Разработка ПО глазами менеджера и разработчика

Alexander Bychuk

VK Tech

@bychuk_as

a.bychuk@vk.team

C++ Russia 2023

tech

Александр Бычук

Director of Engineering

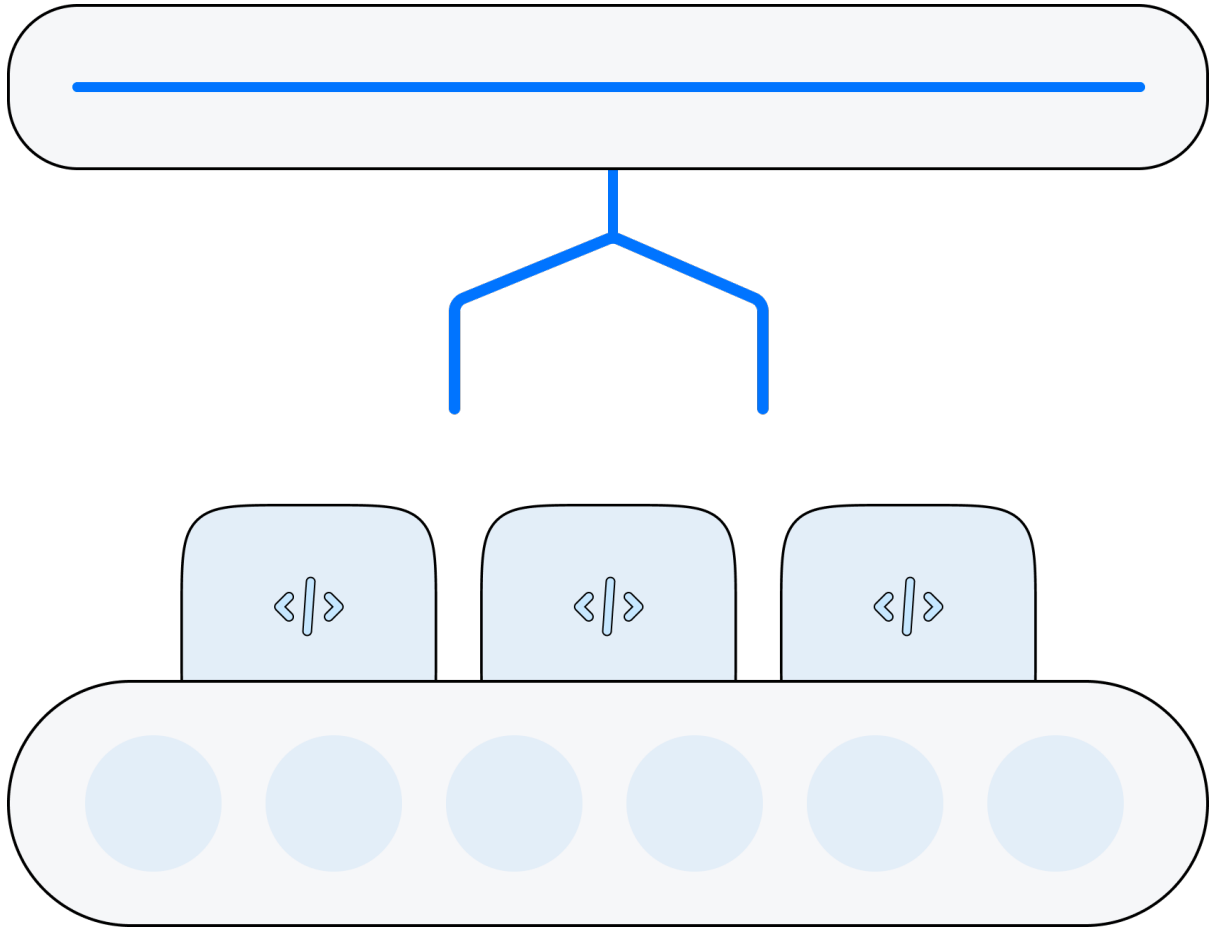
@bas524

bas524 (Alexander B)

Производство и проектная деятельность

Разработка ПО – это производство

Производство – процесс создания какого-либо продукта с использованием первичных (труд и капитал) и промежуточных факторов производства (сырье, материалы и т. п.)



Википедия

Методологии и фреймворки для управления проектами (кратко)

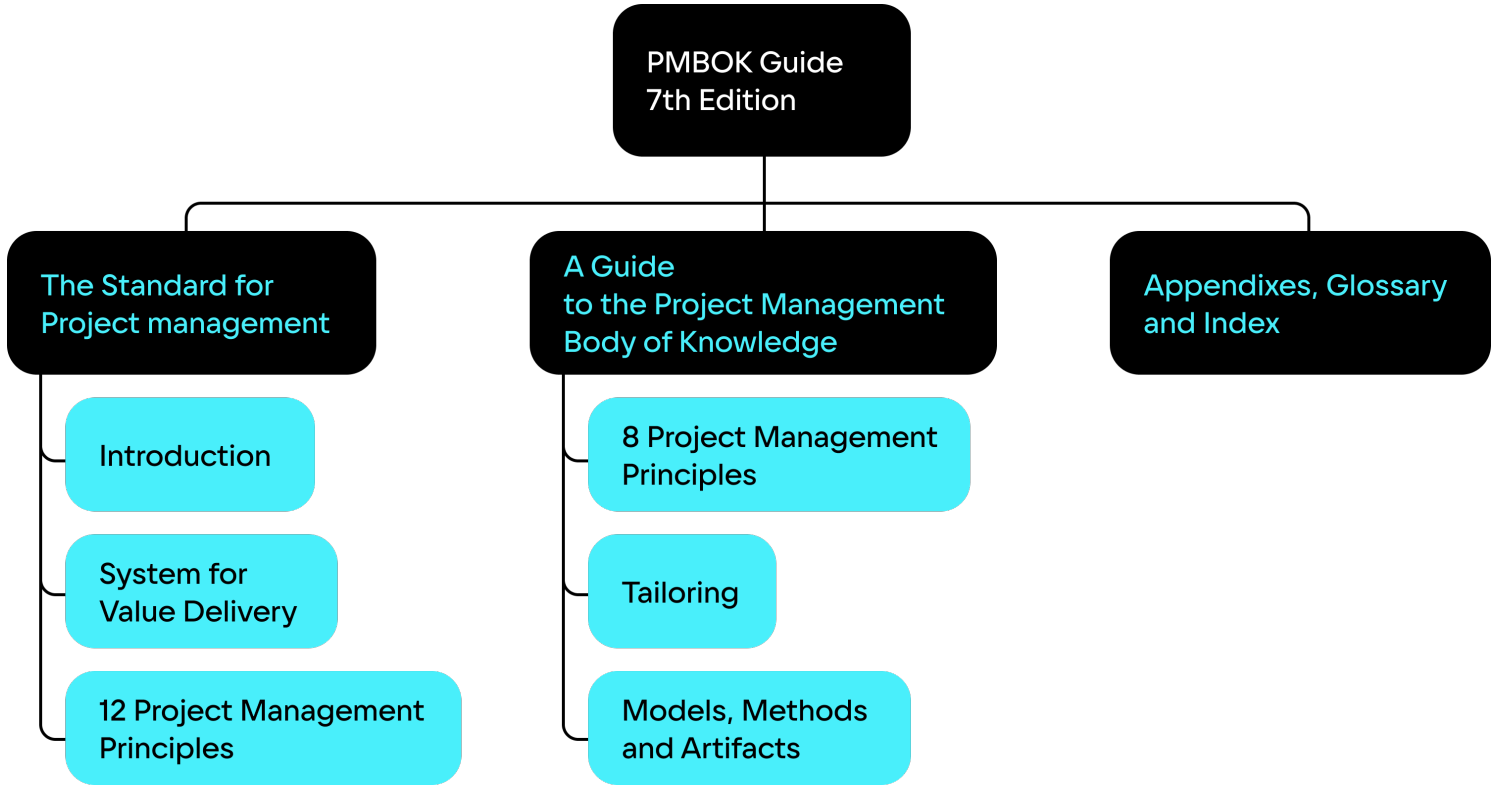
Фреймворки – это не только про JS

```
template<typename Float>
auto add(Float value1, Float value2) requires requires { false; } {
      return value1 + value2;
}
int main() {
      return add<int>(1.2f, 1.3f);
}
```

