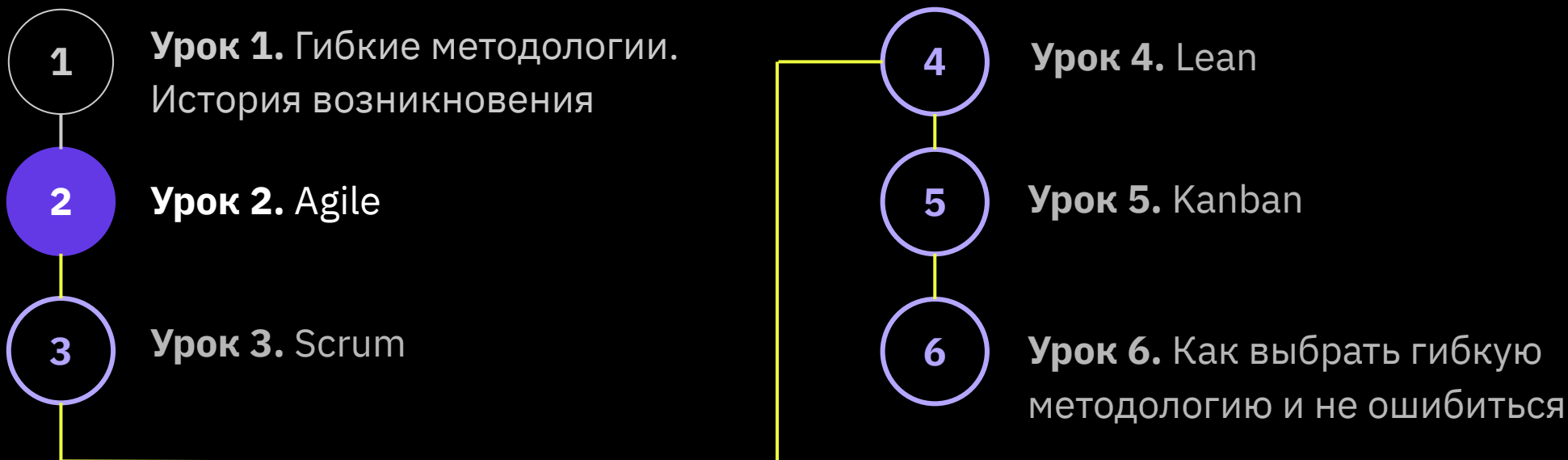


# Гибкие методологии

## Урок 2. Agile

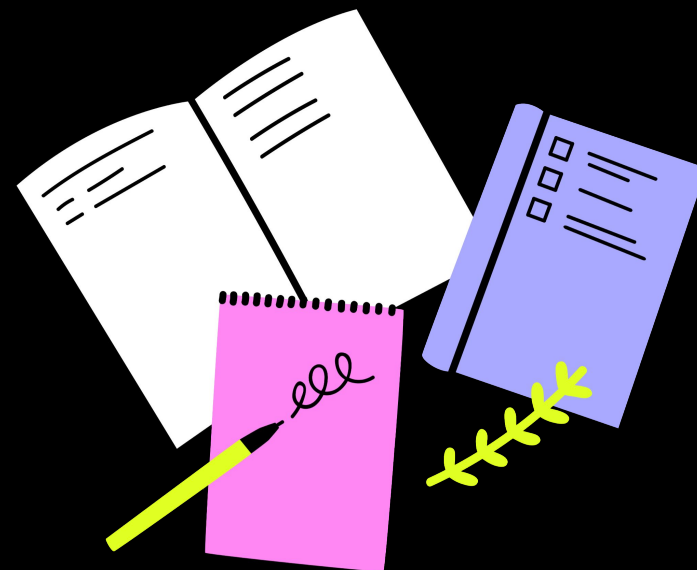


# План курса:



## План лекции:

- Что такое Agile
- История Agile
- Появление и предназначение Agile
- Ценности Agile
- Принципы Agile
- Практики
- Agile методологии
- Преимущества Agile
- Недостатки Agile
- Границы применимости Agile

