

AI-DRIVEN

Product Development Life Cycle

Новая эра разработки цифровых
продуктов

Владимир Долгов
Управляющий директор
Сбер

The Truth Is Out There

Истина где-то рядом

SE 3.0 – следующий фундаментальный сдвиг в разработке ПО

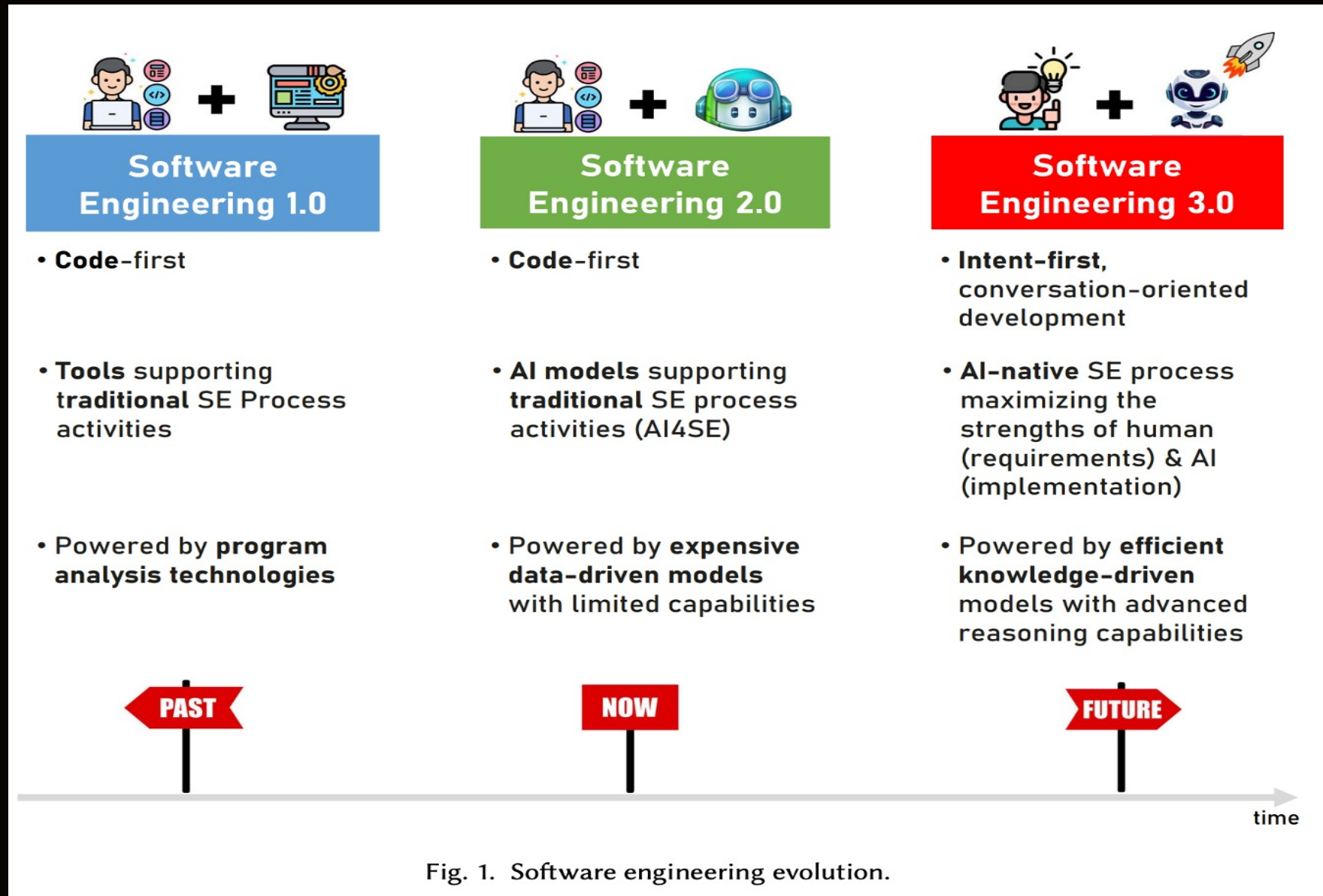


Fig. 1. Software engineering evolution.

SDLC 3.0 - партнерство между AI и разработчиком от работы с требованиями до поддержки решения

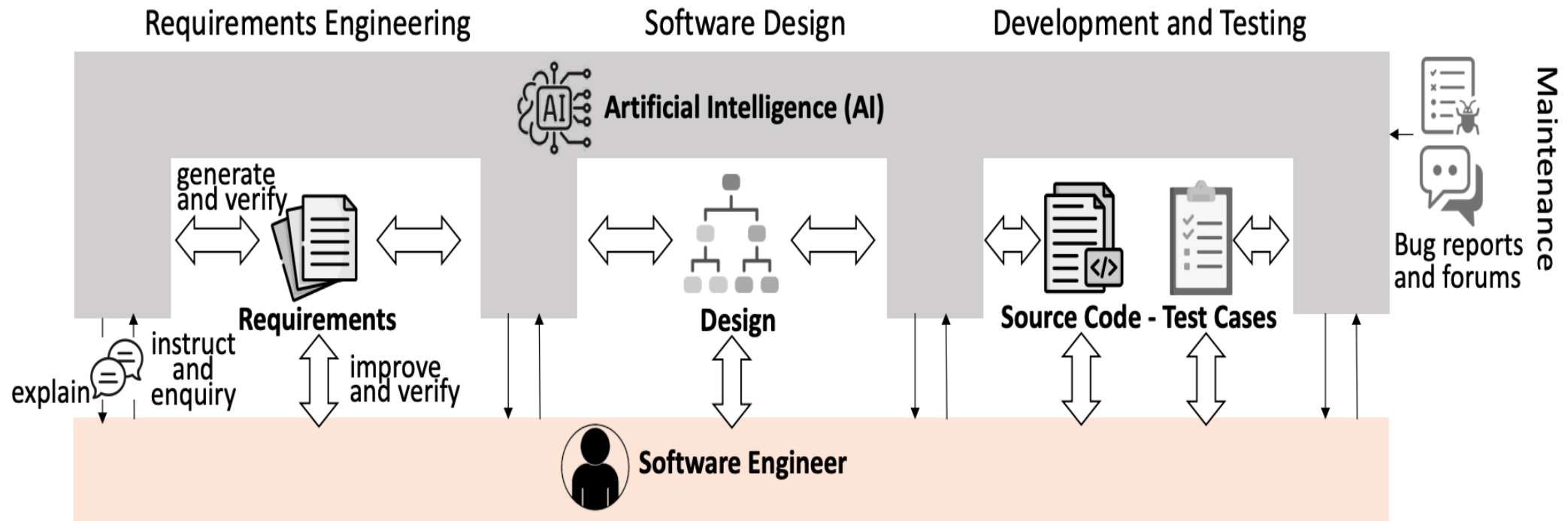


Figure 1: Logical architecture of the envisioned future symbiosis of Software Engineers and AI

Изменение подхода к созданию новой функциональности

Code-centric development loop

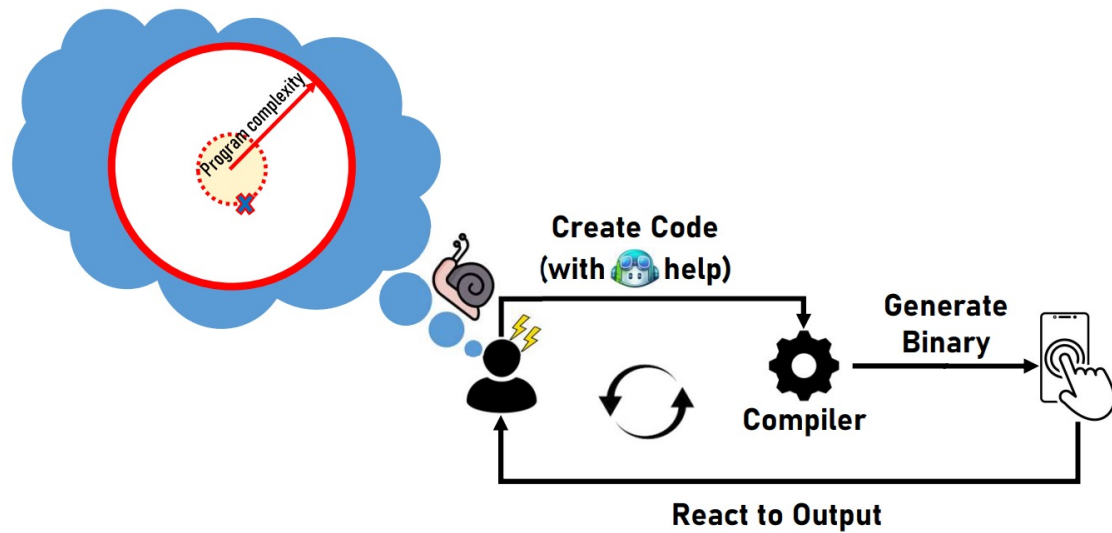


Fig. 2. Code-centric development in SE 2.0.