Agile

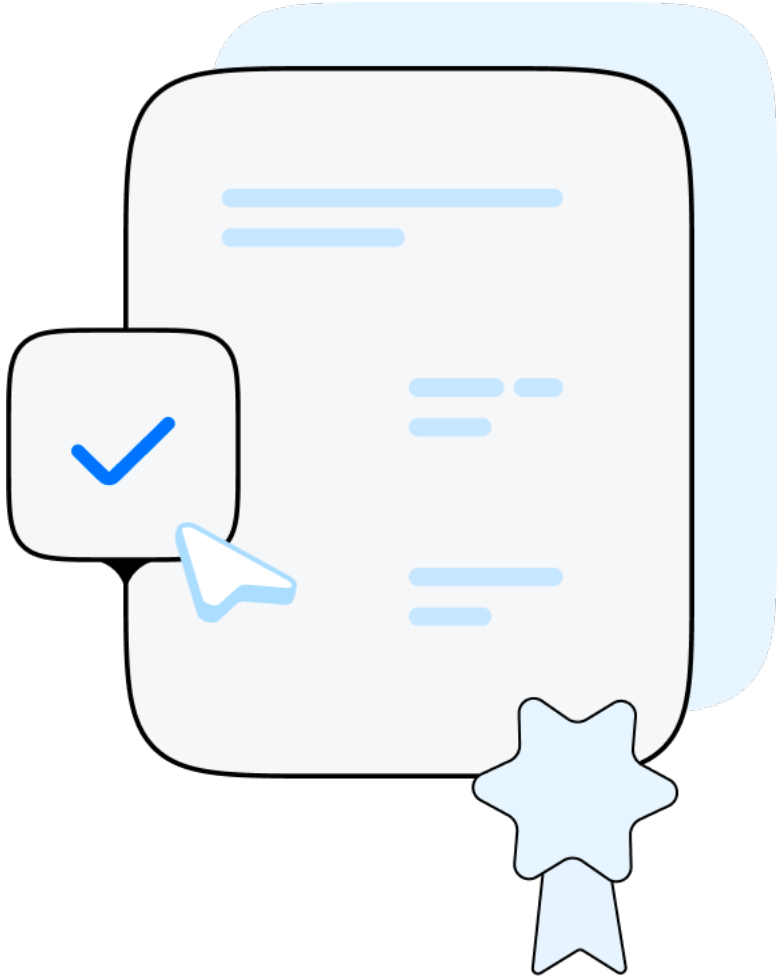
- Scrum
- Kanban
- ...

Rules an approaches

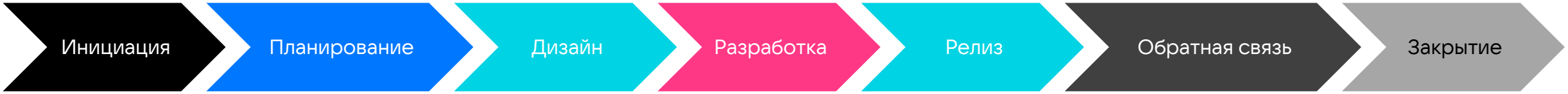


Methodology

- PRINCE2
- P2M
- ...

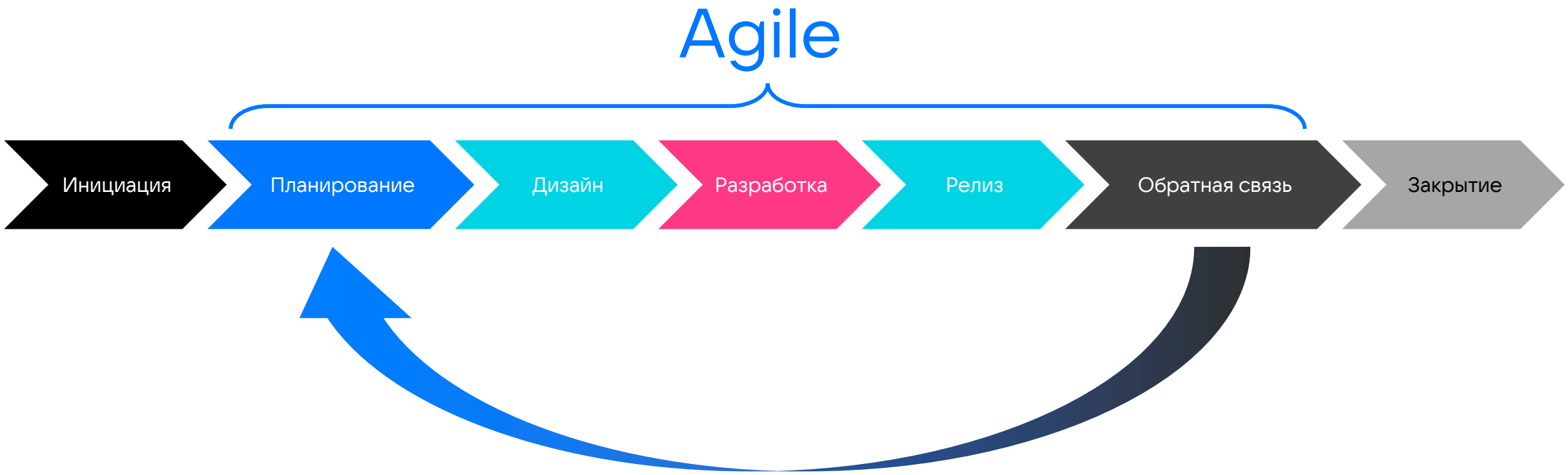


Основные стадии проекта

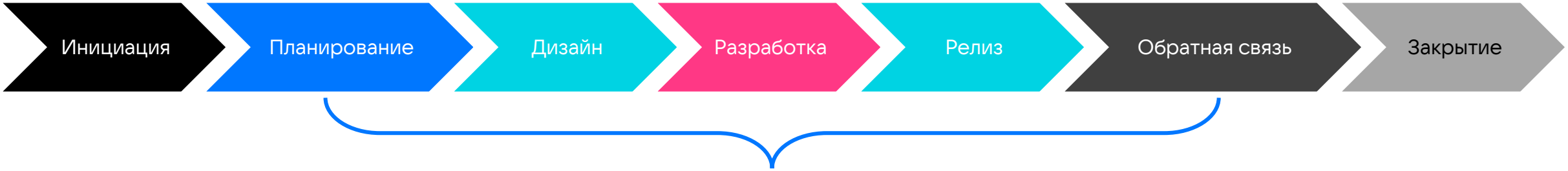



```
consteval int sqr(int n) {
        return n*n;
}
constexpr int dblsqr(int n) {
        return 2*sqr(n);
}
int main() {
        return dblsqr(10);
}
```

Основные стадии проекта



Роли, участники проекта



- Product Manager
- Project Manager
- Analyst
- Architect
- UI/UX Designer
- Developer
- Tester
- DevOps/ SRE
- Support other experts

Роли, участники проекта

Owner and
Top Managers

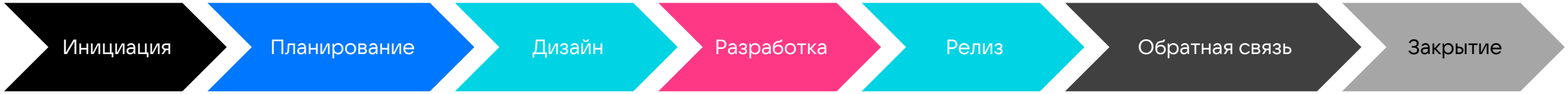
BisDev
Manager

Product
Manager

Sales & Marketing
specialist

Project Manager
other experts

Роли, участники проекта



Product Manager

Project Manager

Разработчик и остальные роли в проекте

```
? std::mutex mutex;
? std::uint32_t data{ 0 };
std::uint32_t increment() {
        std::lock_guard guard(mutex);
        ++data;
        return data;
}
// static_init_counter.h
std::uint32_t increment();
// static_init_data.cpp
#include "static_init_counter.h"
struct safe {
std::uint32_t index = increment();
};
safe safe_object;
```

