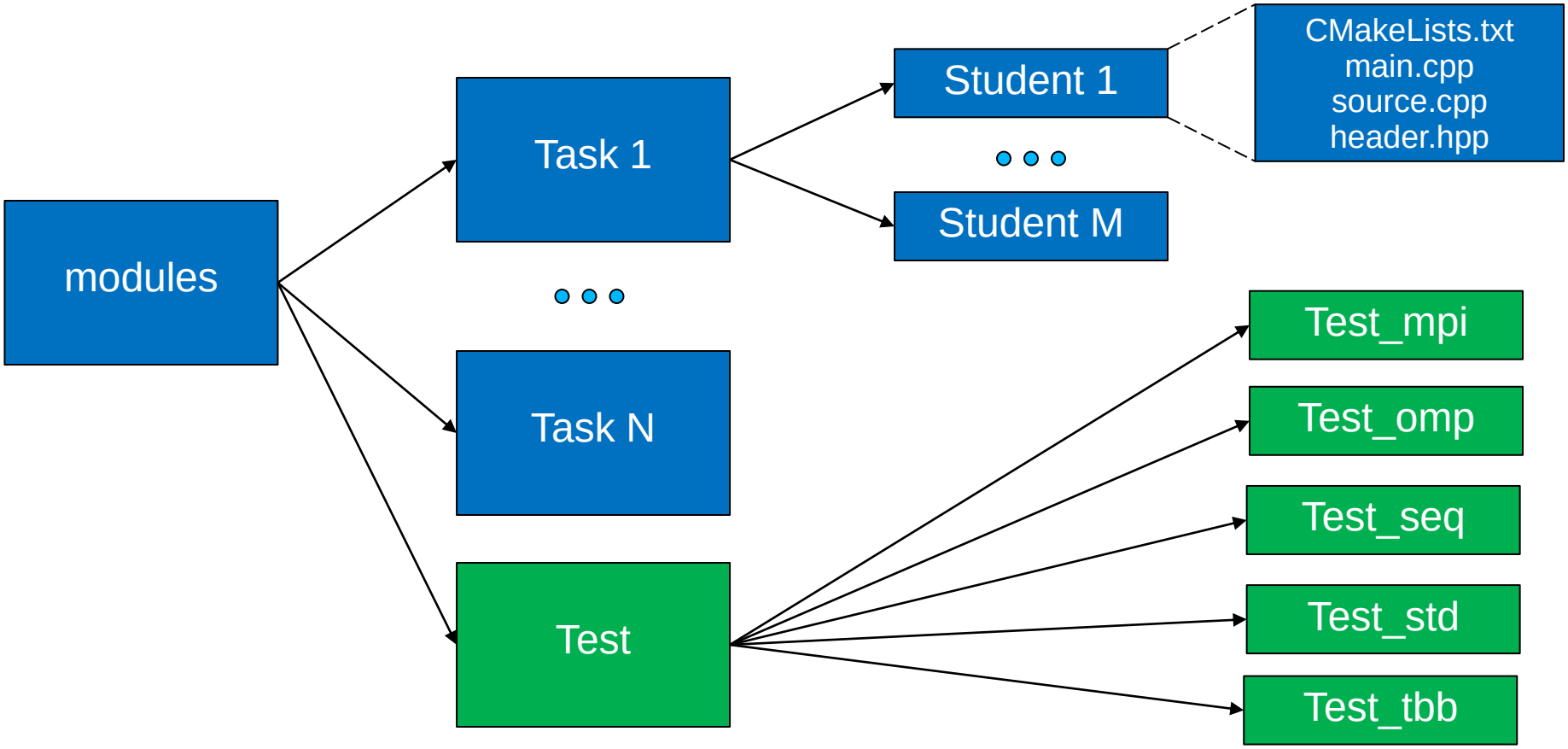
УНИВЕРСИТЕТ ЛОБАЧЕВСКОГО
НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Как выглядит на сейчас



https://github.com/learning-process/parallel_programming_course

Текущая архитектура проекта



Начальная идея жизнеспособности проекта:

- Промышленный проект, библиотека, программа считаются живыми пока есть в них нужда и имеет смысл продолжать разработку.
    - Следствие: идеи берутся из задач бизнеса, динамическое развитие
- Студенческий проект, библиотека, программа живы, пока студентам это нужно.
    - Следствие: идеи дает научный руководитель, сданный диплом, заморозка проекта, стартап
- Откуда берутся идеи для учебного проекта и как сделать его динамически развивающимся?

