

Обмен опытом в программировании

Регламент

Регламент 15-20 минут на выступление

Регламент

15-20 минут на выступление

10 минут на вопросы

Регламент

15-20 минут на выступление

10 минут на вопросы

В конце — кулуарное общение

**Без практики – не работает:
как передавать знания по
программированию
(и не только)**

Продуктовые метрики обучения

Продуктовые метрики обучения

- **процент проходимости**

Продуктовые метрики обучения

- процент проходимости
- оценка контента

Продуктовые метрики обучения

- процент проходимости
- оценка контента
- оценка преподавателей

Продуктовые метрики обучения

- процент проходимости
- оценка контента
- оценка преподавателей
- NPS



Для чего нужны курсы?

Для чего нужны курсы?

- продвинуться в финансах

Для чего нужны курсы?

- продвинуться в финансах
- продвинуться в карьере

Для чего нужны курсы?

- продвинуться в финансах
- продвинуться в карьере
- абстрактно развиваться

Устроиться на работу

Карта компетенций

Go-разработчик

TASKS	KNOWLEDGES	INSTRUMENTS	PROCESSES	SOFT SKILLS
Задачи, которые должен уметь решить	Теоретические знания, которые должен применять в работе	Инструменты, которыми владеет (подразумеваем, что студент уже владеет необходимыми программами на уровне выше среднего)	Рабочие процессы, в сути которых должен разбираться	
Разрабатывать и поддерживать web-приложения на языке Golang	Программирование: Основы алгоритмизации и ооп JavaScript (front-end) Go (back-end) (основы языка, асинхронная работа)	Linux	CI/CD / автотесты	умение вести проект
Создавать и поддерживать приложения с микросервисной архитектурой	Микросервисы. Разработка микросервисной архитектуры: пользовательский интерфейс, БД, сервер	ПО для автоматизации и управления приложением: Git (+Git Flow), Docker, Kubernetes		владение инструментами планирования, управления задачами (Agile, SCRUM, Kanban)
Создавать нагруженные сервисы	Работа с высоконагруженными системами	Системы кэширования и передачи сообщений: Kafka, Rebbit, NSQ		навык таймменеджмента
Работать с Базами Данных	Работа с базами данных (MySQL, NoSQL, PostgreSQL)	Аналитика: ClickHouse		умение работать в команде
Покрывать тестами свой код	Сервер (HTTP/HTTPS, JSON)			
Применять в работе по для автоматизации развертывания и управления приложениями, версиями	Верстка адаптивных макетов/сайтов (HTML, CSS, JS) Веб-сервисы			
Делать код-ревью				