Разработчик

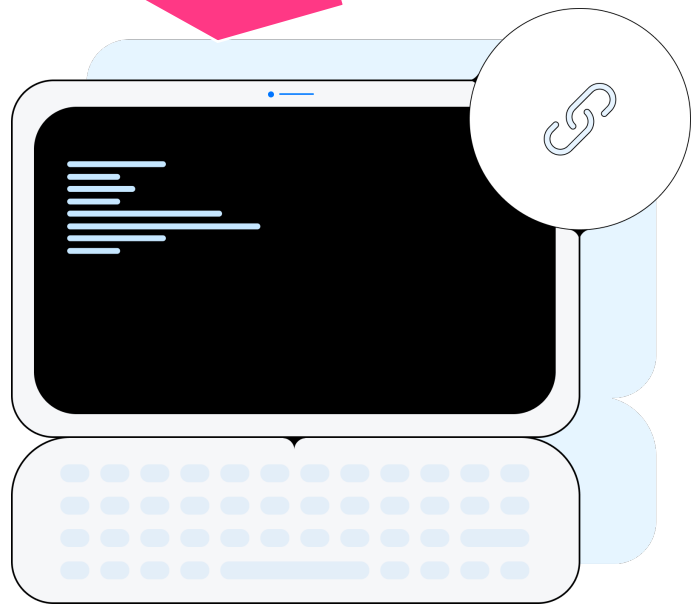
Инициация

Планирование

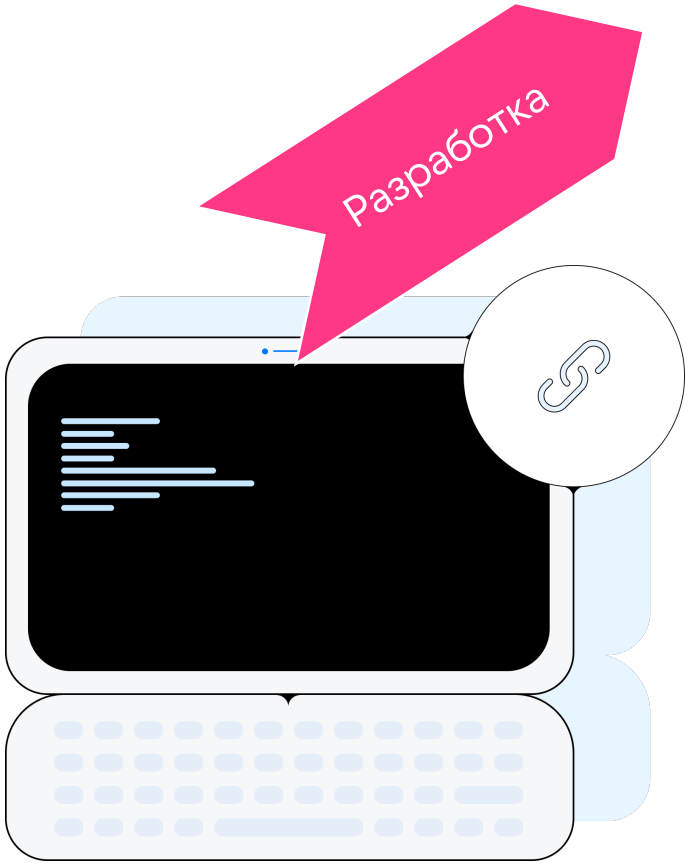
Дизайн

Разработка

Требования

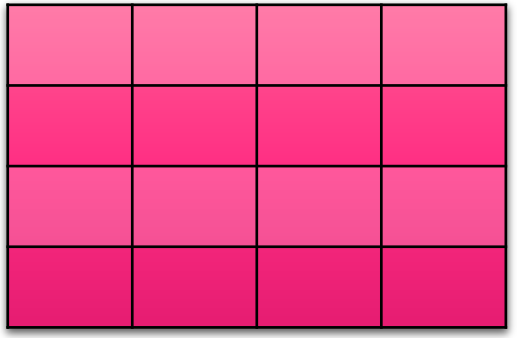


Разработчик

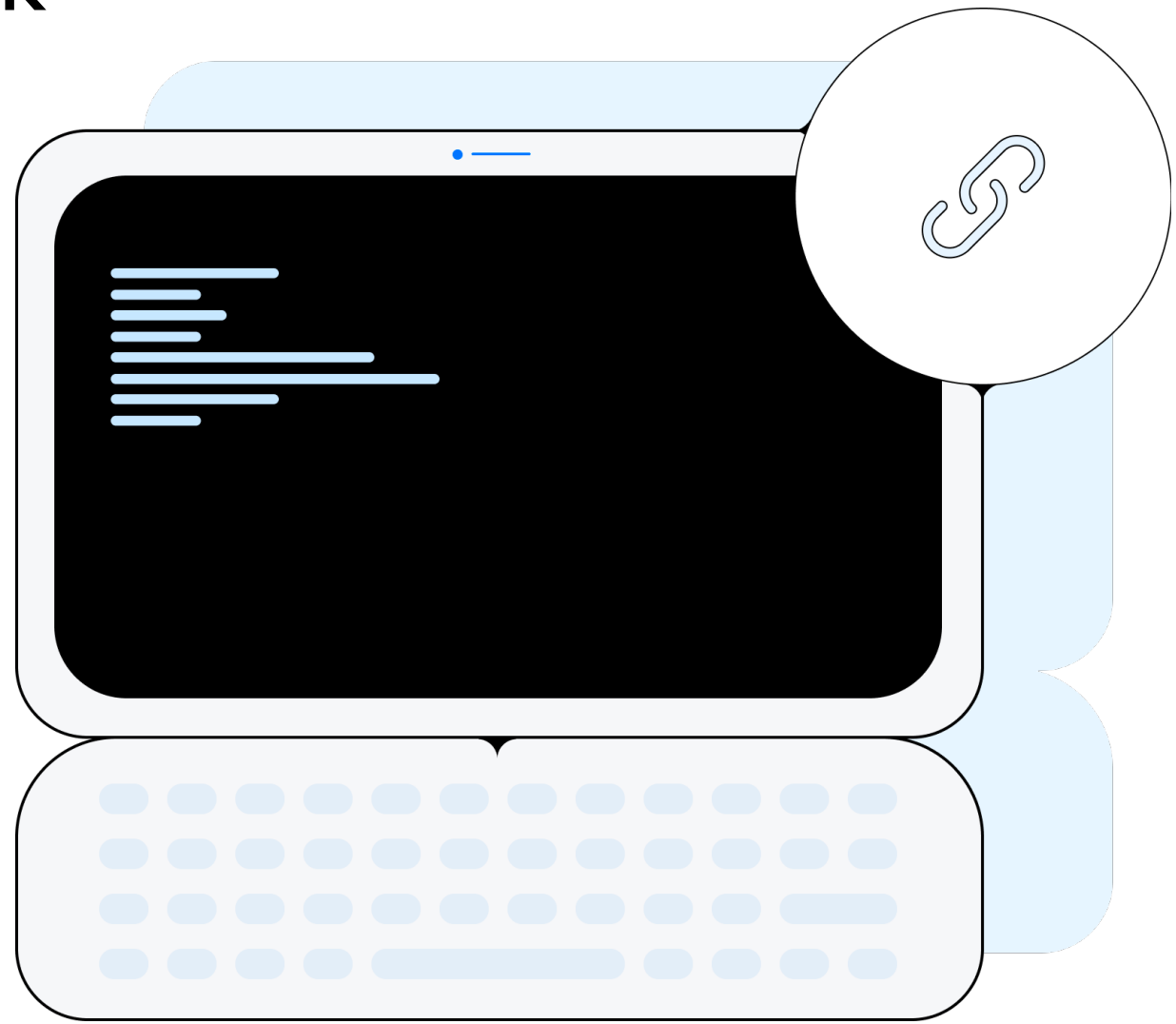


Релиз → Сопровождение → Закрытие

Продукт



Разработчик



Разработчик

Влияние

Для всех подготовительных этапов, разработчик – это источник ограничений	Разработка	Для всех этапов «после разработки», разработчик – это источник частей продукта
PM, Architect, Analyst…	Developers	QA, SRE, Sales…