Intent-centric development loop (AI-first)

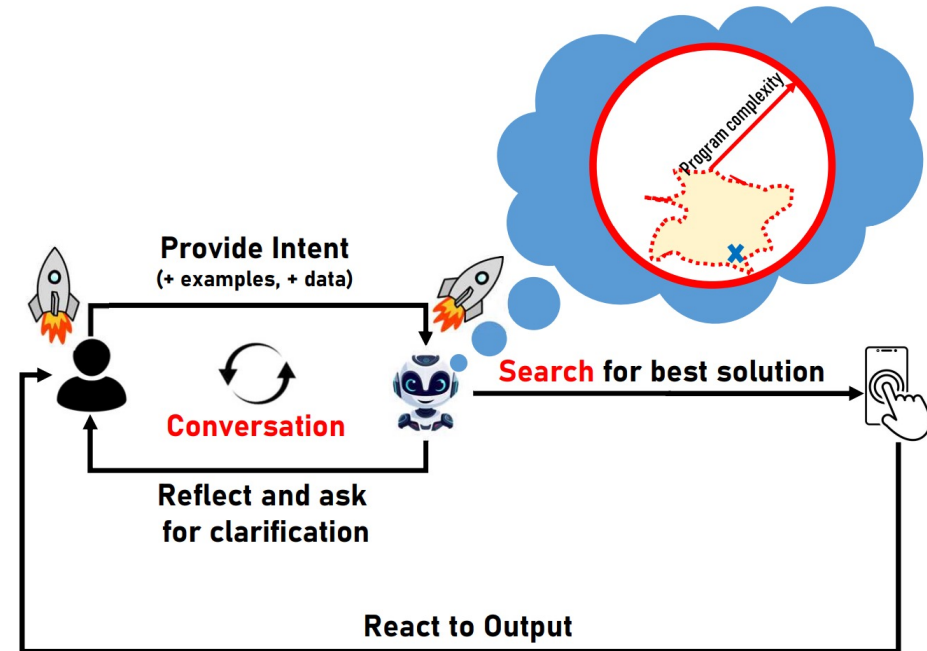
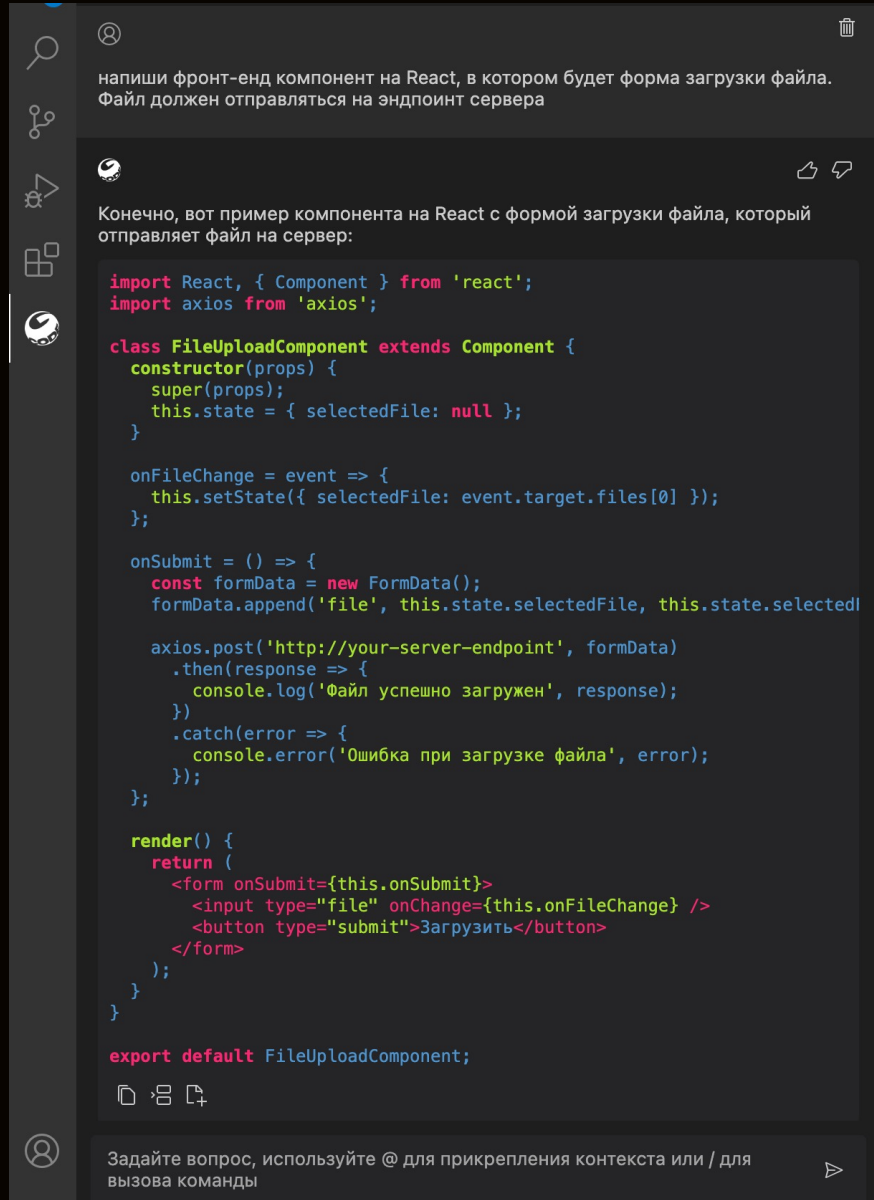


Fig. 4. Intent-first, conversation-oriented development in SE 3.0.

“ИИ не вводит новый тип мышления. Он показывает, что на самом деле требует мышления»



...значение слова

«программирование» изменится и будет означать процесс организации идей в исполняемые программы.

<<Компьютерная наука - это систематическое мышление, а не написание кода>>

<https://www.oreilly.com/library/view/ai-engineering/9781098166298/>

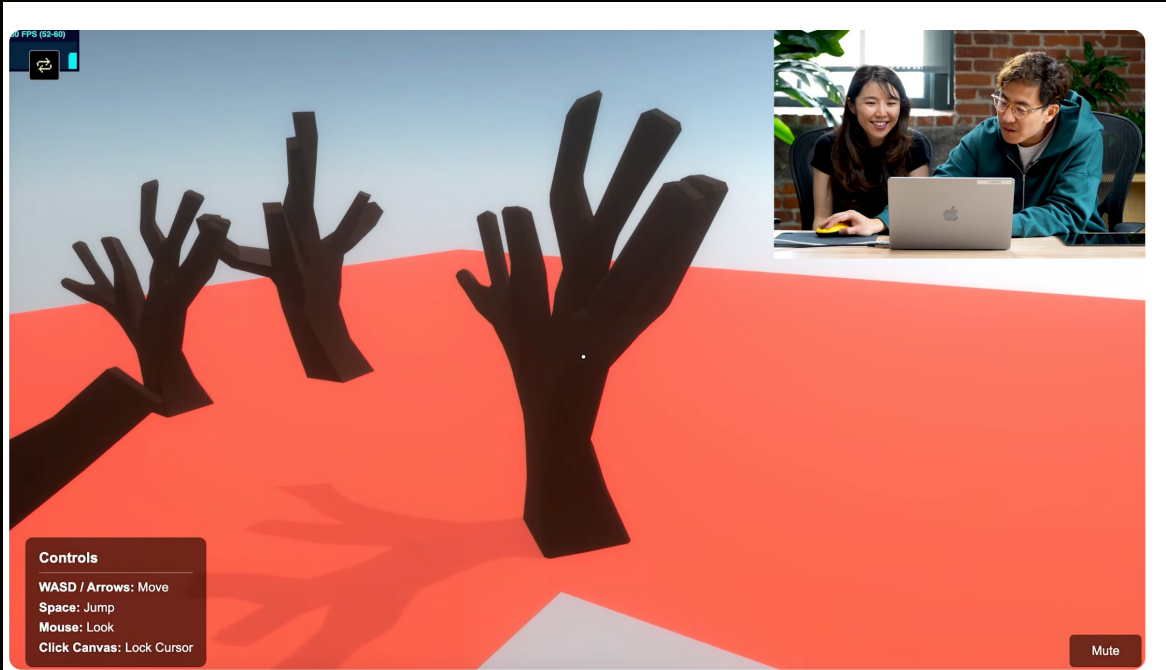
Vibe Coding

первые попытки реализации SDLC 3.0

От любви до ненависти - один шаг




Vibe Coding a 3D Flight Simulator in 60 SECONDS With ONE PROMPT!



Create A Game in Hours, Not Years - Vibe Coding Explained


RIP logic



6:35

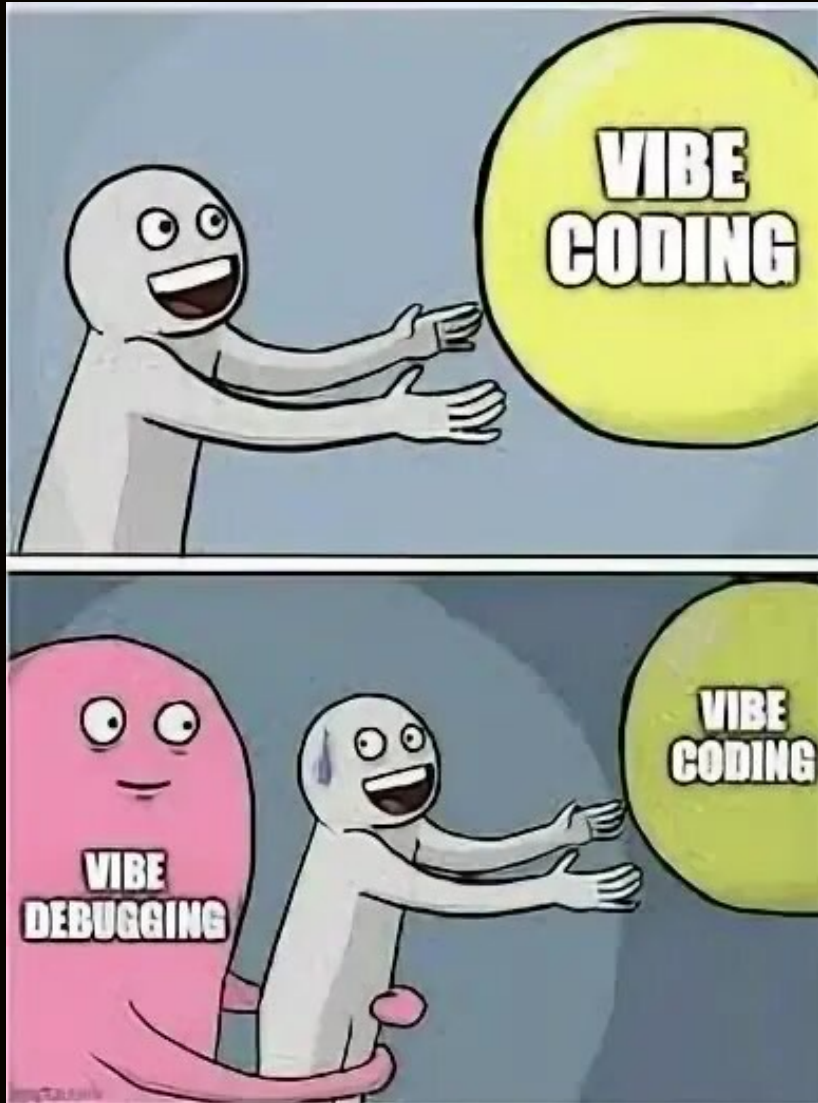
Vibe Coding is Getting Out of Hand

78 тыс. просмотров • 1 месяц назад

 Sajid

AI is great and all, but this Mindless vibe coding is not h

На текущем этапе развития vibe coding имеет ряд преимуществ и недостатков



- Доступность для нетехнических пользователей
 - Быстрое создание прототипов и идей
 - Снижение порога входа в программирование
 - Экономия времени на рутинных задачах
-
- Риск генерации некачественного или уязвимого кода
 - Потеря глубокого понимания архитектуры и логики программ
 - Сложности с масштабированием и поддержкой крупных проектов
 - Снижение критического мышления и навыков у разработчиков