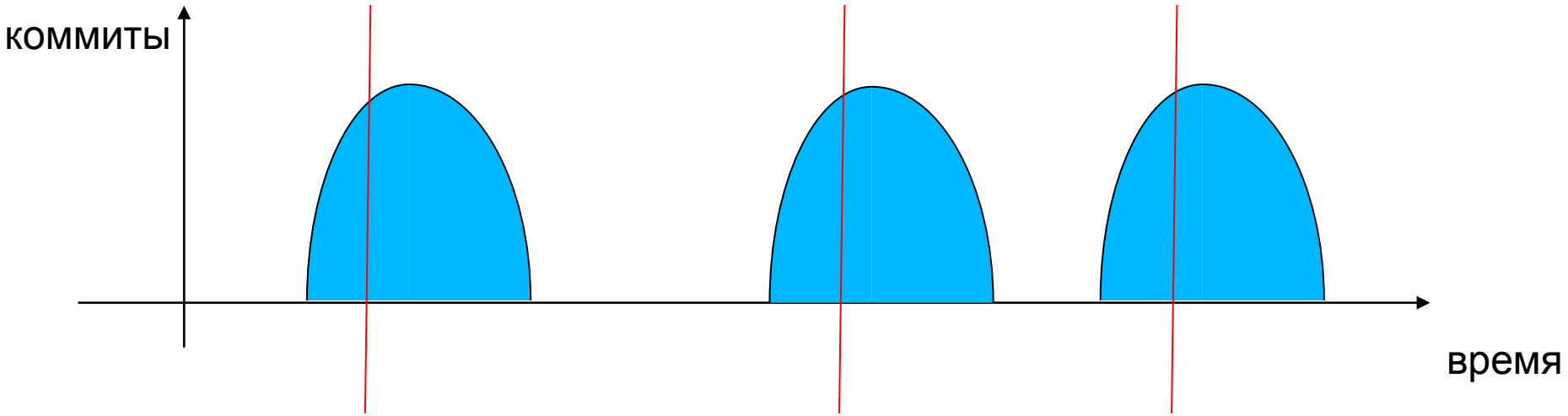
Основа получения идей и изменения проекта:

1) Попытки обмануть систему

2) Простота проекта для простого порога вхождения (для всех)

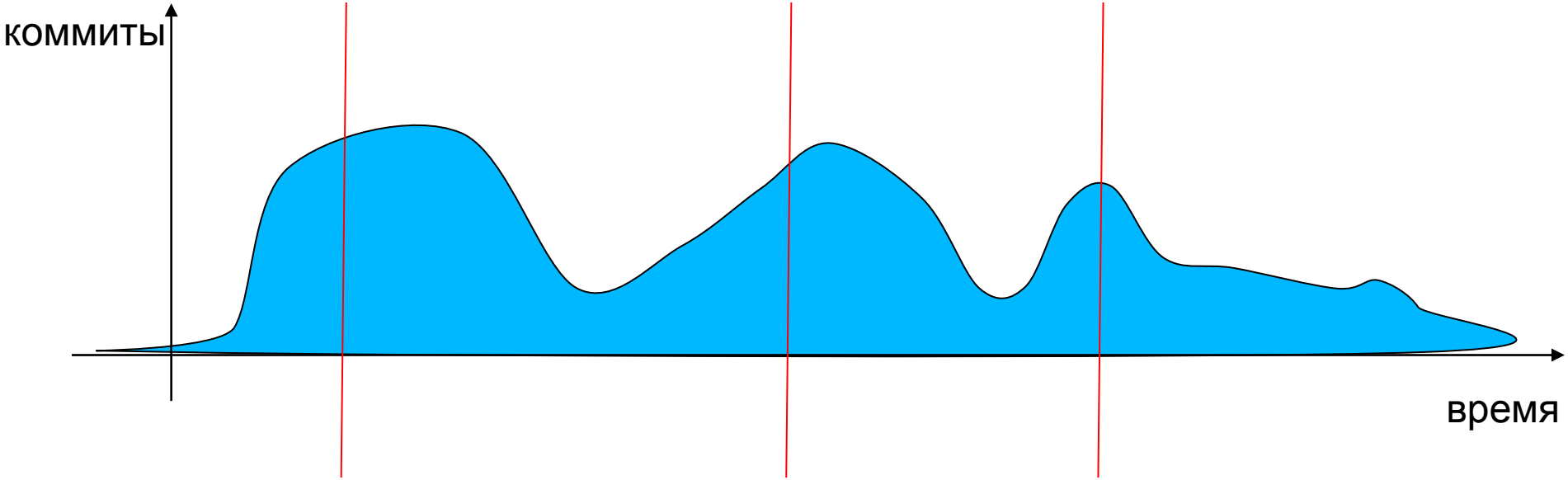
3) Текущая идея контроля обучения

- Примем за основу: идеальный код сразу не пишется
  - Следствие – для улучшения нужны непрерывные изменения кода
  - В противном случае - для нескольких задач в семестре будет дискретные значения изменений