обучение

обучение =

**обучение = процесс
разработки**

обучение = процесс разработки

бэклог = карта компетенций

обучение = процесс разработки

бэклог = карта компетенций

задача = компетенция

обучение = процесс разработки

бэклог = карта компетенций

задача = компетенция

декомпозиция

обучение = процесс разработки

бэклог = карта компетенций

задача = компетенция

декомпозиция

реализация = получение знания

обучение = процесс разработки

бэклог = карта компетенций

задача = компетенция

декомпозиция

реализация = получение знания

тестирование = выполнение заданий

обучение = процесс разработки

бэклог = карта компетенций

задача = компетенция

декомпозиция

реализация = получение знания

тестирование = выполнение заданий

релиз в прод = знание получено

**Составление скиллсэта —
это очень важная
подготовительная работа**

Обучать людей

Обучать людей

- **условие**

Обучать людей

- **условие**
- **цель**

Обучать людей

- **условие**
- **цель**
- **решение**

**Мы не только учим —
но и учимся сами**

Рынок труда



Рынок труда



Соискатели



СВЯЗНОСТЬ ЗНАНИЙ

Тест-дизайн

Тест-дизайн

Определение

Тест-дизайн

Определение

Тест-кейсы

Тест-дизайн

Определение

Тест-кейсы

Техники тест-дизайна

Тест-дизайн

Определение

Тест-кейсы

Техники тест-дизайна

Эквивалентное разделение

Тест-дизайн

Определение

Тест-кейсы

Техники тест-дизайна

Эквивалентное разделение

Анализ граничных значений

Реакция обучаемого





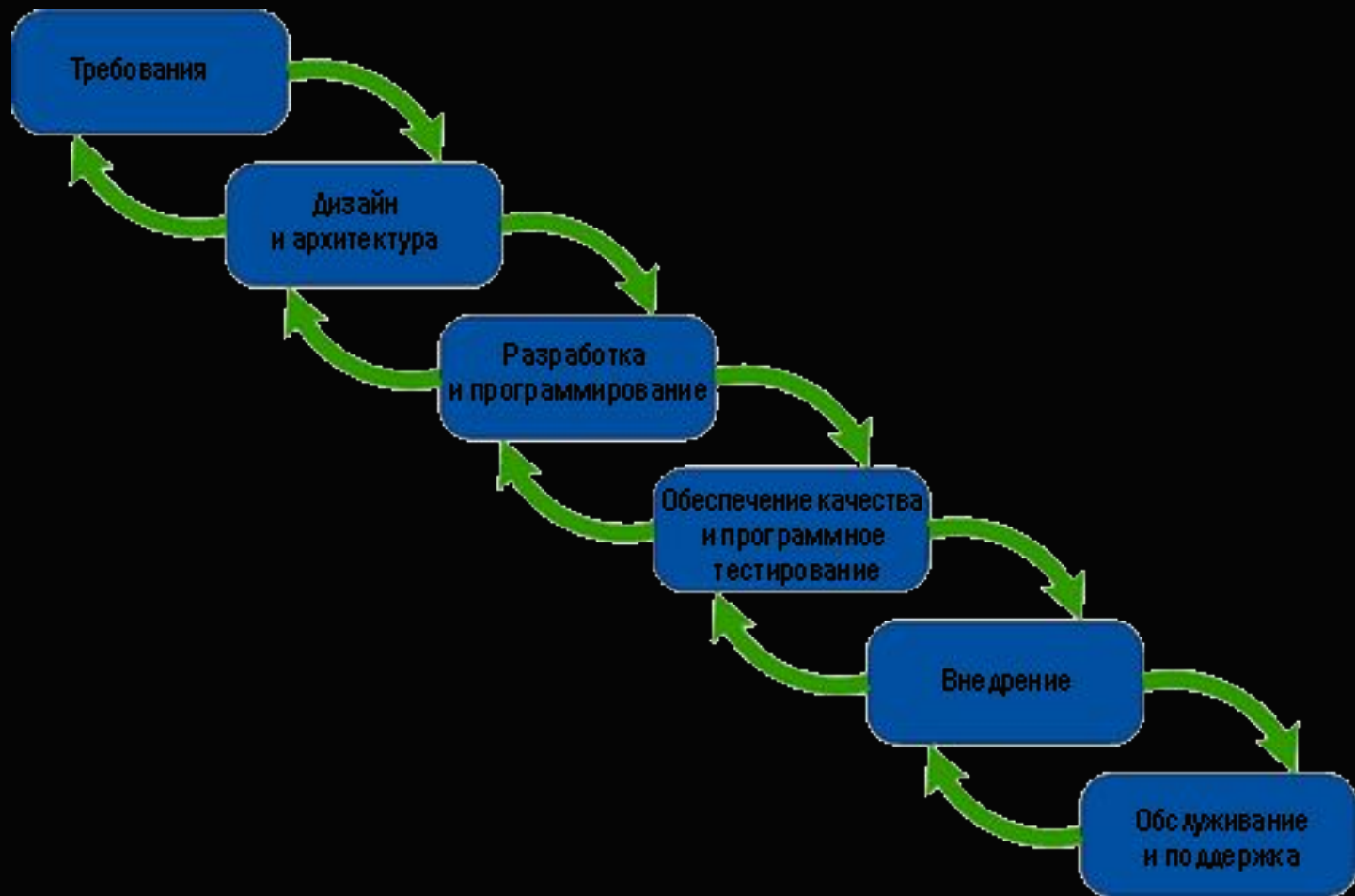
Brenan Keller

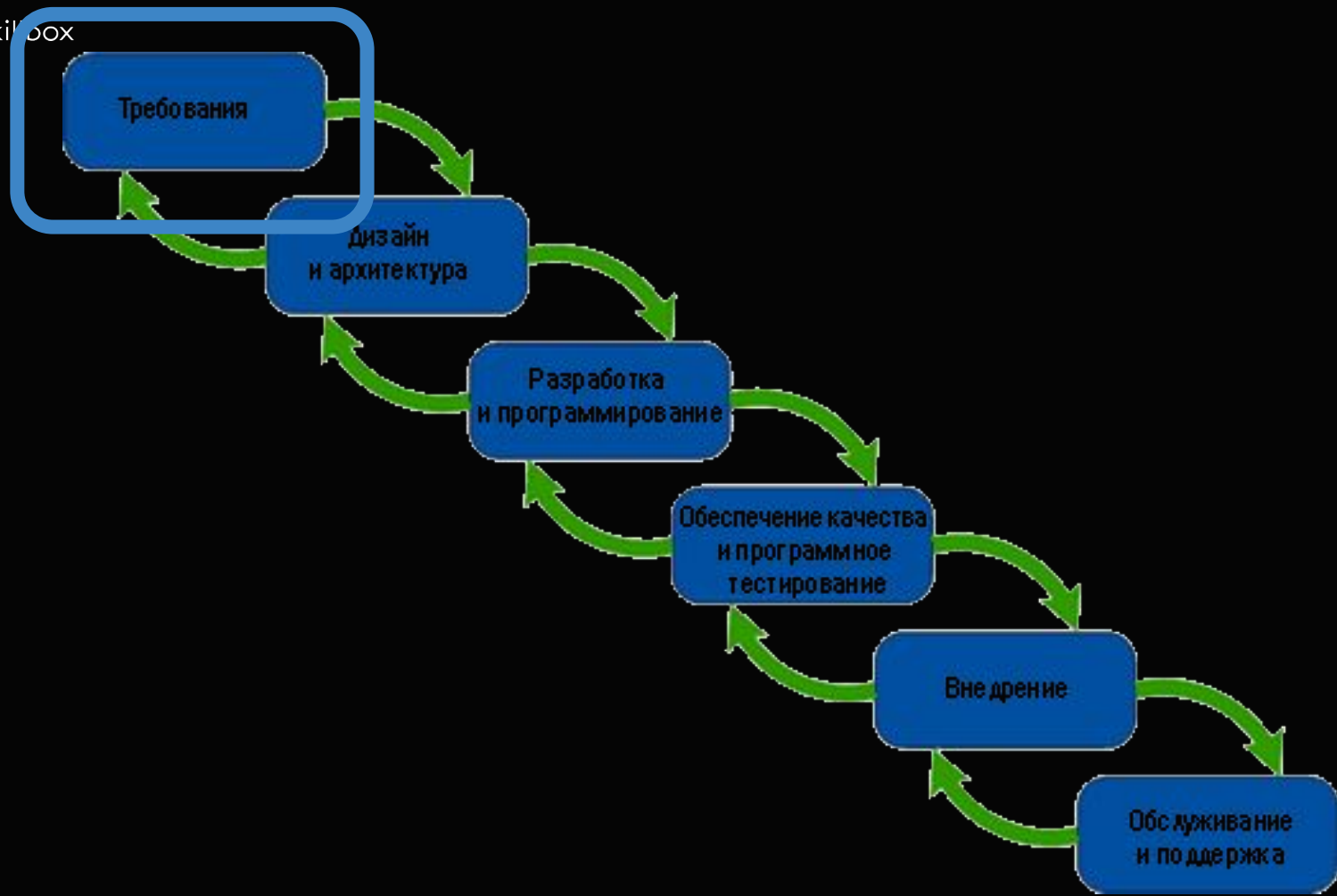
@brenankeller

A QA engineer walks into a bar.
Orders a beer. Orders 0 beers.
Orders 99999999999999 beers.
Orders a lizard. Orders -1 beers.
Orders a ueicbksjdhd.

First real customer walks in
and asks where the bathroom
is. The bar bursts into flames,
killing everyone.