Эволюция подходов к разработке



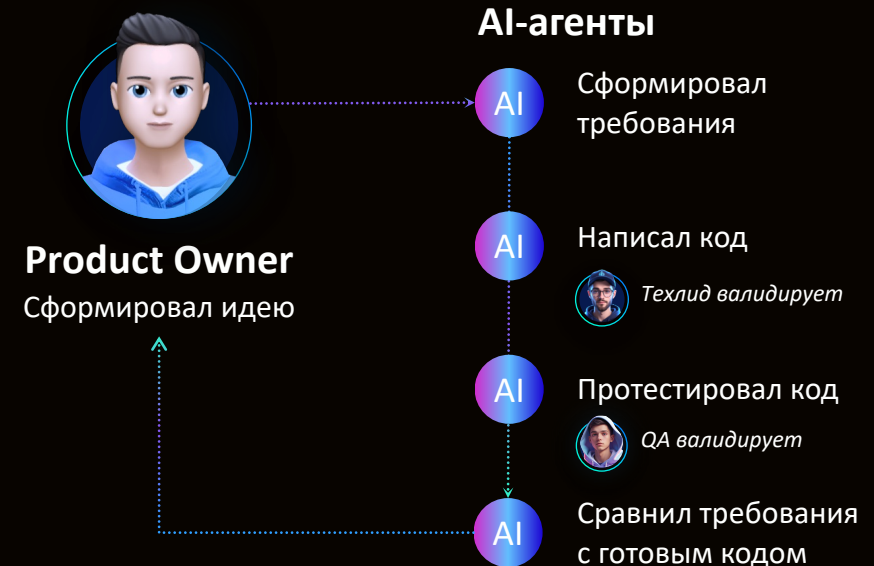
Новые паттерны состава и структуры команд

AI assisted (подход «Ремесленный»)



Этот подход наиболее подходит для уникальных, инновационных, нестандартных и сложных задач, связанных с созданием новых продуктов на инновационных технологиях, в том числе создание агентов

AI driven (подход «Фабрика»)



Этот подход лучше всего подходит для решения шаблонных задач в рамках корпоративных технологий, таких как разработка стандартных интерфейсов, исправление типовых ошибок или миграция кода на новые стеки

AI может поменять не только структуру AGILE команды, но и требования к компетенциям



Команда «ромб»

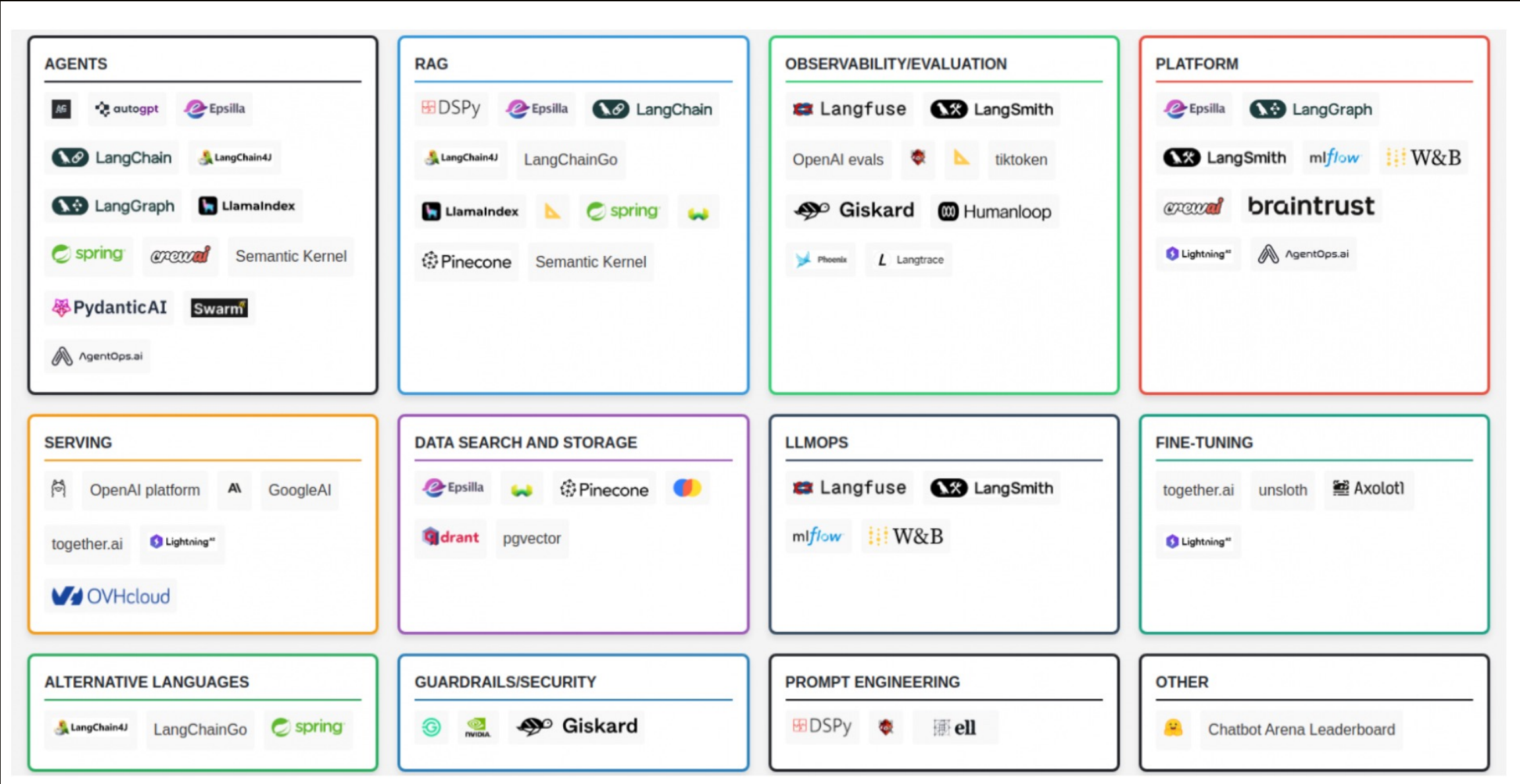
Строится вокруг разработчиков так как это наиболее дорогой и «медленный» ресурс



Команда «песочные часы»

Создание кода становится все более легким и надо больше усилий по правильной постановке задачи и проверке полученных результатов

Новые технологии уже собираются в новый стек разработки



Что нас ждет в ближайшие годы?