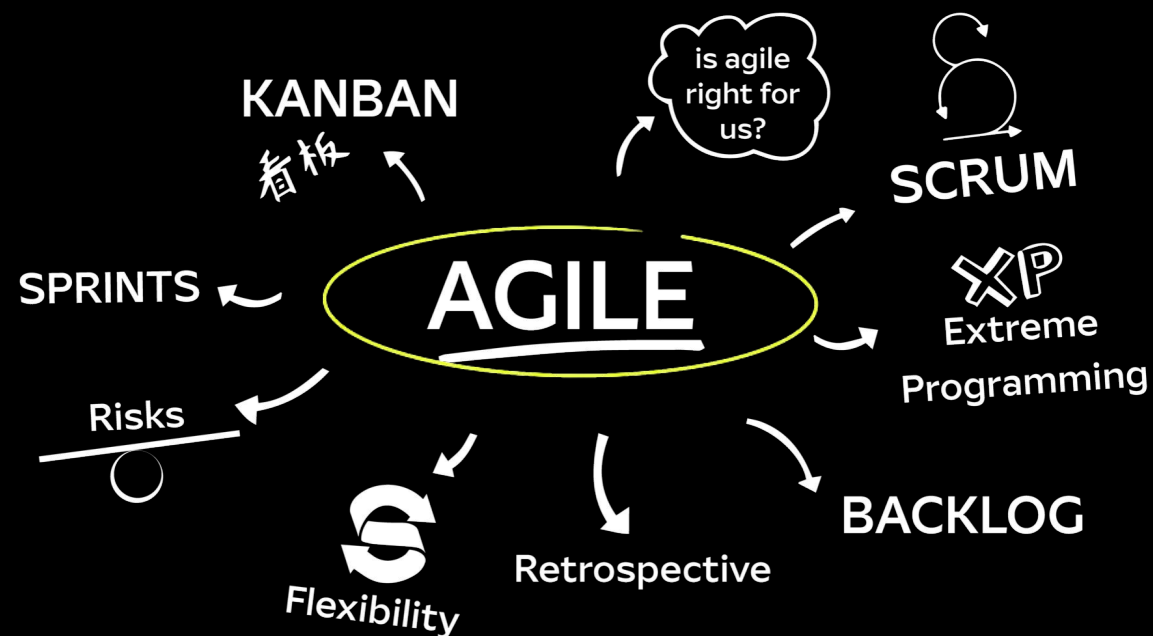


# Что такое Agile



# Agile

- С английского переводится как «подвижный, быстрый, проворный»
- «Гибкие методологии»
- Agile-методология динамично организует разработку IT-продукта, постоянно адаптируя проект к новым обстоятельствам и требованиям





# Вспомним Waterfall



# Как устроена разработка по Waterfall



# Как устроена разработка по Waterfall

1. Выдвигаются требования к ПО, разрабатывается техническое задание (ТЗ).
2. Поставленные задачи воплощаются в коде.
3. Выполняется тестирование ПО.
4. Готовое ПО внедряется в работу.





## Различия

### Waterfall

- Конечный продукт должен идеально соответствовать требованиям, целям и задачам, которые были сформулированы до разработки

### Agile

- В приоритете не исходные установки, а актуальные потребности пользователя
- Нет предварительного генерального плана
- ИТ-продукт создается экспромтом



# Разработка по Agile

---

---

---



# Как устроена разработка по Agile



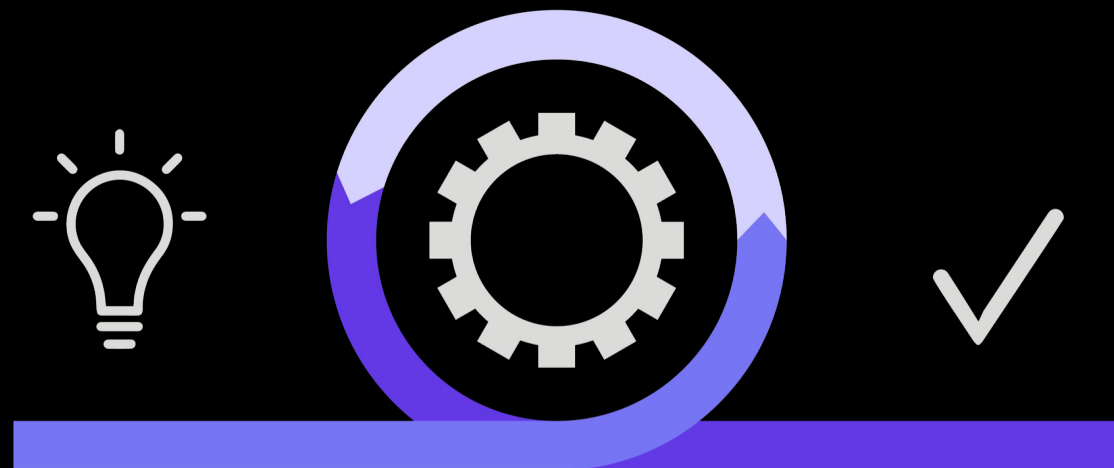
# Как устроена разработка по Agile

1. Проходит через ряд циклов — итераций



# Как устроена разработка по Agile

1. Проходит через ряд циклов — итераций
2. Каждая итерация — это фактически отдельный проект



# Как устроена разработка по Agile

1. Проходит через ряд циклов — итераций
2. Каждая итерация — это фактически отдельный проект
3. В рамках итерации разрабатывают фрагмент программы, улучшают функциональность, добавляют новые возможности





# ПРИМЕР по Agile



## Пример по Agile

1. Коллектив разработчиков создаёт аудиоплеер.
2. Уже написан интерфейс и базовый функционал, воспроизводятся файлы формата MP3, WAV и OGG.
3. Пользователи предлагают добавить проигрывание CD-дисков и подключить горячие клавиши, чтобы быстро управлять плеером.