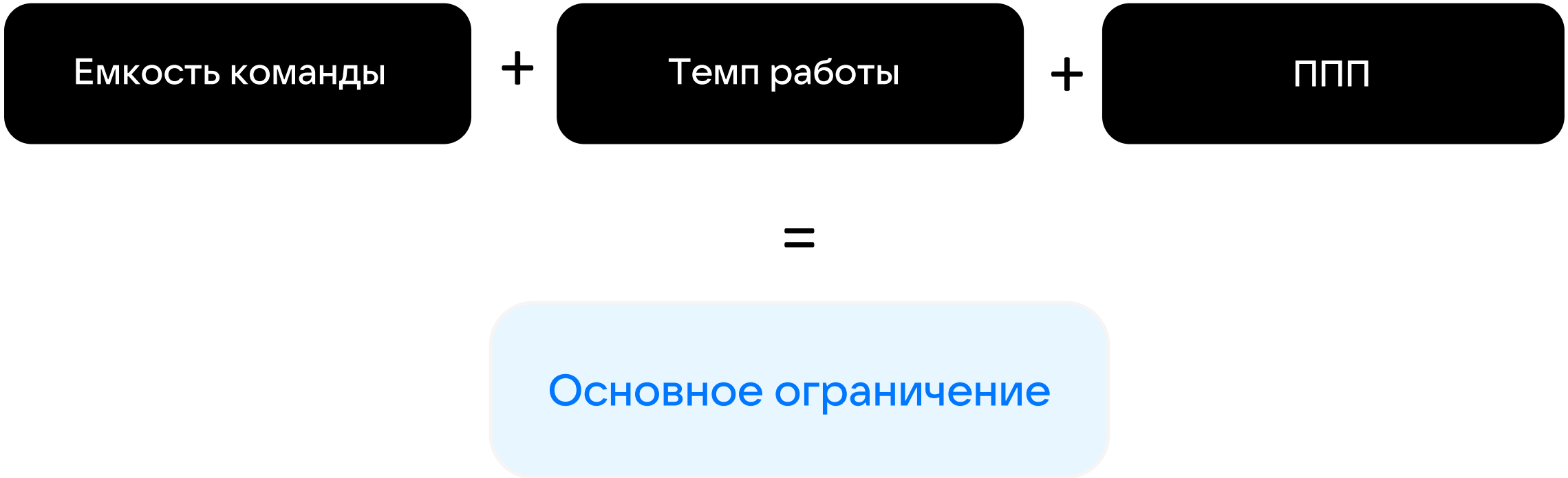
Разработчик

Влияние

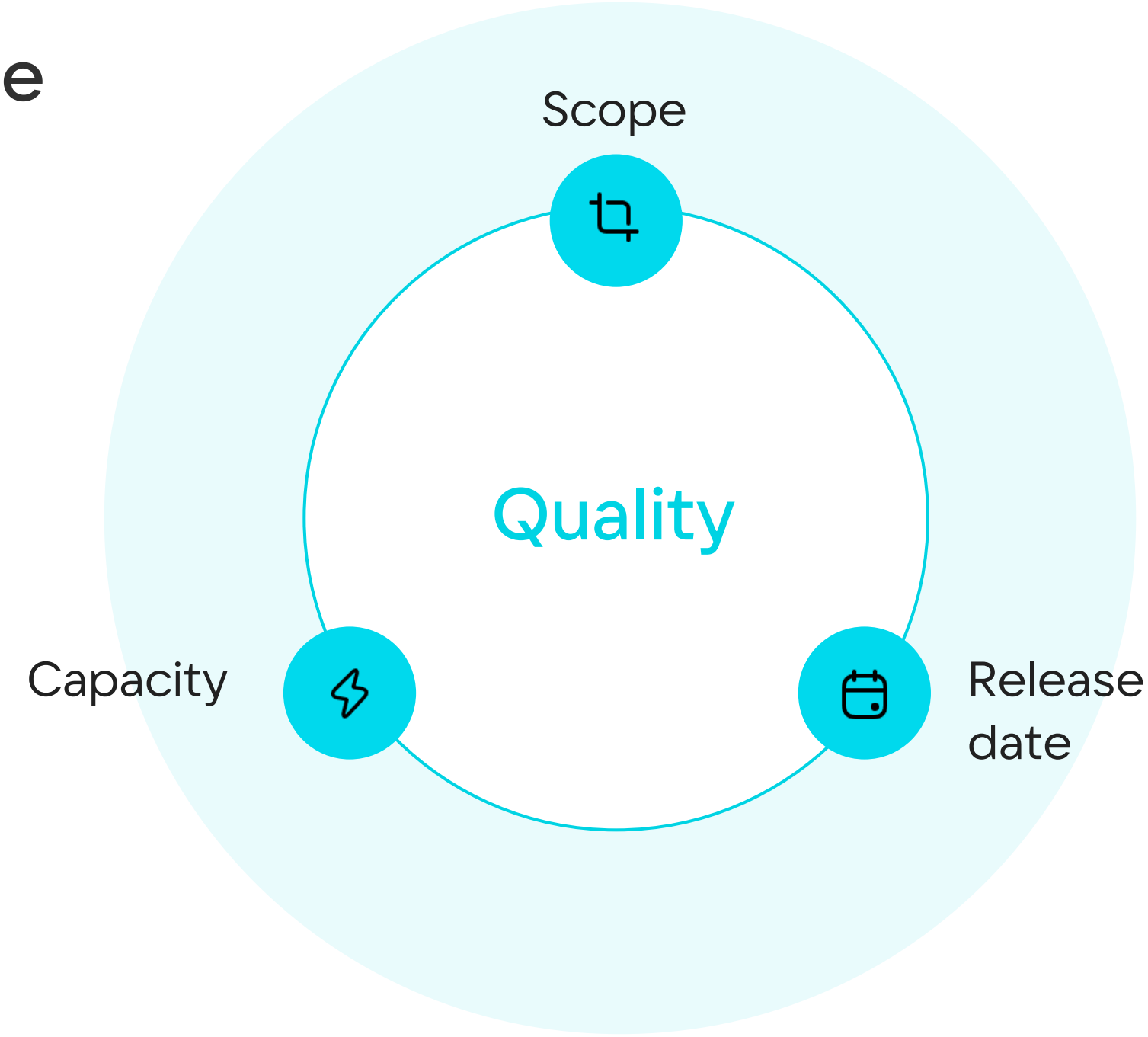
Для всех подготовительных этапов, разработчик – это источник ограничений	Для друг друга, мы - звенья цепи производства	Для всех этапов «после разработки», разработчик – это источник частей продукта
PM, Architect, Analyst...	Developers	QA, SRE, Sales