Рост ИИ-генерируемого
кода

До 90–100% в некоторых задачах

Трансформация IT-
профессий

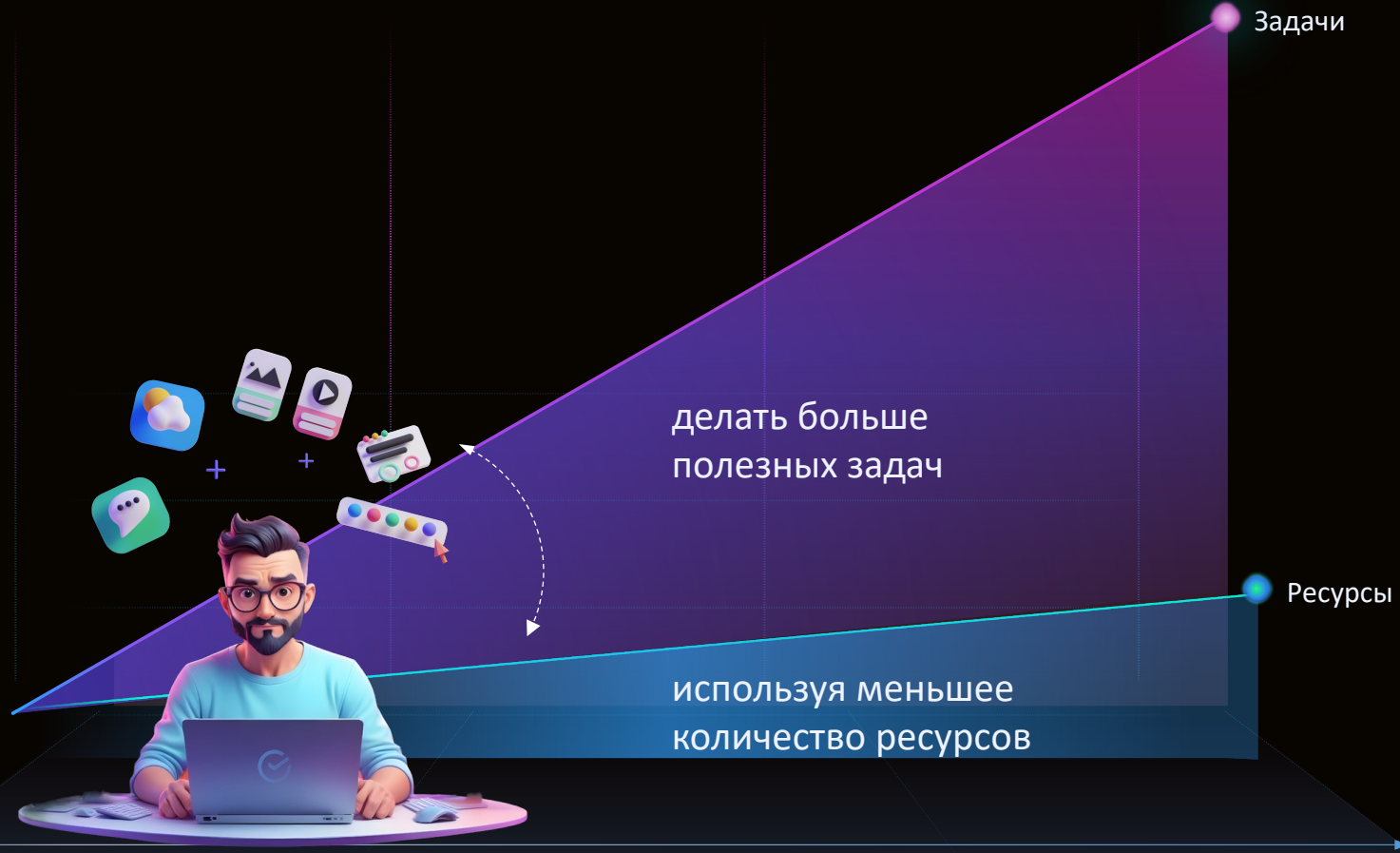
Масштабные изменения в индустрии

Новая культура и парадигмы
управления

Управление изменениями, контроль
качества

**От общего к частному:
как мы двигались и какие
уроки извлекли**

Текущий вызов – значительно ускорить темп роста не наращивая численность



+30%

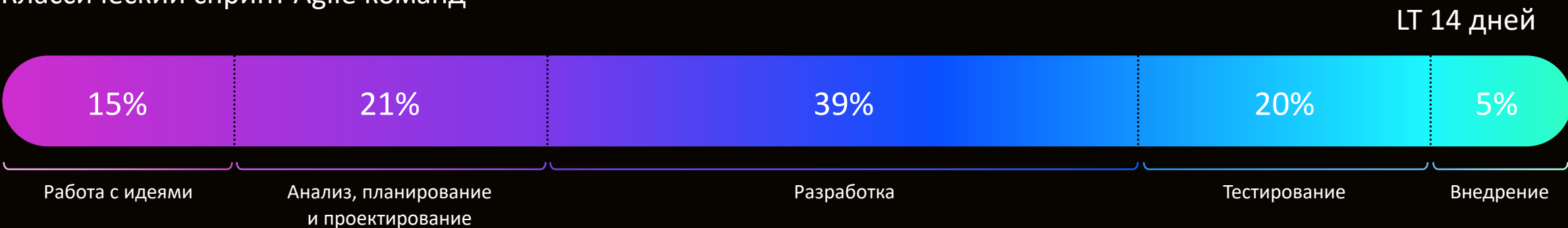
рост количества
задач на команду

Соблюдая:

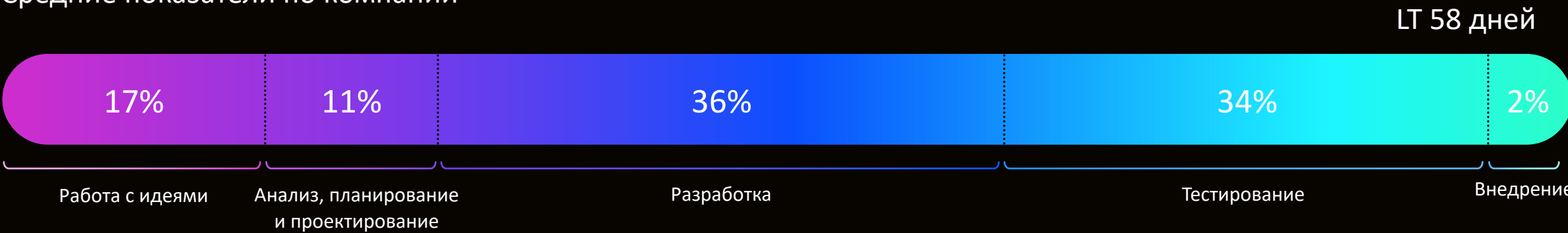
- Технологические ценности безопасности и надежности
- Требования регуляторов и государства

Провели аналитику текущего PDLC в компании

Классический спринт Agile команд

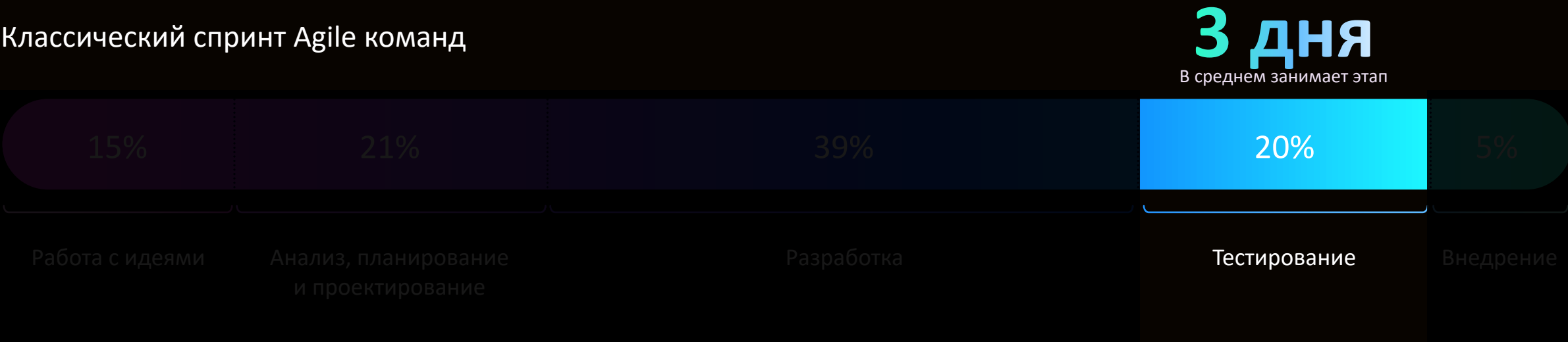


Средние показатели по компании

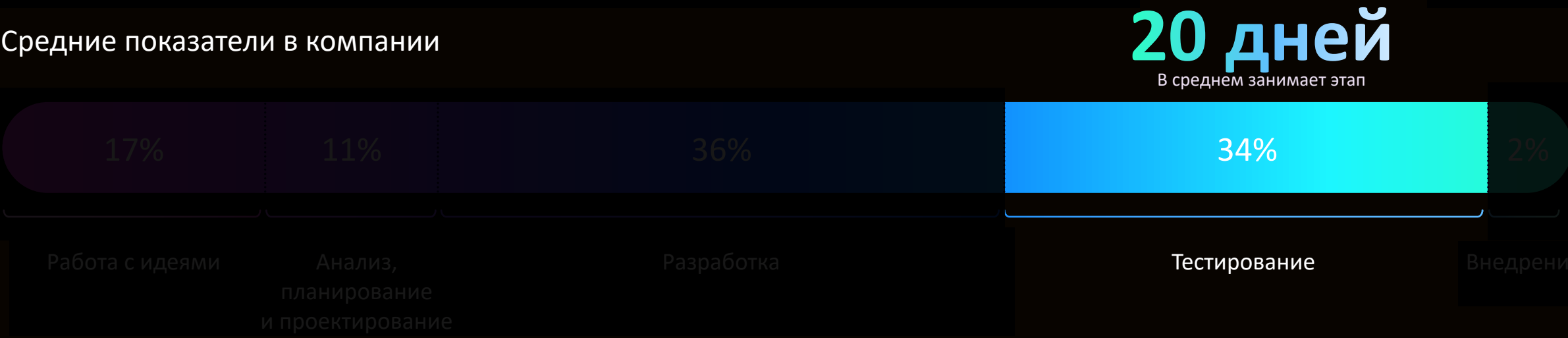


Один из самых проблемных этапов — «Тестирование»

