

# Из python в go: как научить писать команду на другом языке

Вова Тельбухов  
Старший разработчик





# Яндекс Смена

## Содержание

Преимущества перехода Python -> Go	01
Опыт обучения команды	02
Вызовы и решения	03
Результаты на примере моей команды	04

## Сравнение языков

	Python	Golang	C++
Скорость	Медленный, интерпретируемый язык	Быстрый, компилируемый язык	Очень быстрый, компилируемый язык
Простота синтаксиса	Простой, читабельный	Умеренный, проще C++, сложнее Python	Сложный, много особенностей
Скорость развертывания	Быстрая, много инструментов стандартной библиотеки	Умеренная, требует компиляции	Медленная, требует компиляции и настроек
Параллелизм	Можно, но не нужно	Встроенная поддержка, легковесные горутины	Можно, нужно
Управление памятью	Автоматическое (сборка мусора)	Автоматическое (сборка мусора)	Ручное, возможны утечки
Железо	Требует много ресурсов	Эффективное использование	Очень эффективное
Обновления	Регулярно	Регулярно, чаще чем у Python	Редко, но стабильно

# Этапы перехода на go

01

## Оценка, анализ, постановка целей и найм

Проведение оценки и анализа текущего состояния проекта, постановка целей перехода на Go. Найм ключевых разработчиков, имеющих промышленный опыт разработки на Go.

02

## Создание сервиса на go и init commit

Разработка начальной версии сервиса: выполнение init commit и создание первой ручки. Создание базовой архитектуры и принципов разработки.

03

## Обучение команды

Проведение лекций и тренингов для команды, охватывающих основные аспекты Go

04

## Окончательный вынос сервиса на Go

Полный вынос функционала сервиса на go, прекращение разработки новых фич на Python (с исключениями). Все новые задачи выполняются только на Go

05

## Полное разделение сервиса и команд

Полное разделение команд по направлениям: одна команда занимается развитием сервиса для исполнителей, другая — для партнеров. Минимизация взаимных помех и повышение эффективности работы.

# Трудности

## 01 Найм опытных go-разработчиков

Найти специалиста, готового возглавить процесс перехода на Go, оказалось сложной задачей, так как не многие готовы взяться за создание нового сервиса с нуля

## 02 Обучение команды с различными уровнями знаний

В команде были разработчики с разным уровнем знаний о Go, от новичков до тех, кто уже немного работал с языком

## 03 Принятие новых архитектурных подходов

Команде было сложно перейти на новые архитектурные подходы и практики, принятые в Go

## 04 Начальные этапы развития инфраструктуры под golang

Некоторых необходимых инструментов разработки не хватало, иногда надо будет изобретать велосипеды

## 05 Сложность разделения монолита

Разделение монолита и перенос функционала на Go сопровождалось сложностями, связанными с тем, что новые фичи должны быть напилены в сжатые сроки

# Методы обучения

01

## Индивидуальные синки

Созвоны по конкретным техническим вопросам по моей инициативе или просьбе коллег с целью разобраться в каком-то конкретном вопросе

02

## Постоянная поддержка и помощь в чатах

Код-ревью, обсуждения в чате, помощь в реальных задачах. Эффект от обучения проявляется в долгосрочной перспективе

03

## Общие лекции

Регулярные лекции, охватывающие основные аспекты Go и сравнение с Python для облегчения понимания



# Первые лекции

## 01 Общий обзор Go

- IDE, слои архитектуры, подходы к тестированию, основные фреймворки
- Разбор полного цикла задачи: от создания хэндлера до интеграции с разными слоями

## 02 Основные структуры данных

- Обработка ошибок, бенчмарки, указатели, различные структуры данных
- Интерактивные задачи: «Что выведет этот код?»





# Интерактивный подход

## 01 Положительный фидбек

- Коллегам понравился интерактивный формат
- Отмечена высокая полезность знаний и реальных примеров
- Готовность к символическим взносам для призов на будущих лекциях

## 02 Повышение интереса

Интерактивные задачи стимулируют внимание и интерес к материалу



# Запланированные этапы обучения

## 01 Лекция 3: Параллелизм в Go

Принципы параллелизма, примеры и лучшие практики. Разбор реального примера от поиска узких мест, в том числе с помощью профилировщика, до реализации

## 02 Лекция 4: Архитектура проекта и ограничения, которые она за собой несет

Искусственные ограничения для устойчивой архитектуры и легкой поддержки проекта. Разберем более подробно архитектуру и уже допущенные ошибки



# Достигнутые результаты

## Сокращение количества тредов в Pull Request'ах

Определите, какую проблему аудитории вы хотите решить или на какой вопрос хотите попытаться ответить



## Скорость адаптации

Первые заметные результаты появились через 2 недели после начала обучения



## Качественные изменения в коде

Коллеги стали следовать правилам архитектуры, размещая код в правильных слоях, что улучшило структуру, читаемость кода, а в перспективе облегчит поддержку проекта



## Улучшение коммуникации

Увеличение активности в чатах, рост числа вопросов и открытость к обсуждению, что способствует лучшему взаимодействию и обучению в команде



# Из python в go: как научить писать команду на другом языке

Вова Тельбухов  
Старший разработчик