Разработчик

Планирование



Влияние

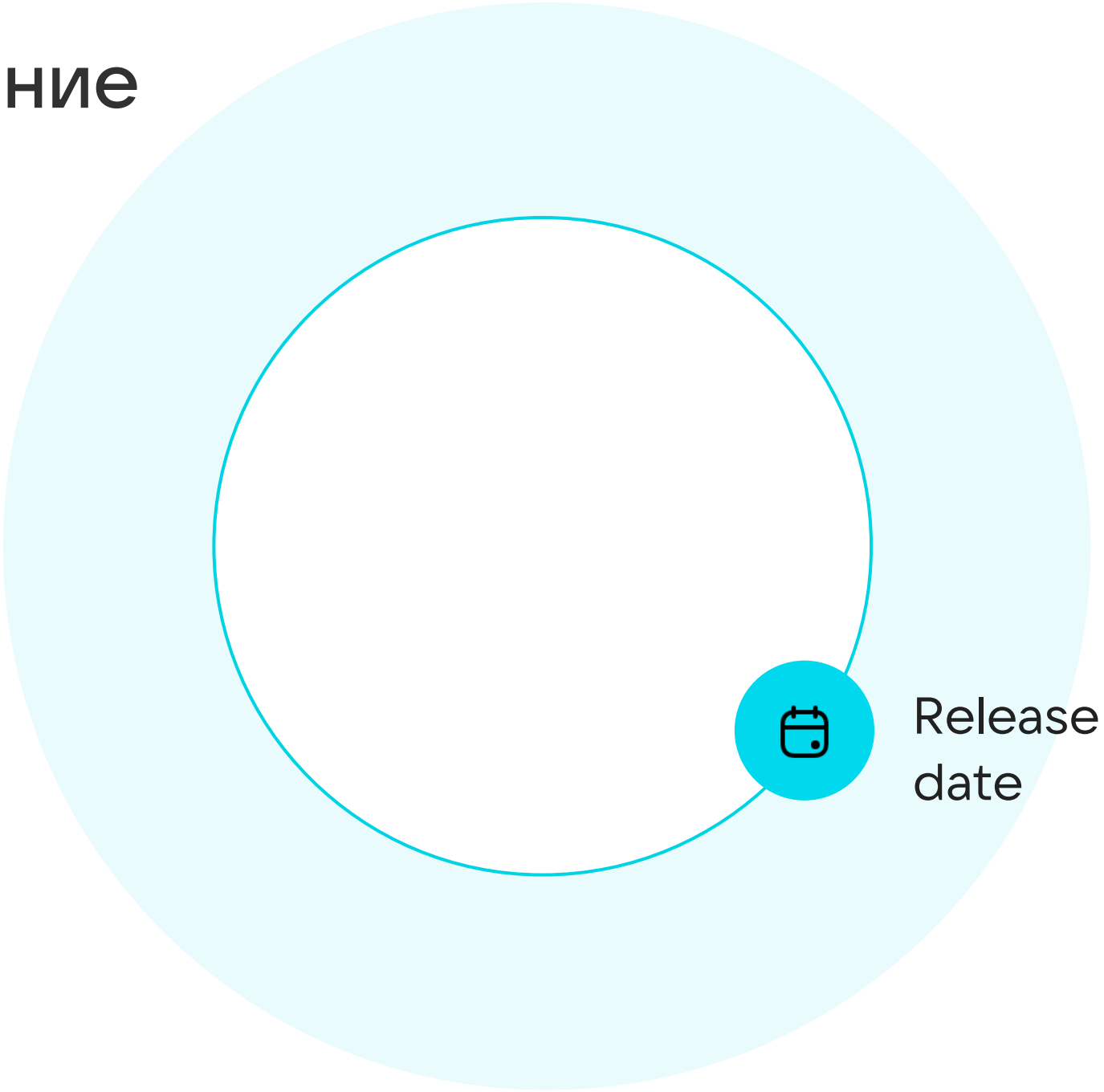


Влияние

Toolset

SDL - практики

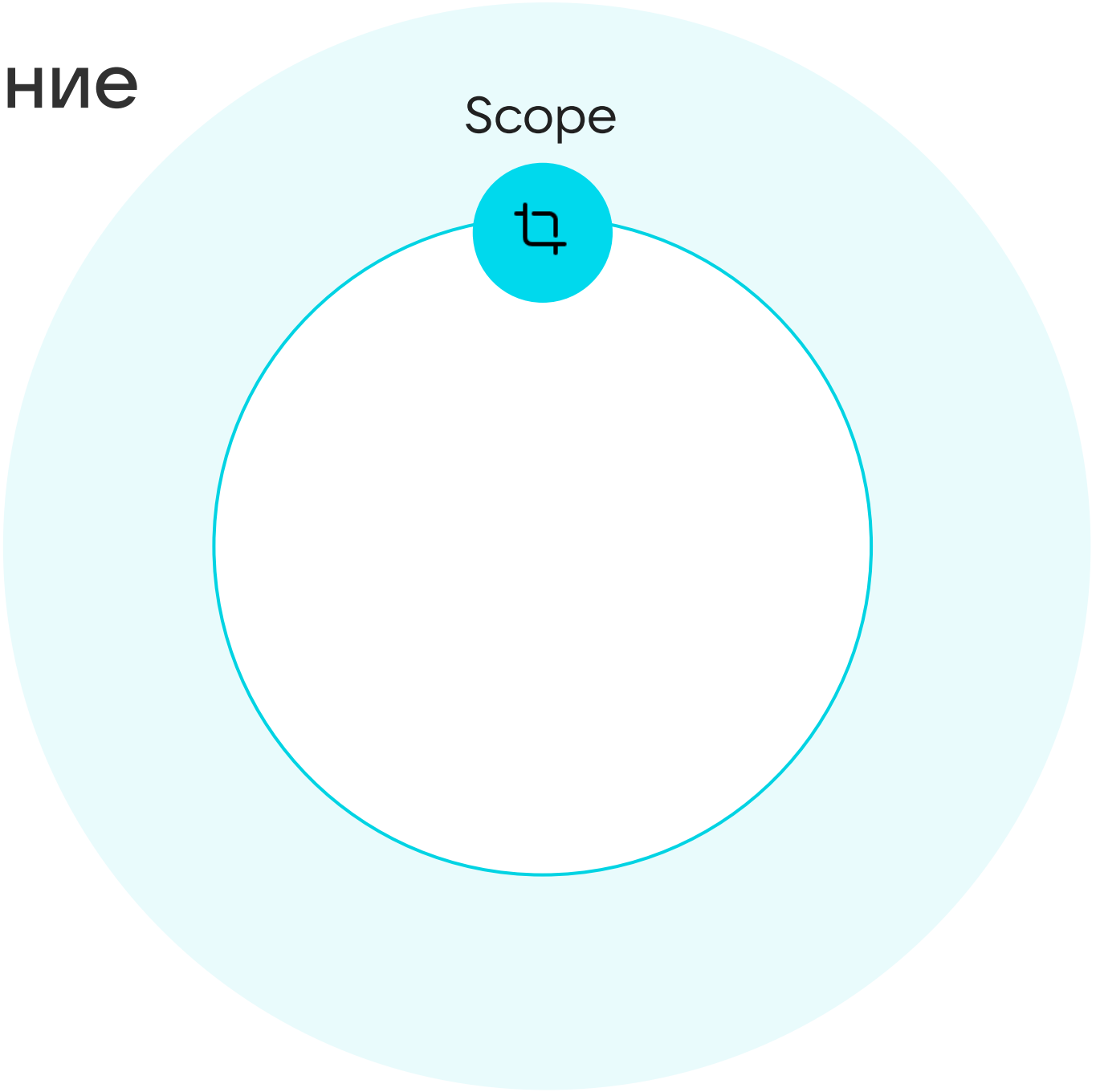
Влияние



Темп работы

Автоматизация

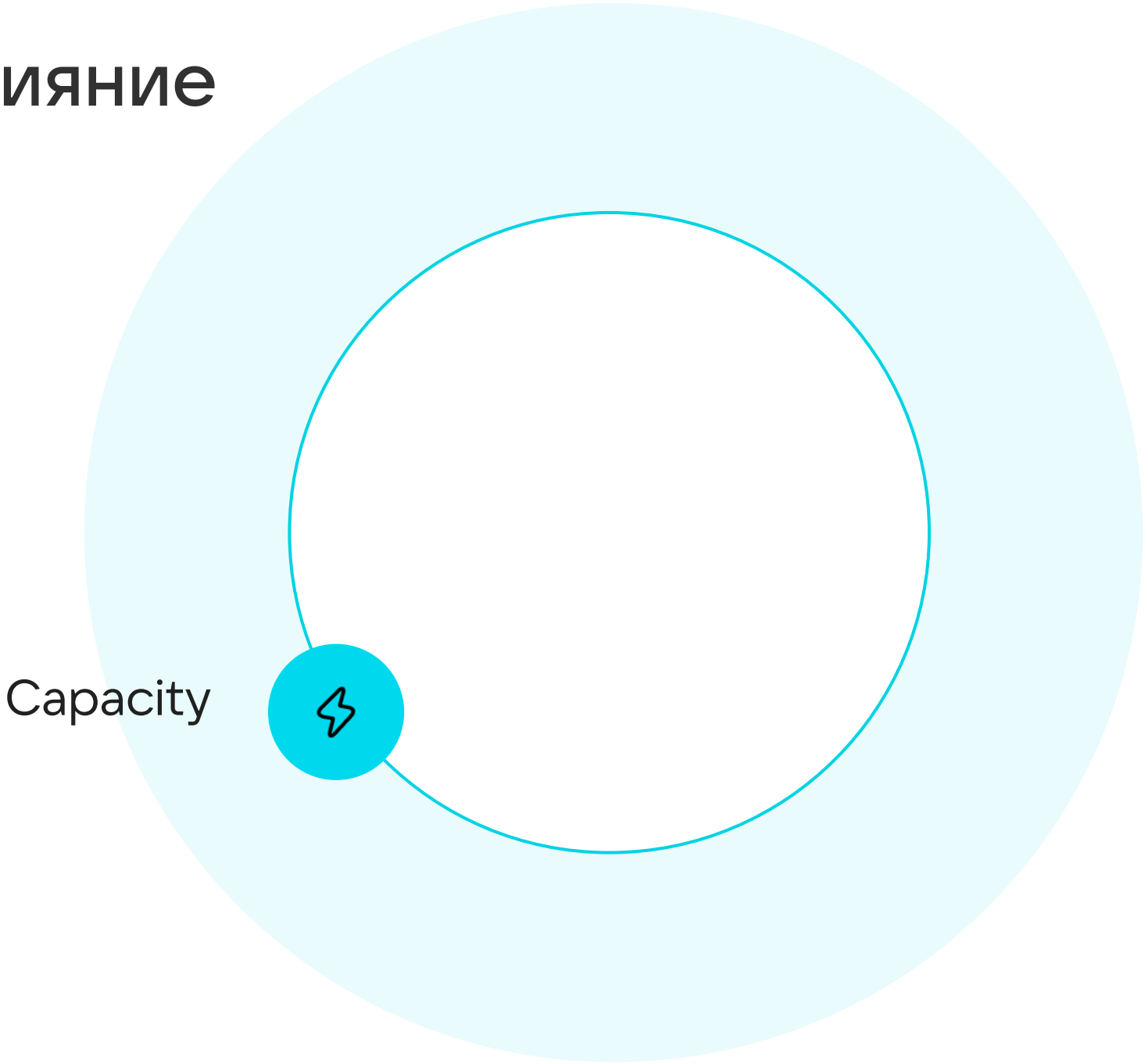
Влияние



Емкость команды

Зрелость продукта

Влияние



Емкость команды

Темп работы

Мнение

Команда

это станок производства

Влияние разработчика

это не прямое управление

```
constexpr int f() {
  if (std::is_constant_evaluated()) {
    return 5;
  } else {
    return 10;
  }
}
int main() {
  return f();
}
```

Техдолг, рефакторинг, улучшение перформанса – почему это не в почёте

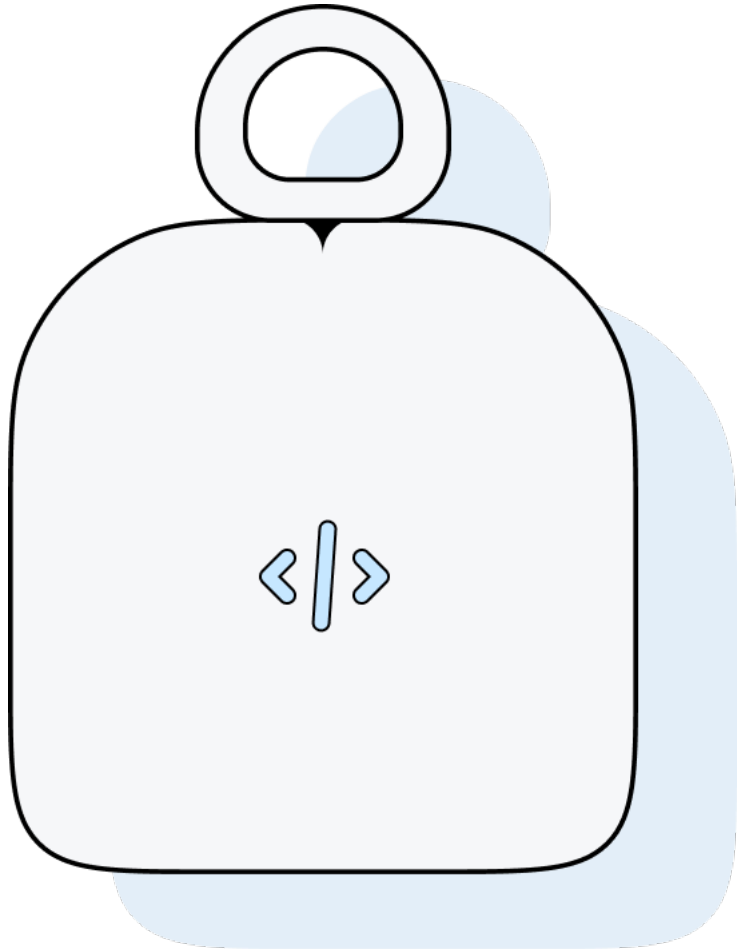
«Технический долг – это накопленные в программном продукте проблемы