Классический спринт Agile команд

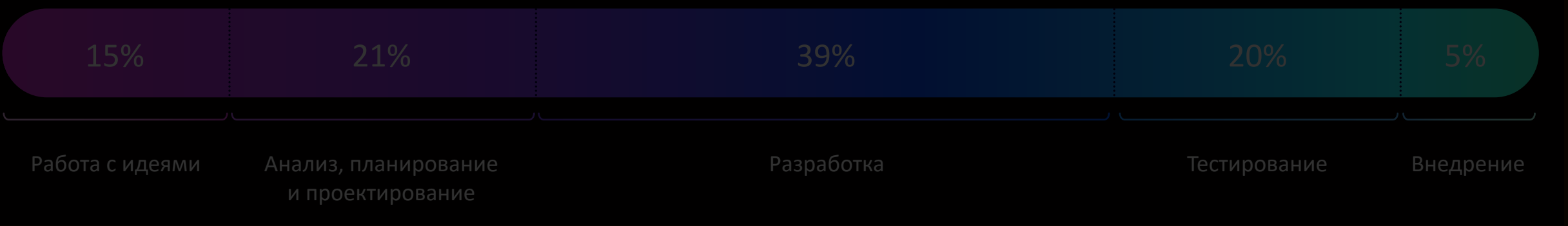


Средние показатели в компании

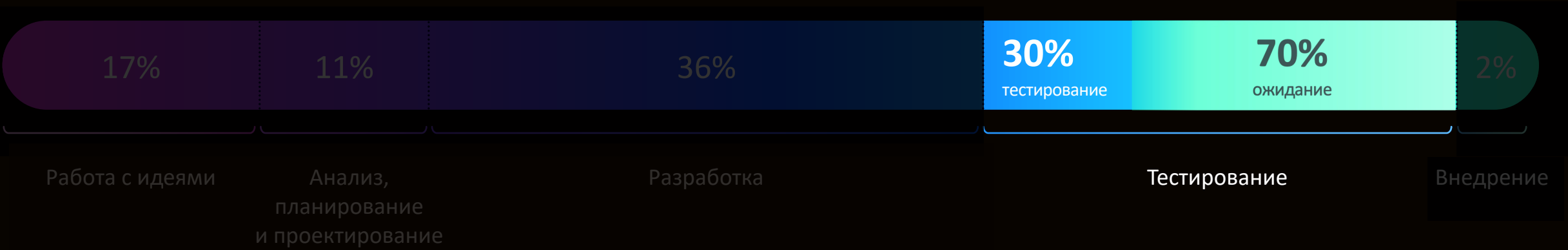


Длительное ожидание командами смежников

Классический спринт Agile команд



Средние показатели в компании



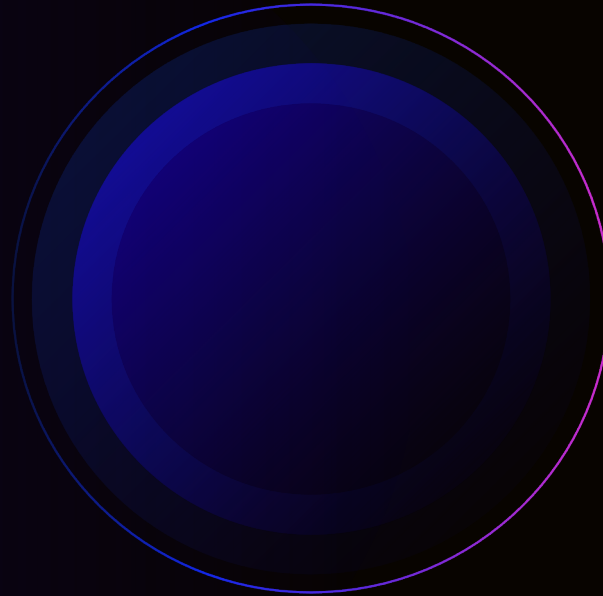
35%

задач в ПРОМ — это
исправление дефектов

Основные причины проблем на этапе «Тестирование».

Причины:

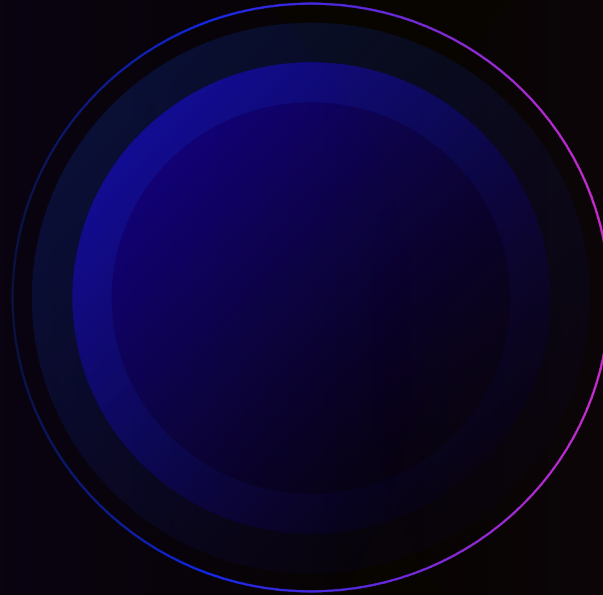
- Недостаточное качество требований
- Некачественное проектирование
- Отклонения в реализации от требования
- Недостаточное покрытие unit-тестами
- Не выполнение практики CodeReview
- Не полные или не точные планы тестирования
- Большая доля ручных тестировщиков
- Недостаточный уровень автоматизации тестирования (30%)



Disrupt этапа «Тестирование»

Причины:

- Недостаточное качество требований
- Некачественное проектирование
- Отклонения в реализации от требования
- Недостаточное покрытие unit-тестами
- Не выполнение практики CodeReview
- Не полные или не точные планы тестирования
- Большая доля ручных тестировщиков
- Недостаточный уровень автоматизации тестирования (30%)



Кандидаты на AI ассистентов

- AI декомпозиция требований
- AI оценка реализуемости требований
- AI ассистент планирования и распределения задач
- AI генерация unit-тестов
- AI CodeReview
- AI проверка реализации кода на соответствие требованиям (Code2Rec)
- AI генерация плана тестирования
- AI генерация регрессионных тестов

Технологические задачи занимают значительную долю бэклогов

Бенчмарк

75% Развитие функционала

Реализация функциональности

25% Технологические задачи

Миграция на новые версии библиотек

Оптимизация приложений

Оптимизация логирования

Оптимизация тестовой модели

Оптимизация регрессионных тестов

Миграция на новый тех. стэк

Компания

60% Развитие функционала

Реализация функциональности

40% Технологические задачи

ЗОКИИ

L2P

Вендорозамещение

Миграция на новые версии библиотек

Оптимизация приложений

Оптимизация логирования

Оптимизация тестовой модели

Оптимизация регрессионных тестов

Миграция на новый тех. стэк

Проблемы с ресурсами на технологические задачи

Причины:

Технологические задачи требуют высокой экспертизы в архитектуре и опыт работы с высоконагруженными системами.

На текущий момент
20% не имеют
TechLead в команде

50% имеют одного
TechLead на команду

т.е. не обладают достаточными ресурсами
для оптимизации производительности систем



Кандидаты на AI ассистенты

AI ассистент по оптимизации
Solution-архитектуры приложения

AI оптимизации объема
логирования кода проекта

AI оптимизация характера
обеспечения транзакционности кода

AI ассистент по переходу
на новые версии языков

AI ассистент по переходу
на новые версии библиотек

AI оптимизация модульной
структуры кода проекта

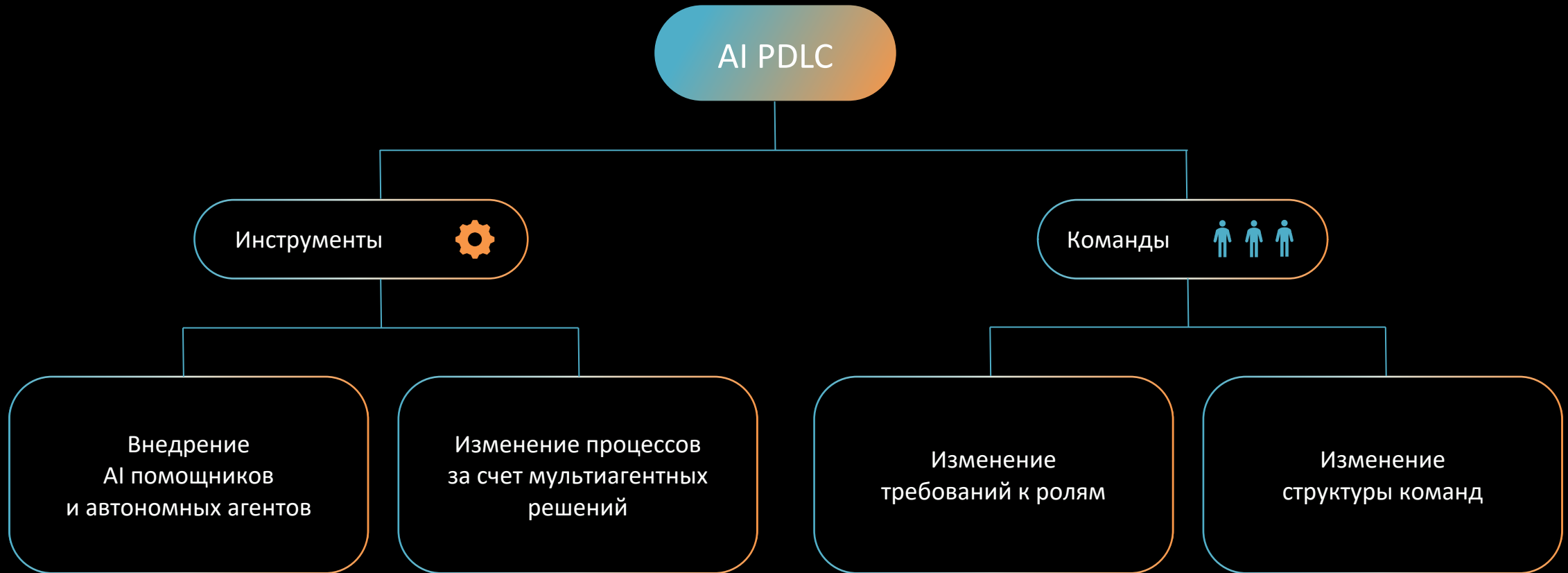
Начали делать ...

... получили ценный опыт

Урок 1. Стараться двигаться системно и избегать излишнего энтузиазма



Что должно быть в фокусе при внедрении AI



Работа с людьми

Правильные подходы
к обучению

Учиться применять
AI должны все

Найти правильную
позитивную мотивацию



Нужно быть очень аккуратным со всеми типами мотивации

Отрицательная мотивация очень сильно мешает нативно продвигать продукты.

Но осторожно с экстремально положительной мотивацией, особенно хакатонами.

Потом надо разбирать что делать с их результатами.