**Начинается новая итерация разработки.**





## Пример по Agile

1. Коллектив разработчиков создаёт аудиоплеер.
2. Уже написан интерфейс и базовый функционал, воспроизводятся файлы формата MP3, WAV и OGG.
3. Пользователи предлагают добавить проигрывание CD-дисков и подключить горячие клавиши, чтобы быстро управлять плеером.

### Начинается новая итерация разработки.

4. Программисты собираются на короткое совещание для реализации нового функционала.  
Один из разработчиков предлагает добавить воспроизведение онлайн-радио.
5. Разработка. Создается программный код, интегрируется в продукт, выполняется тестирование.
6. Новый функционал готов – выпускается новая версия продукта.

**Итерация завершается и начинается новый виток разработки.**



# Как это выглядит по Waterfall



## Как это выглядит по Waterfall

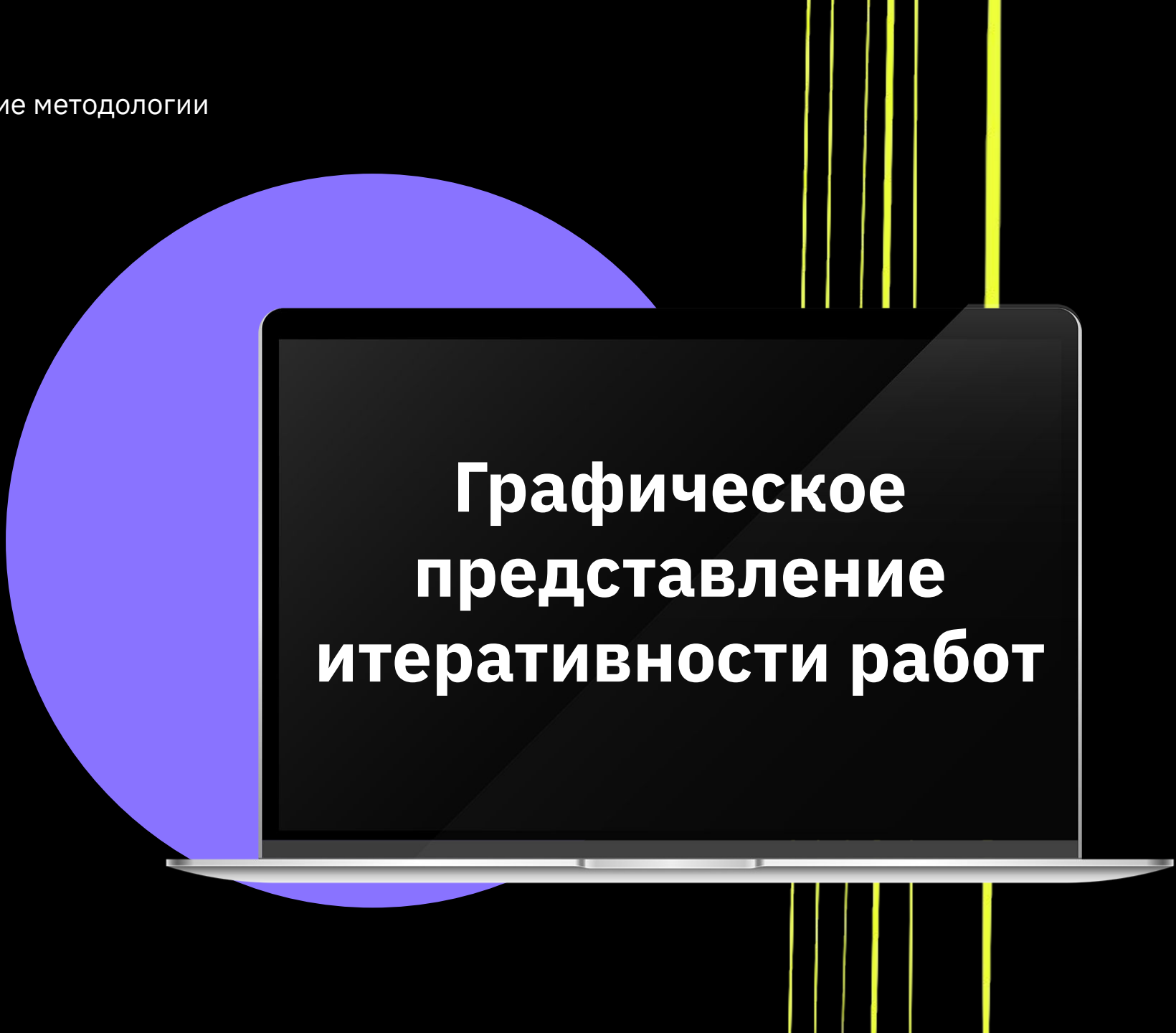
1. Коллектив разработчиков создаёт аудиоплеер.
2. В ТЗ прописано детально, какие форматы должен проигрывать плеер, как должен выглядеть дизайн и с какими ПО плеер может взаимодействовать.
3. Мы не можем выпустить MVP с минимальным набором функций = не узнаем, что пользователи хотят видеть в плеере горячие клавиши и проигрывание CD-дисков.