Википедия

Такое тоже приводит к техдолгу

```
template <typename Float>
concept floating_point = std::is_floating_point_v<Float>;
auto add(const floating_point auto value1,
         const floating_point auto value2) {
      return value1 + value2 + 1.f;
}
template <typename Float,
typename = std::enable_if_t<std::is_floating_point_v<Float>>>
auto add(const Float value1, const Float value2) {
      return value1 + value2 + 2.f;
}
int main() {
      return add(1.2, 1.3f);
}
```

Технический долг
Откуда берётся?



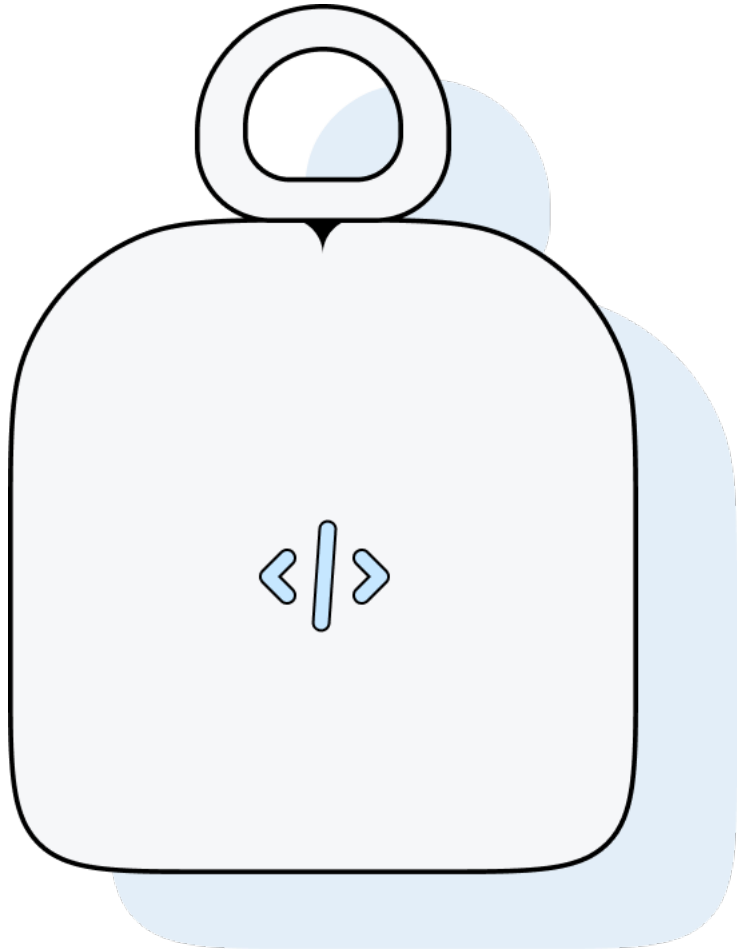
Косячим сами

- Коду «много лет»
- В команде «много» людей

Нам помогают накосячить

- Бизнес не дал времени на проработку
- PM не запланировал написание тестов/внедрение SDL практик