3) Текущая идея контроля обучения

- Примем за основу: идеальный код сразу не пишется
  - Следствие – для улучшения нужны непрерывные изменения кода
  - В противном случае - для нескольких задач в семестре будет дискретные значения изменений



Принципы работы в курсе

Принципы коллективной ответственности

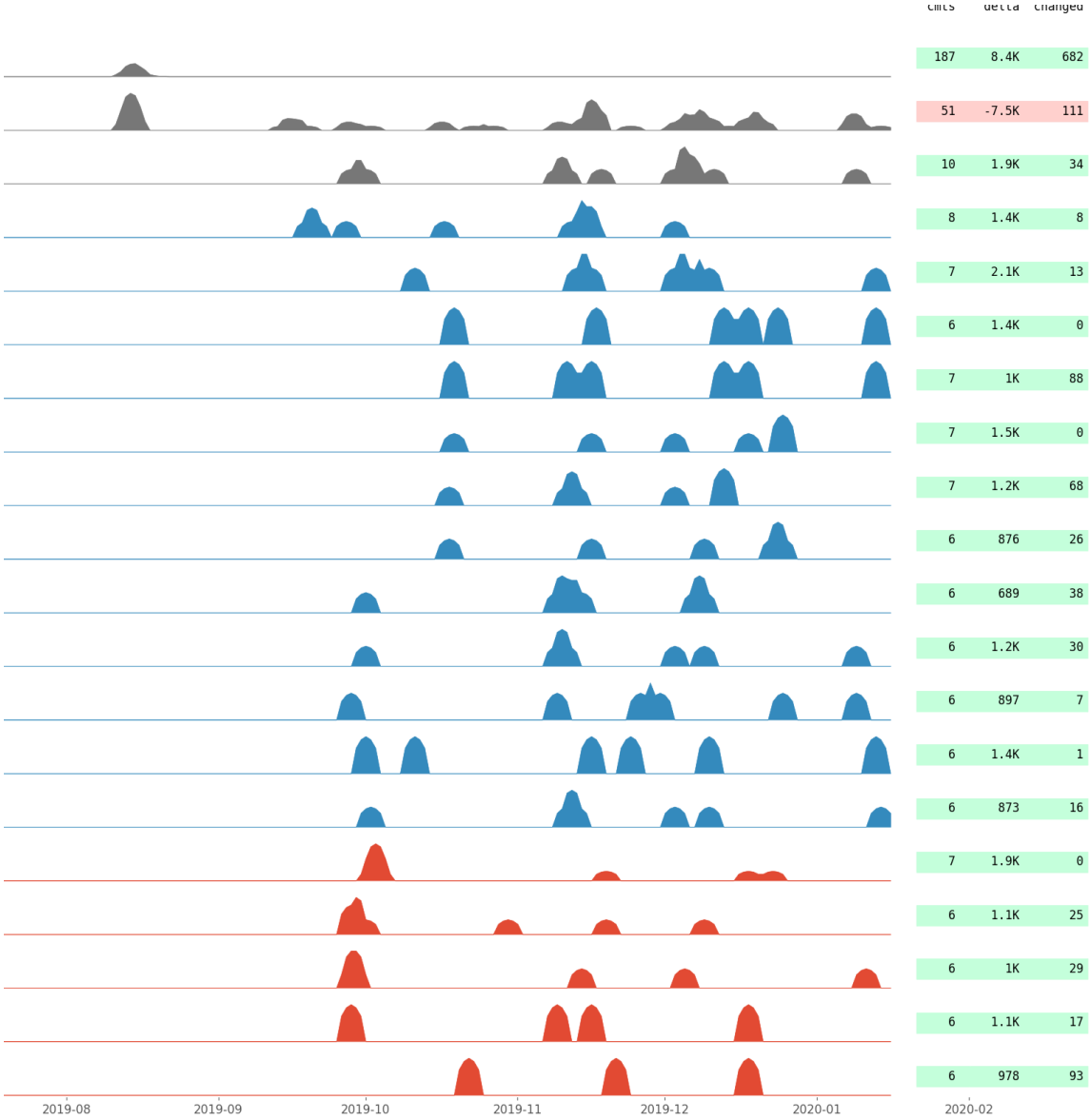
- Общая очередь в системах непрерывной интеграции
- Откат работы студента при падении всей библиотеки (стохастическое поведение параллельной программы)