# Как это выглядит по Waterfall

1. Коллектив разработчиков создаёт аудиоплеер.
2. В ТЗ прописано детально, какие форматы должен проигрывать плеер, как должен выглядеть дизайн и с какими ПО плеер может взаимодействовать.
3. Мы не можем выпустить MVP с минимальным набором функций = не узнаем, что пользователи хотят видеть в плеере горячие клавиши и проигрывание CD-дисков.
4. По мере разработки плеера у наших программистов появляются идеи, как улучшить продукт – добавить воспроизведение онлайн-радио.
5. Это интересная фича, но необходимо будет посчитать бюджет, составить детальный план, просчитать риски, согласовать с заказчиком, внести изменение в проект и двигаться дальше.
6. Мы снова не можем протестировать эту фичу, пока не будет готов весь продукт.





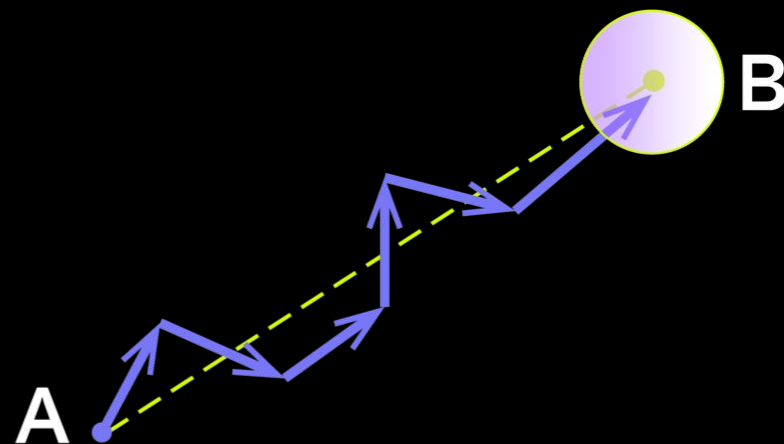
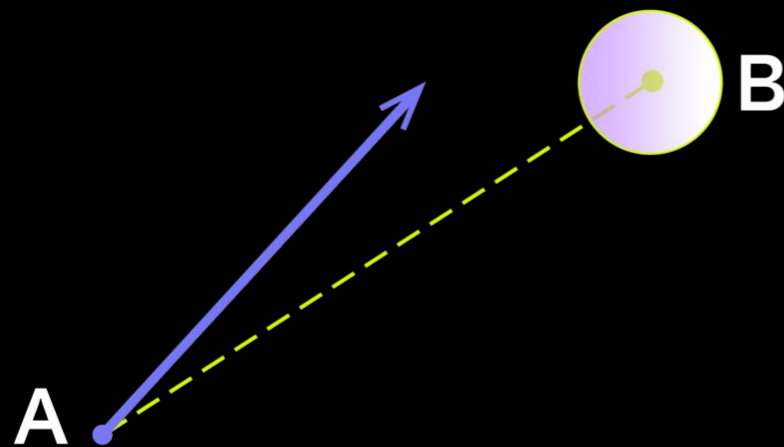
# Графическое представление итеративности работ



# Waterfall

VS

# Agile



# Почему это важно?



## Почему это важно?

При коммерческой разработке у нас постоянно меняются условия:

- ✓ выходят новые законы
- ✓ изменяются бизнес-требования
- ✓ конкуренты вдруг выпускают что-то похожее на наш продукт или даже круче

**Мы должны соответствовать этим условиям,  
делать продукт лучше, чем конкуренты!**





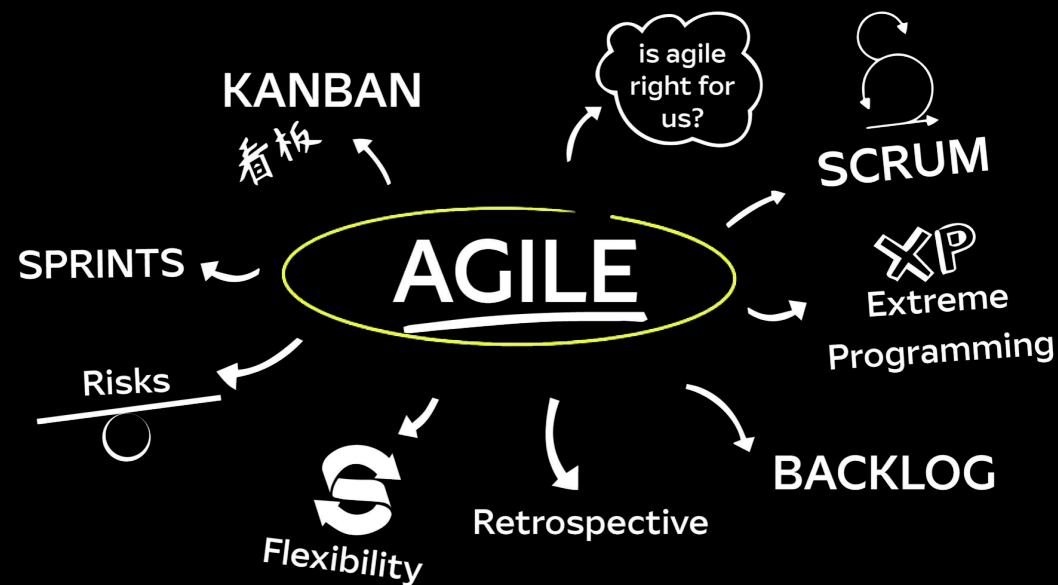
## Почему это важно?