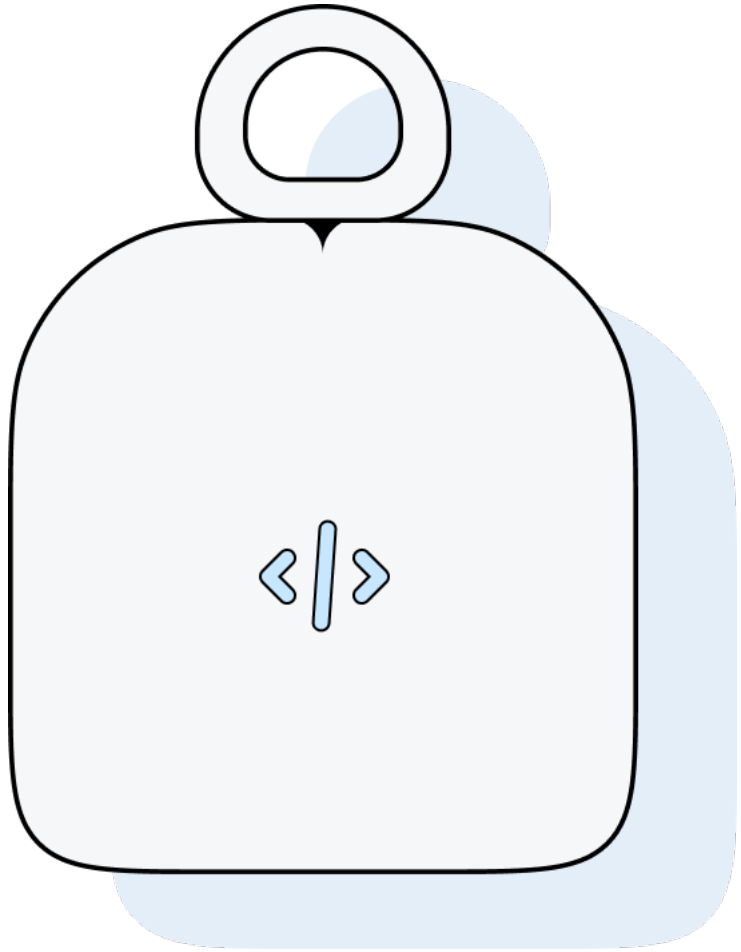
Технический долг
Сколько стоит?



Эффективная емкость команды
обратно пропорциональна размеру технического долга

Вероятность закрытия проекта
прямо пропорциональна размеру технического долга

Технический долг
Например



Команда

Емкость – 5dev
(6ч*5д*5dev = 150ч/ч в неделю)

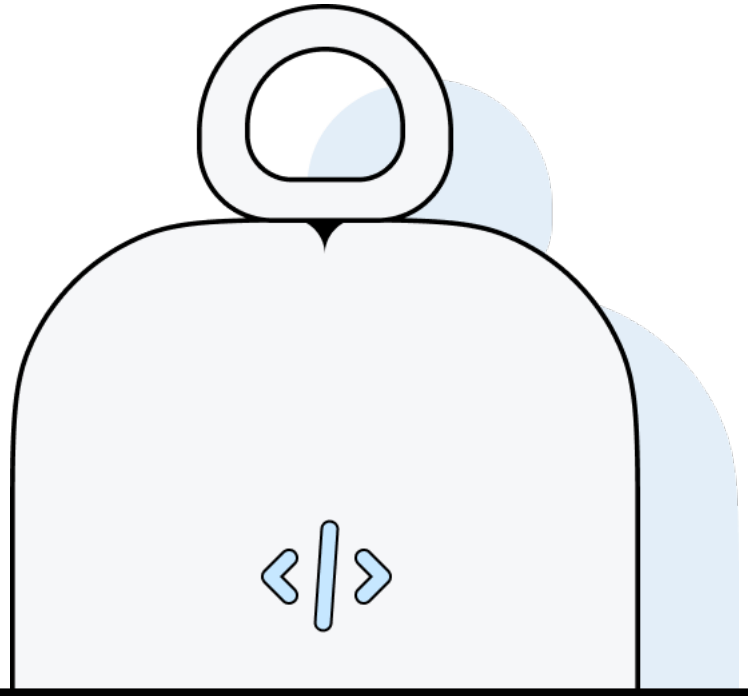
Эффективная емкость - ?

(105 или 45)

Технический долг

«Хороший» Фокус фактор – 0,7
«Плохой» Фф – 0,3

Технический долг
Рекомендация



Техдолг – это, в первую очередь,
проблемы проекта/продукта,
а не отдельного человека.

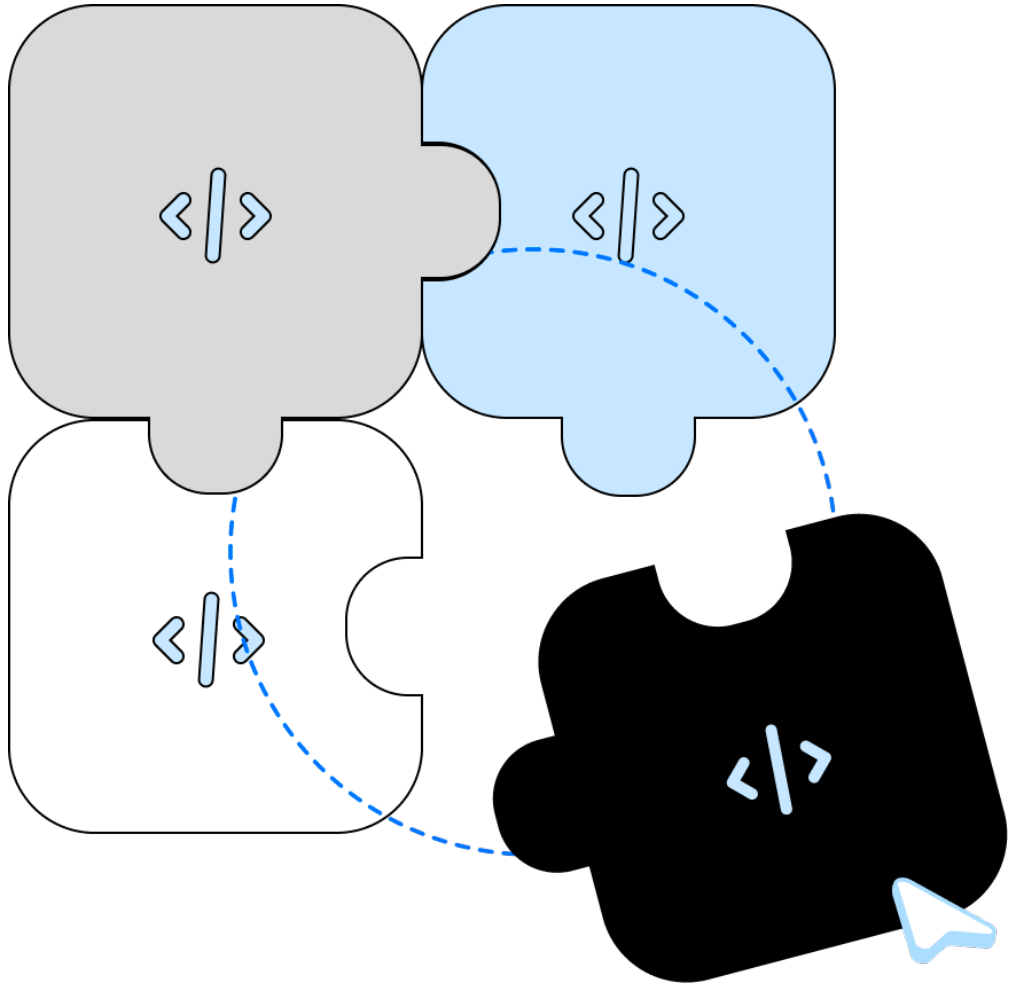
Оценка

должна включать технический долг

PM должен

- планировать работы по техдолгу
- быть «умным»

Рефакторинг



Рефакторинг обоснован, если...