Принципы достижения максимального качества

- Использование нескольких видов операционных систем и компиляторов
- Контроль стиля кодирования
- Добавление статического анализатора (cppcheck – тестовый режим)
- Добавление valgrind-анализатора для контроля утечек памяти (где возможно).
- Анализ кода с помощью искусственного интеллекта (тестовый режим)

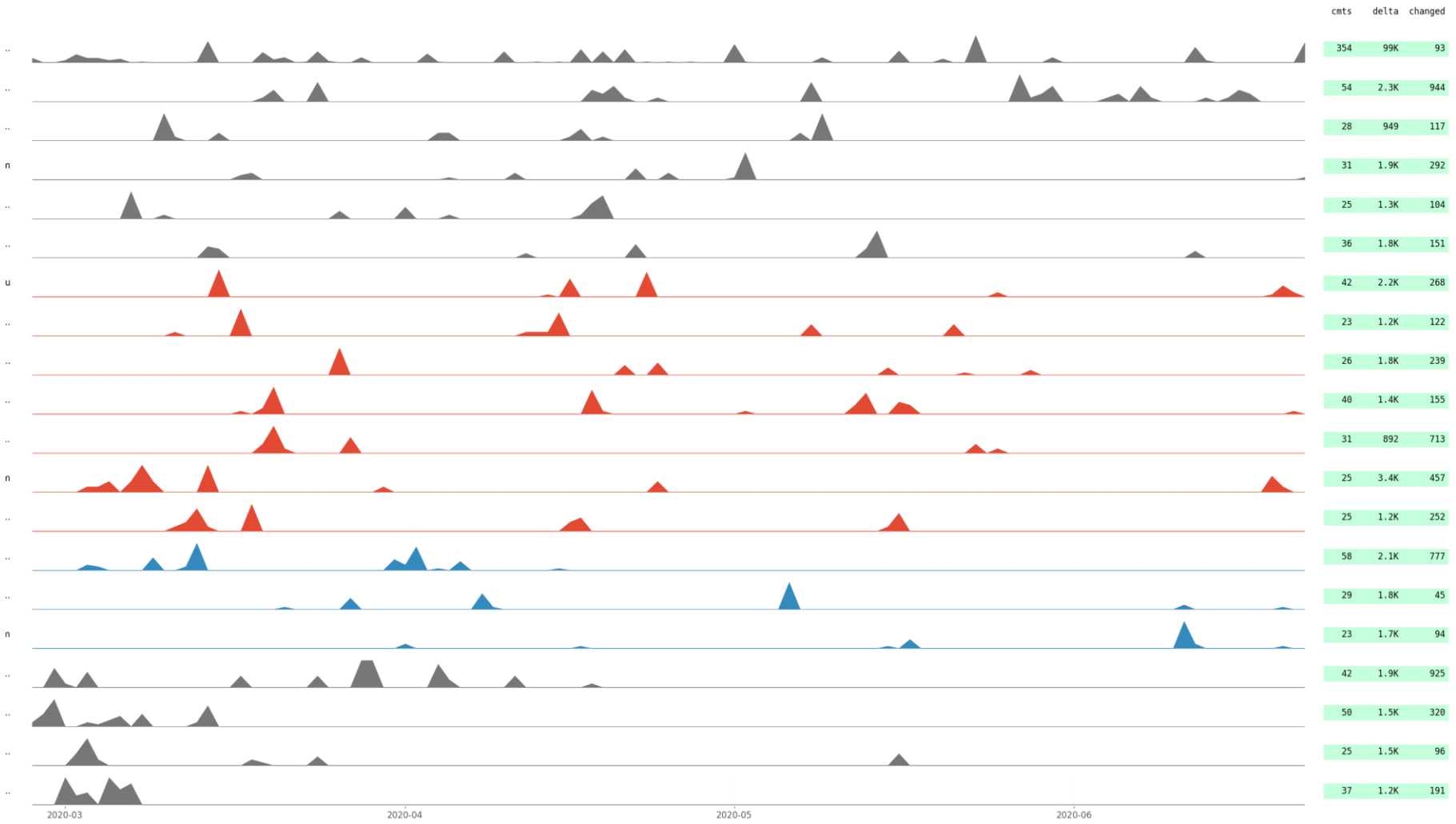
Состав проекта

УНИВЕРСИТЕТ ЛОБАЧЕВСКОГО
НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ



Состав проекта

УНИВЕРСИТЕТ ЛОБАЧЕВСКОГО
НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ



Дальнейшие планы работы:

- Унификация API проекта по разделам задач
- Исследование зависимости нагрузки репозитория от количества студентов

Спасибо за внимание!

УНИВЕРСИТЕТ ЛОБАЧЕВСКОГО

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