Мы должны соответствовать этим условиям,  
делая продукт лучше, чем конкуренты!

---

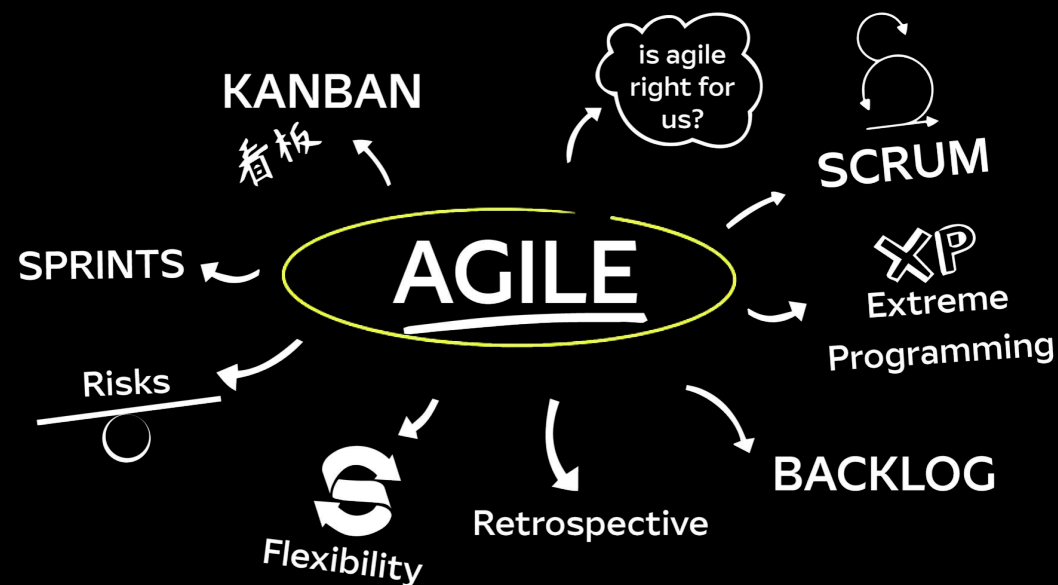


# Agile методология подразумевает



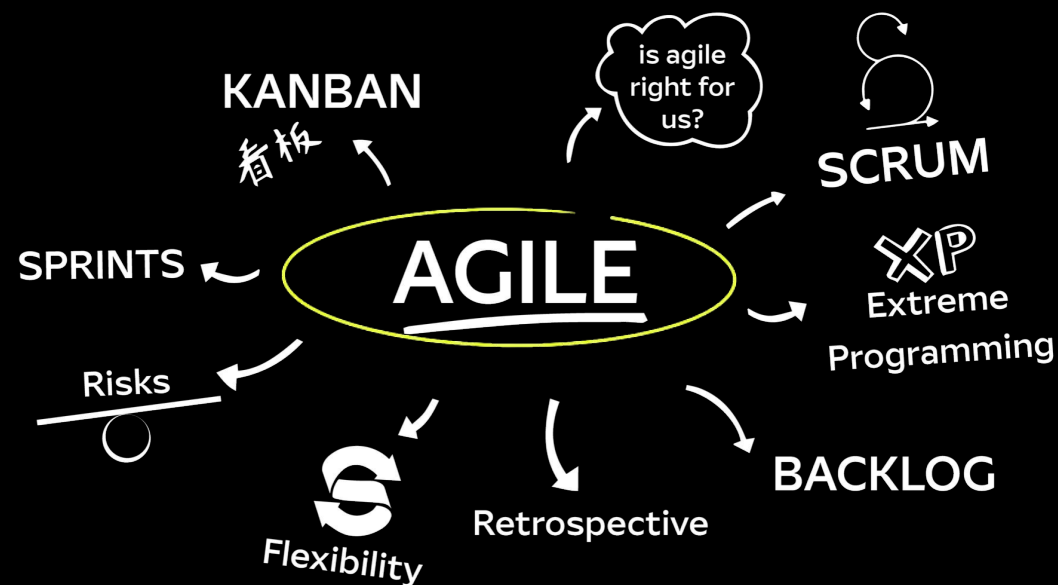
# Agile методология подразумевает

- ✓ После выпуска первого релиза вы снимаете метрики, что-то переделываете, делаете следующий релиз, и т.д.