AI становится членом команды. Новые подходы к созданию клиентских путей

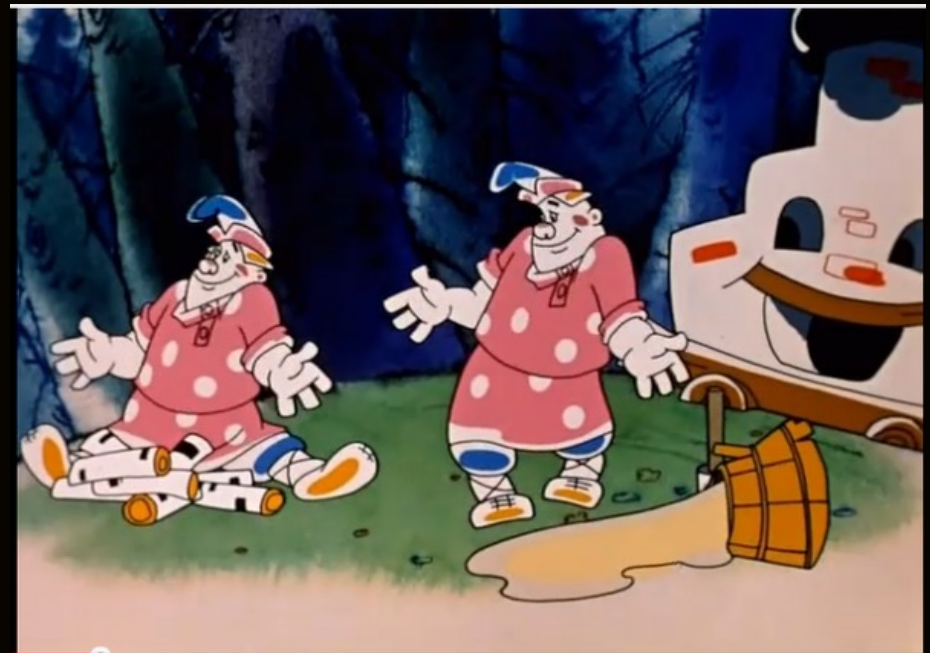
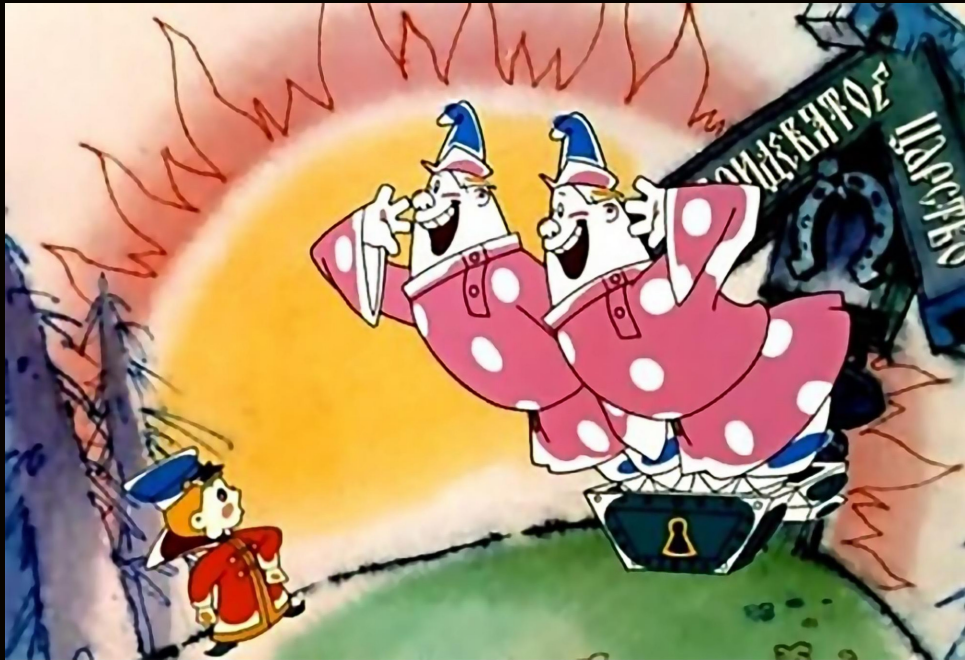
Дополнительно к UX и DevX появляется AX (Agent Experience)

Еще более повышается роль интегрированности инструментов для более эффективного взаимодействия агентов и людей на разных этапах

Должна появиться коллаборативная среда в которой будет удобно общаться людям и агентам



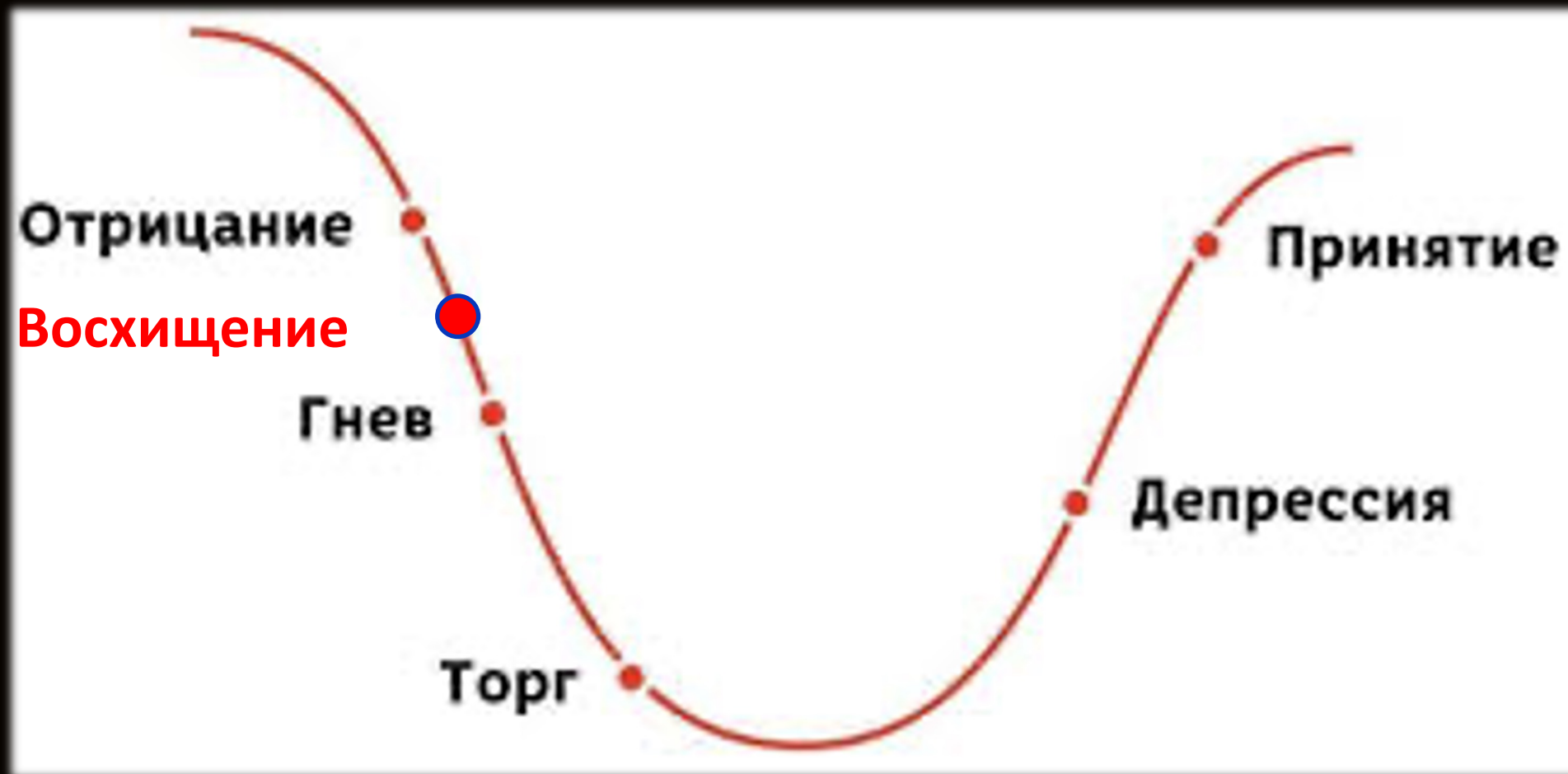
Урок 2. Не все так просто, как показывают в презентациях



Как мы начинали: варианты использования

Тип	Пример
Чат (ответы на вопросы, помощь в решении задач, беседы для развлечения)	Я готовлю питч. Задавай мне каверзные вопросы
Генерация текста	Создай план проекта с такими вводными:...
Классификация	Распредели обязанности арендодателя арендатора в два разных столбца
Открытый вопрос	Что такое хранение персональных данных? Почему компания хранит их даже после отзыва на обработку?
Закрытый вопрос	Какой методов диагностики наиболее эффективен для выявления сердечно-сосудистых заболеваний? 1. Электрокардиография (ЭКГ) 2. Ультразвуковое исследование (УЗИ) 3. Рентгенография
Мозговой штурм (дискуссионные вопросы, не имеющие точного ответа)	Где и как можно использовать генеративную модель в сегменте B2B?
Извлечение	Вот текст договора, напиши полное наименование заказчика, ИНН и ОГРН из текста
Обобщение (краткая сводка, анализ данных)	Вот инструкция. Что должен сделать сотрудник техподдержки в случае утечки персональных данных?
Переформулирование текста (изменение стиля и тональности, упрощение)	Вот сообщение. Перепиши его кратко и в деловом стиле

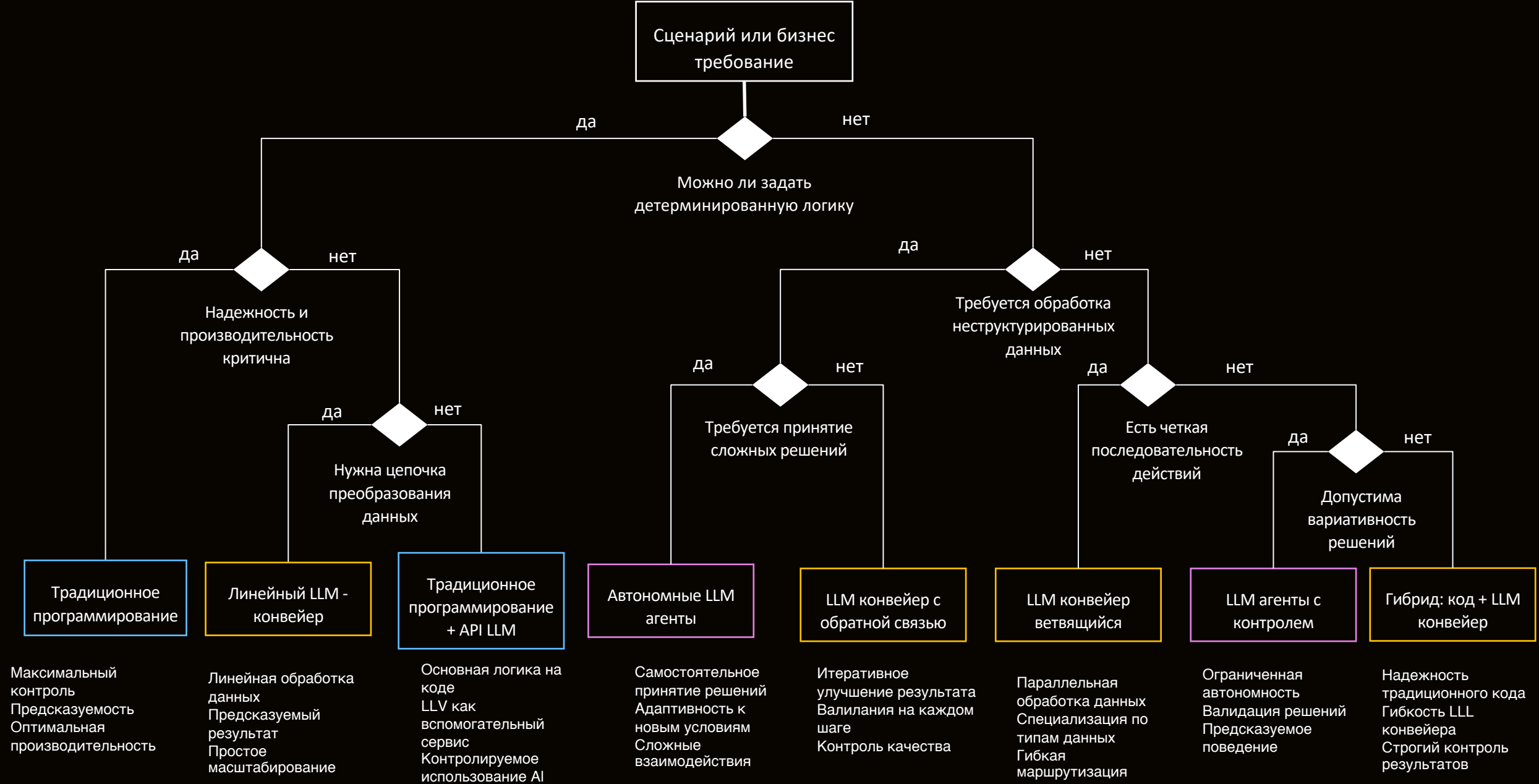
Погружение в AI. 5 (6) стадий принятия неизбежного