~~...позволит выступить с докладом на C++Russia и получить Senior Developer~~

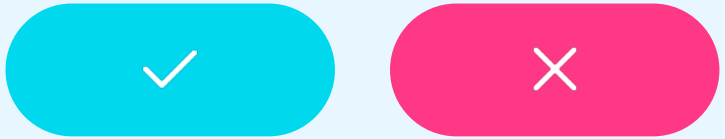
Стоимость значительно ниже затрат

Помимо исправления дефектов принесёт новые «фичи»

Обоснован с точки зрения бизнеса, а не технологий

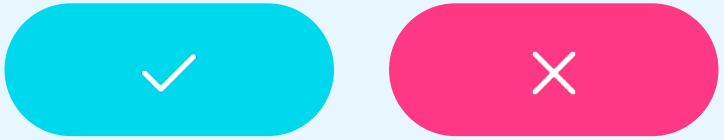
Рефакторинг
Например

Производительность можно повысить на n%



Approve
or
Decline

Покупка нового/модного инструмент поможет лучше «работать работу»



А стоит ли рефачить?

```
// Надо удалить элемент со значением 5
std::vector<int> vec = {1,2,5};
// 1. - ?
vec.erase(std::remove(begin(vec), end(vec), 5), end(vec));
// 2. - ?
std::erase(vec, 5);
// 3. - ?
vec.erase(begin(vec), end(vec), 5);
```

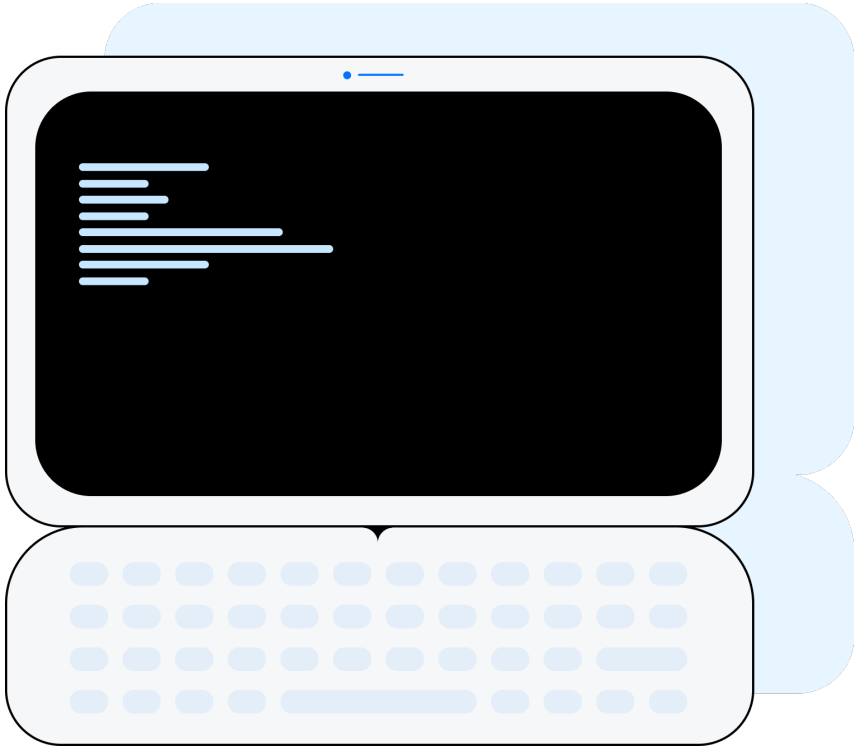
Мнение

Менеджер не ~~тупой~~ дилетант,
он считает PnL

Используйте бизнес-аргумены
для получения времени
на интересную работу

Особенности C++ разработки в выпуске продуктов и сервисов

Рынок труда
C++ разработчик



Трудно найти, легко потерять и ...

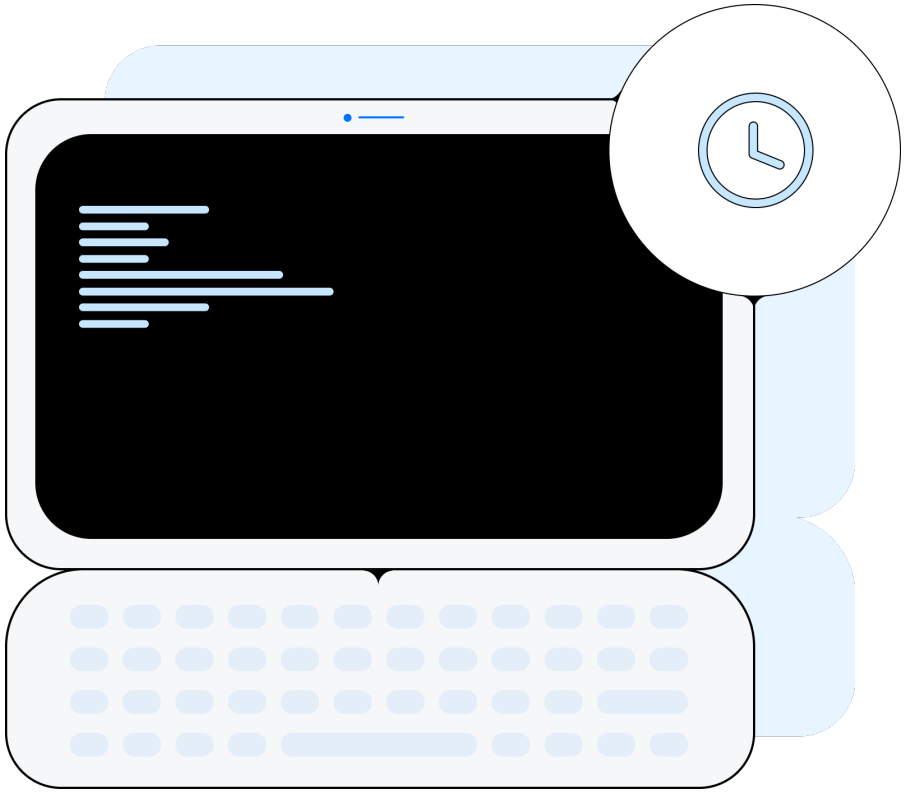
Очень большой разброс
в уровнях квалификации

Что влияет на «разброс» ЗП

Применимость ограничивается
Рынок растёт очень быстро,

C++ - это не про fast-dev,
а про fast-in-prod

Новый проект
C++ разработчик



Кода нам нужен именно он?

Всегда!

...

Реальные причины

- Так исторически сложилось
- Нужны системные инструменты/приложения
- Делаем новую железку
- Делаем крутой ML
- ...

C++ в производстве

C++ - хорошо известный инструмент

SDL практики стоят дорого

SAST

DAST

Fuzzing

C++ vs JavaScript

Соотношение работ ~ 3 к 1

C++ в производстве

C++ - хорошо известный инструмент

Сертификация

- Долго
- Дорого

C++ vs JavaScript

- C++ - практически 100% гарантия получения сертификата
- JS – всё «туманно»

Мнение

Использование C++ - вдумчивый, осознанный выбор

Вложения в «удешевление ошибки» окупаются

На этом все
Можно задавать вопросы

Ну и на последок...

```
template<typename Callable, typename ...Param>
auto bind_values(Callable callable, Param ...param) {
      return [my_callable = std::move(callable), ?my_param?](){
            return my_callable(my_param...);
      };
}
int add(int lhs, int rhs) {
      return lhs + rhs;
}
int main() {
      const auto bound = bind_values(add, 1, 2);
      return bound();
}
```

tech

C++ Russia 2023

Бычук Александр
Director of Engineering

@bas524

Спасибо за внимание

бизнес