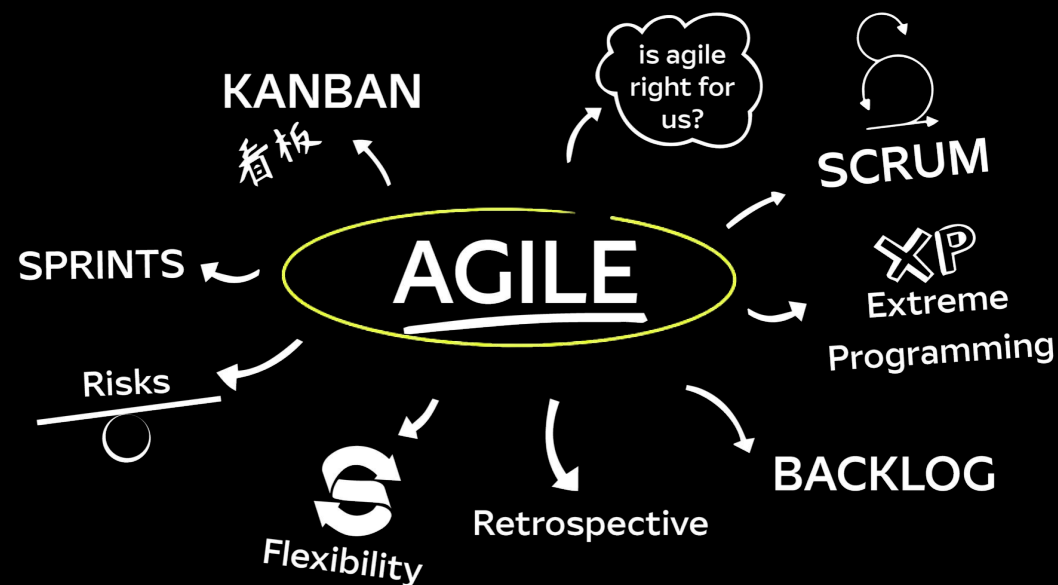
# Agile методология подразумевает

- ✓ После выпуска первого релиза вы снимаете метрики, что-то переделываете, делаете следующий релиз, и т.д.
- ✓ На каком-то этапе понимаете, что конкурент выпустил какую-то фичу, и вам нужно идти не туда



# Agile методология подразумевает

- ✓ После выпуска первого релиза вы снимаете метрики, что-то переделываете, делаете следующий релиз, и т.д.
- ✓ На каком-то этапе понимаете, что конкурент выпустил какую-то фичу, и вам нужно идти не туда
- ✓ В этот момент вы можете остановиться, принять другие требования и начать двигаться в новом направлении