Мой кодекс для начинающего

- Не надо бояться недостатка знаний
- Нужно уметь быть фантазером
- Нужно быть готовым к неудачам
- Нужно быть готовым непрерывно учиться
- **Придется перейти от постановки требований к проверке гипотез**

Как выглядит сейчас: варианты использования



Урок 3. Для каждого решения надо понять, что такое хорошо и что такое плохо

*Что такое
хорошо
и что такое
плохо*



Важно понять, как выбудете принимать решение, что продукт работает достаточно хорошо, чтобы его использовать широко

1. Выберите метрику или несколько метрик качества решения
2. Определите пороги хорошего результата по каждой метрике
3. Подготовьте тестовые корзины

На примере продукта по оценке производительности разработчиков:

- Выбрали метрики точность оценки производительности и качества разработчиков
- Провели исследование и выяснили, что разработчики дают оценки с точностью 80% с разбросом +-20%. Взяли это как достаточную точность для модели.
- Подготовили тестовые корзины по каждому языку для проверки результатов работы модели

Team/Developer Productivity		Date																								
Team	2023						2024												2025							
	July	Augu.	Septe.	Octo.	Nov.	Dece.	Janua.	Febru.	March	April	May	June	July	Augu.	Septe.	Octo.	Nov.	Dece.	Janua.	Febru.	March	April	May	June	Q3 July	
Team 1	55%	57%	80%	63%	49%	30%	37%	77%	45%	37%	31%	36%	86%	35%	77%	52%	42%	70%	29%	9%		40%	68%	17%		
Team 2				10%										10%		43%			25%		33%	38%	13%	50%		
Team 3	6%	81%	36%		5%	14%	18%		40%	55%	13%	96%	11%	4%	45%		21%	69%	25%	24%	38%	5%	10%	42%	9%	
Team 4	68%	64%	81%	71%	53%	65%	69%	48%	58%	44%	51%	66%	51%	42%	62%	71%	67%	37%	53%	66%	65%	35%	65%	82%	100%	
Team 5	72%	27%	24%	45%	40%	38%	51%	35%	67%	48%	26%	72%	33%	44%	37%	48%	58%	36%	50%	42%	42%	30%	65%	70%	100%	
Team 6	56%	50%	96%	61%	73%	54%	49%	74%				79%	83%	27%		67%	34%	60%	87%	7%	29%	83%	9%	40%	51%	4%
Team 7							54%	56%	75%	65%	39%	86%	47%	54%	72%	53%	54%	28%	46%	71%	72%	76%	36%	63%		
Team 8	64%	68%	64%	78%	63%	72%	39%	55%	62%	62%	45%	46%	53%	49%	75%	50%	43%	27%	33%	41%	63%	53%	66%	75%	76%	
Team 9	84%	55%	96%	77%	77%	40%	95%	4%		14%		6%	12%		53%	29%					17%			84%		
Team 10	69%	64%	25%						80%	48%	5%	95%	20%	23%	49%	49%	11%	20%	18%	31%	63%	26%	46%	64%	9%	
Team 11	8%						6%	3%									31%	44%	56%	61%	47%	36%	31%	67%	90%	
Team 12	53%	74%	69%	54%	72%	71%	74%	85%	63%	61%	74%	78%	70%	45%	64%	49%	88%	64%	65%	55%	78%	86%	82%	88%	85%	
Team 13	52%	90%	96%	41%	66%	38%	66%	59%	84%	63%	34%	73%	47%	48%	62%	94%	72%	30%	86%	43%	39%	32%	77%	60%	100%	
Team 14	7%		41%							16%	23%				19%	10%	49%	15%	52%	80%	53%	38%	58%	63%	56%	
Team 15	69%	38%	47%	62%	24%	89%	23%	63%	51%	46%	34%	28%	58%	4%	57%	30%	43%	50%	74%	86%	74%	59%	39%	48%	44%	
Team 16	81%	67%	75%	38%	60%	39%	61%	45%	29%	54%	47%	59%	61%	45%	65%	81%	84%	63%	65%	65%	81%	79%	59%	47%	52%	

Урок 4. Модели важны. Иди сверху вниз и учитывай все аспекты



Несколько советов

1. Начинай пробовать с самых лучших, чтобы убедиться, что гипотеза вообще потенциально реализуема.
2. Дальше начинай исследовать более маленькие и специализированные решения, чтобы найти наиболее эффективное решение. Проверь лицензионную чистоту.
3. Старайся не думать про дообучение моделей. Стоимость подготовки данных и дообучения может убить весь ожидаемый эффект. Используй другие технологии типа RAG
4. Учитывай не только качество ответа, но и скорость. Очень часто это важно.
5. Когда сравниваешь стоимость использования сервисов с отдельной инсталляцией, не забывай об обеспечении высокой доступности сервиса и Disaster Recovery. Это может сильно повлиять на расчет эффективности

Урок 5. Корпоративная разработка имеет много важных нюансов



Основные нюансы корпоративной разработки

1. Корпоративная разработка это в большинстве случаев доработка существующих систем
2. Корпоративные проекты фактически всегда в проекте имеют большое количество файлов, что приводит к невозможности полноценно использовать агентный режим.
3. Самые современные сервисы используют внешние модели, которые размещены за границами РФ, что противоречит большинству корпоративных требований безопасности
4. Фактически все сводится к использованию opensource моделей и инструментов по причине санкционных ограничений
5. 1С – пока вне AI гонки

Урок 6. Прежде чем начинать делать три раза посчитай



Подходы к оценке эффекта AI-инициатив

- 1. Целевая аудитория:** Определяем целевую аудиторию пользователей, для которой реализован функционал и использование AI функции дает максимальный прирост по производительности
- 2. Драйверы расчета:** Вычисляем драйвер для расчета эффекта на основе влияния, которое дает использование AI функционала пользователю
- 3. Потенциал:** Рассчитываем максимально возможный эффект, если бы все пользователи из целевой аудитории применяли AI функционал в своей работе и получили максимальную пользу
- 4. Эффект:** Выводим итоговый расчет эффекта и результатов, которые дает AI инициатива

Нельзя управлять тем, что невозможно измерить...

Билл Хьюлетт, HP

... но будьте готовы к вопросам по метрикам

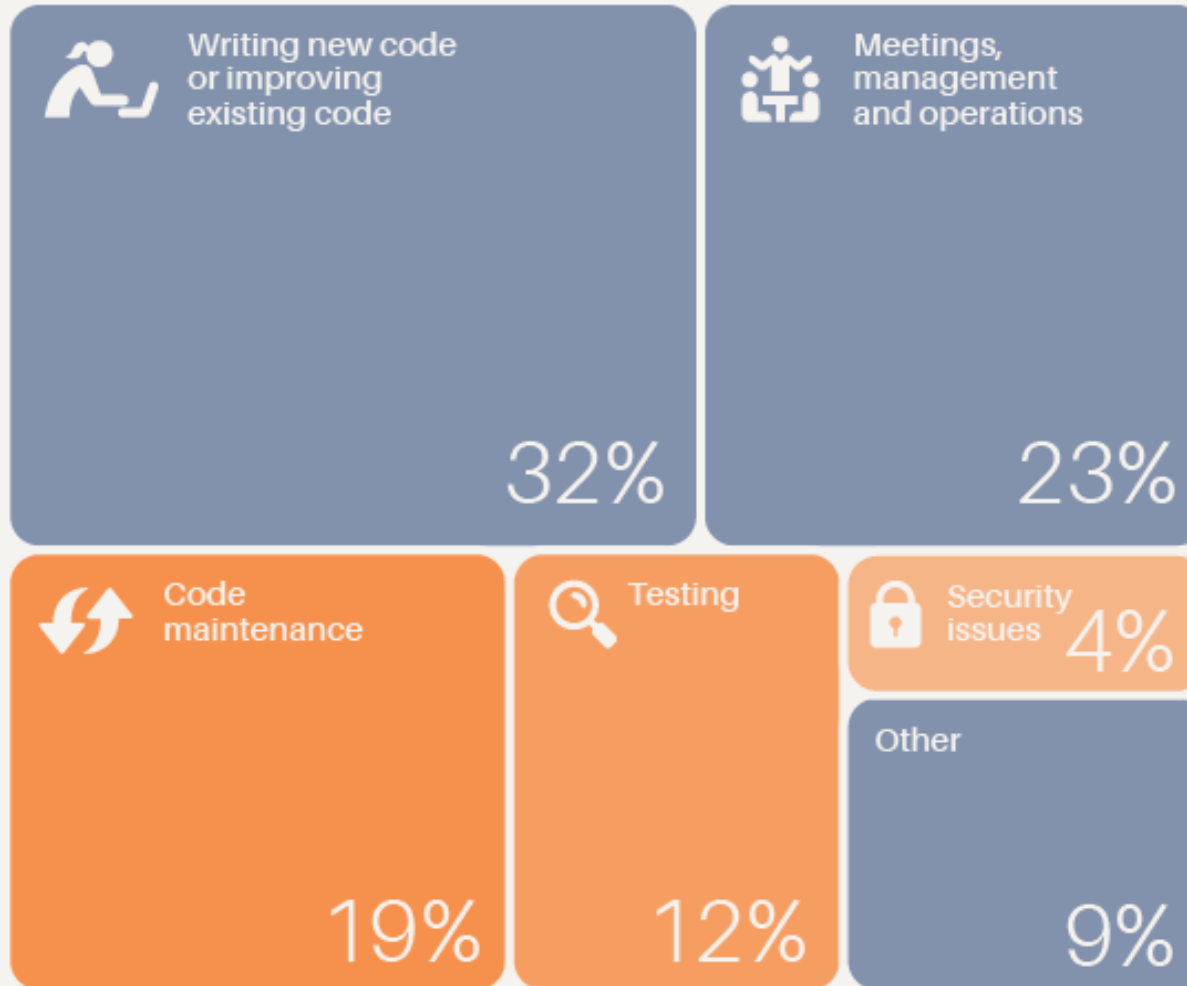
У вас уже 51% кода
генерится с помощью AI.