Анонсируйте контекст





Skill box

Требования

дизайн
и архитектура

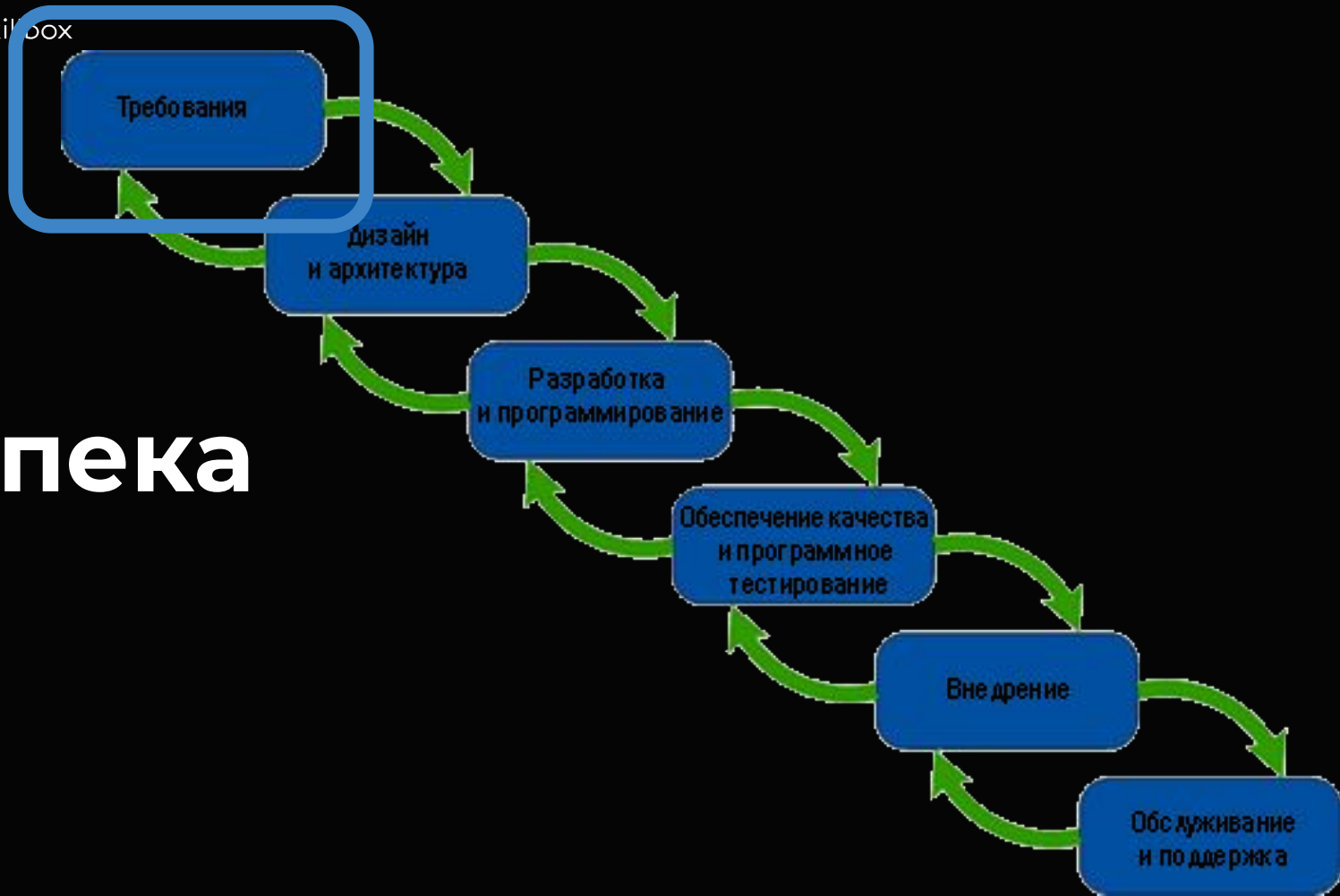
Разработка
и программирование

Обеспечение качества
и программное
тестирование

Внедрение

Обслуживание
и поддержка

Спека



Skill box

Требования

дизайн
и архитектура

Разработка
и программирование

Обеспечение качества
и программное
тестирование

Внедрение

Обслуживание
и поддержка

Спека Требования



Спека Требования



Skill box

Спека Требования

Перепроверь



Курс-документация

Переменные

Переменные Определение

Переменные

Определение

10 типов данных

Переменные

Определение

10 типов данных

Приведение типов

Переменные

Определение

10 типов данных

Приведение типов

Операторы

Переменные
Определение
10 типов данных
Приведение типов
Операторы
Преобразование

**Связность должна быть не
между знаниями, которые
вкладываются в человека**

**Связность должна быть
между тем, что ты хочешь
дать, и тем, что человек
уже знает**

Дать концепцию

Дать концепцию
Дать кусочек знания

Дать концепцию
Дать кусочек знания
Практика

Дать концепцию
Дать кусочек знания
Практика
Следующий кусочек

Дать концепцию

Дать кусочек знания

Практика

Следующий кусочек

Практика

Дать концепцию

Дать кусочек знания

Практика

Следующий кусочек

Практика

Знание посложнее

Говорите с аудиторией

Люди боятся показаться некомпетентными

Люди боятся показаться некомпетентными



Самостоятельные и кросс- проверки знаний и навыков

Когда можно использовать

Когда можно использовать Если что-то простое

Когда можно использовать Если что-то простое

Java. Как сделать так,
чтобы при
возникновении
Exception программа
не завершалась?

Когда можно использовать Если что-то простое

Java. Как сделать так,
чтобы при
возникновении
Exception программа
не завершалась?

*Написать в коде
конструкцию `try...catch`,
чтобы этот **Exception**
отлавливался*

**Когда можно использовать
Если что-то простое
Когда обучаемый уверенно
себя чувствует**

Спасибо!