# Подход к успешности проекта

Agile

Waterfall



# Подход к успешности проекта

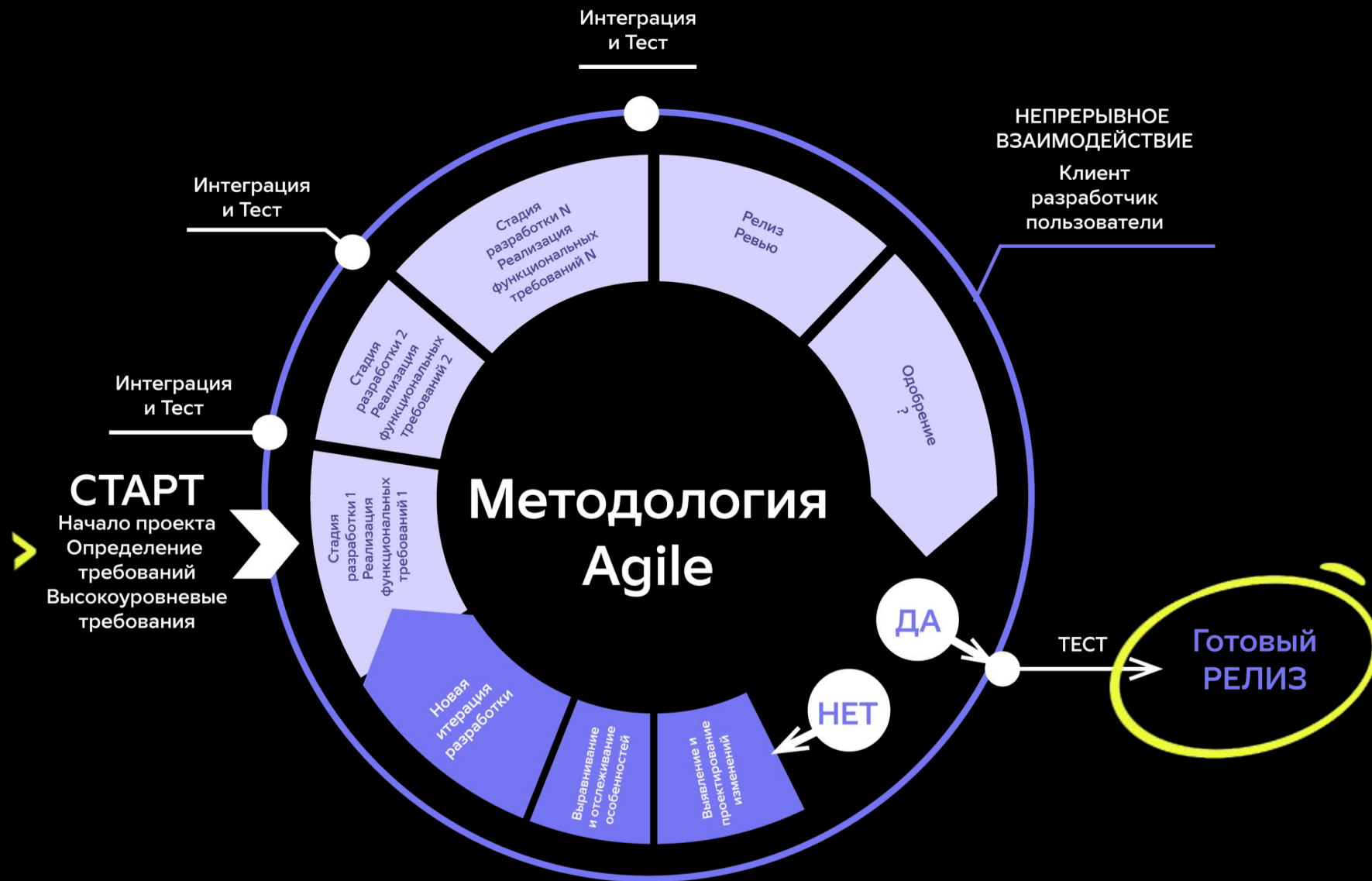
## Agile

- Чем больше ценностей мы поставили Заказчику в рамках разработки, тем успешнее наш проект

## Waterfall

- Когда он уложился в рамки железного треугольника проекта: сроки, бюджет, содержание и качество









# История возникновения



# Как появился Agile

- В феврале 2001 года 17 независимых практиков нескольких методик программирования, именующих себя «Agile Alliance», собрались на горном курорте в штате Юта (США), отдохнули, пообщались и составили небольшой документ – **Agile-манифест**.
- <http://agilemanifesto.org/iso/ru/manifesto.html>



# Создатели Agile



# Agile-манифест

- основной документ
- содержит описание ценностей и принципов гибкой разработки ПО
- текст манифеста доступен на более чем 50 языках
- включает в себя 4 ценности и 12 принципов



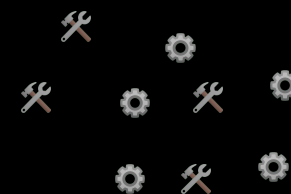
1 mindset



4 values



12 principles



Many  
processes & tool



# Появление и предназначение Agile

---



# Задачи перед ИТ-сферой – способствовали появлению Agile



# Задачи перед ИТ-сферой – способствовали появлению Agile

- Ускорение вывода продукта на рынок
- Управление изменениями в приоритетах
- Улучшение взаимодействия между ИТ и бизнесом

Ответ на все эти вызовы – Agile-манифест



# Ускорение вывода продукта на рынок

 **1-ая компания  
по разработке ПО**

 **2-ая компания  
по разработке ПО**



# Ускорение вывода продукта на рынок

## 1-ая компания по разработке ПО

- ✓ Использует Waterfall
- ✓ Пишет ТЗ
- ✓ Проектирует систему
- ✓ Рисует дизайн
- ✓ Уходит несколько месяцев

## 2-ая компания по разработке ПО



# Ускорение вывода продукта на рынок

## 1-ая компания по разработке ПО

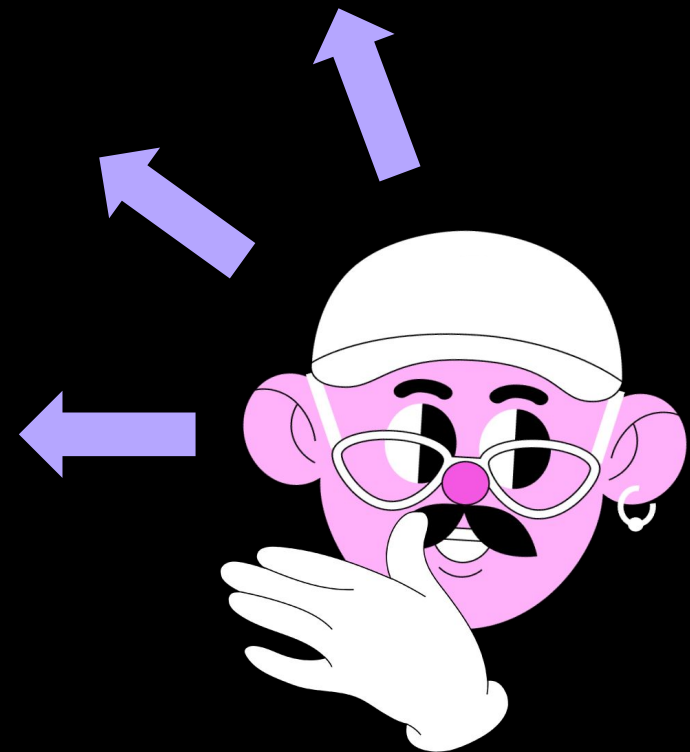
- ✓ **Использует Waterfall**
- ✓ Пишет ТЗ
- ✓ Проектирует систему
- ✓ Рисует дизайн
- ✓ Уходит несколько месяцев