Почему я не вижу этого
двухкратного ускорения
разработки?



Вспоминаем куда разработчик тратит свое время в течении дня

How developers spend their time



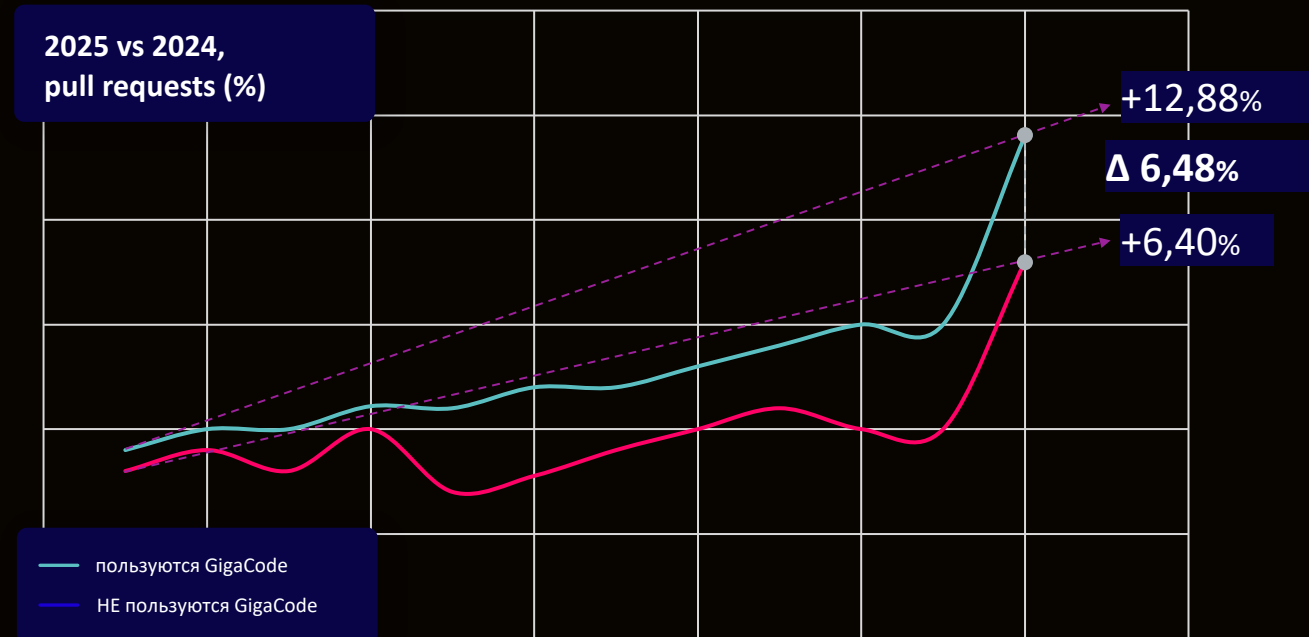
BASED ON 295 RESPONSES



Пример расчета эффекта от внедрения на более высоком уровне

Параметры исследования

- Роль – разработчик (кроме языков SWIFT и ObjectiveC)
- Количество сотрудников – 2091
- Период работы с 01.01.2024 и ранее по н.в.
- Артефакты – pull requests



Можно определить цели более высокого уровня

1. Ставим цель:
например, повышение эффективности на 10% (считаем через повышение количества закрытых задач)

2. Определяем дополнительные поддерживающие метрики

- Качество
- Lead Time

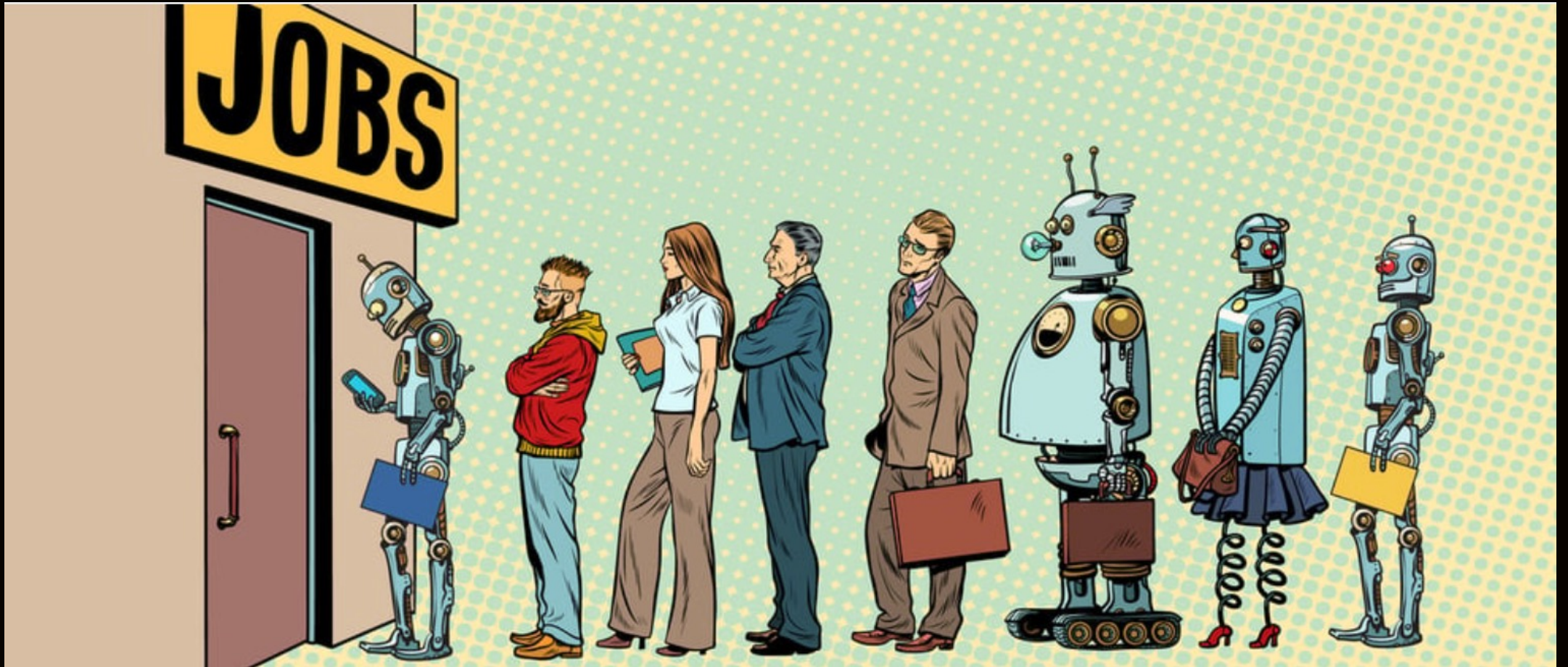
3. Считаем:

- Весь производственный процесс
- Атомарный эффект от каждого агента

ЭФФЕКТ ОТ КАЖДОГО АГЕНТА = ПРОГНОЗ X ПРОНИКНОВЕНИЕ



Урок 7. Повышение производительности не всегда позволяет оптимизироваться



Пример расчета эффекта от AI PDLC

Роль (5 ролей на которые есть влияние из 15)	Влияние AI инициатив						
	Численность*	AI агент1	AI агент2	AI агент3	AI агент4	AI агент5	Повышение производительности
Разработчик	10000	√	√	√		√	6,47%
Тестировщик	5000		√			√	0,06%
Аналитик	5000		√	√	√		0,35%
Владелец продукта	2000		√		√		1,16%
Дизайнер	1000		√		√		1,20%

Вывод:

Текущий набор AI-инструментов не покрывает полный спектр набор производственных ролей. Основной эффект видим на разработку.

Пример расчета возможности оптимизации численности

Возможность получения эффекта от AI Инструментов при централизации на разных уровнях

	Эффект повышения производительности для разработчика % (ПШЕ)	Минимальный квант	Эффект оптимизации при реструктуризации
AS IS			0
Кластер			7
Трайб	6,47% (167)	15	49
Юнит			70
Направление			167

Вывод:

Мы получаем эффект в повышении производительности по отдельным ролям, но в текущей структуре не можем оптимизировать численность

И еще одна вещь...

Начались проблемы с подбором
и своевременной оценкой
знаний сотрудника на
испытательном сроке

Спасибо

- Не знаешь с чего начать?
- Есть задачи, которые не получается решить сейчас обычным способом?
- Нужна помощь в подготовке системного внедрении AI?

Пиши и мы обязательно поможем

vladimirdol@gmail.com