## 2-ая компания по разработке ПО

- ✓ **Использует Agile**
- ✓ К этому времени уже запускает сайт, выпускает ПО
- ✓ Начинает зарабатывать деньги
- ✓ Захватывает рынок
- ✓ Привлекает новых пользователей

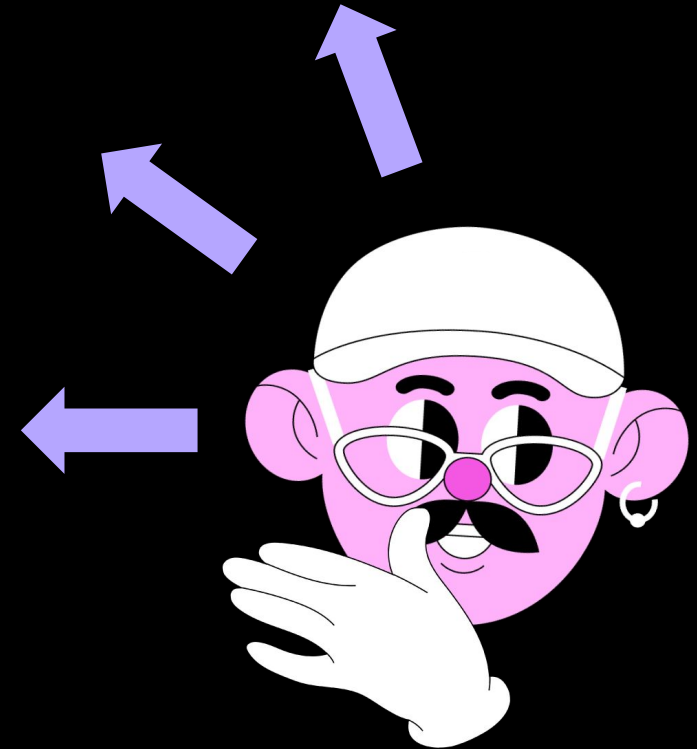


# Управление изменениями в приоритетах



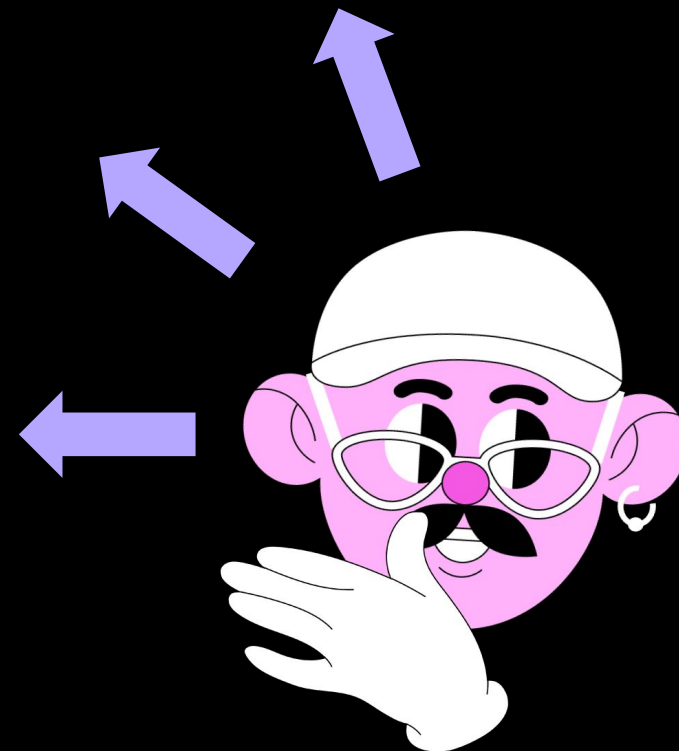
# Управление изменениями в приоритетах

- Одна из самых болезненных проблем
- Если проект длится хотя бы несколько месяцев, то обязательно поменяются требования



# Управление изменениями в приоритетах

- Одна из самых болезненных проблем
- Если проект длится хотя бы несколько месяцев, то обязательно поменяются требования
- Никто изначально не знает в деталях, что нужно заказчику
- Пока пользователь не попробует функционал сайта или ПО, мы не узнаем, нужен он или нет



# Улучшение взаимодействия между IT и бизнесом

1. У бизнеса меняются требования
2. Каждый говорит на своём языке
3. IT не понимает бизнес
4. Бизнес не понимает IT





# Ценности Agile



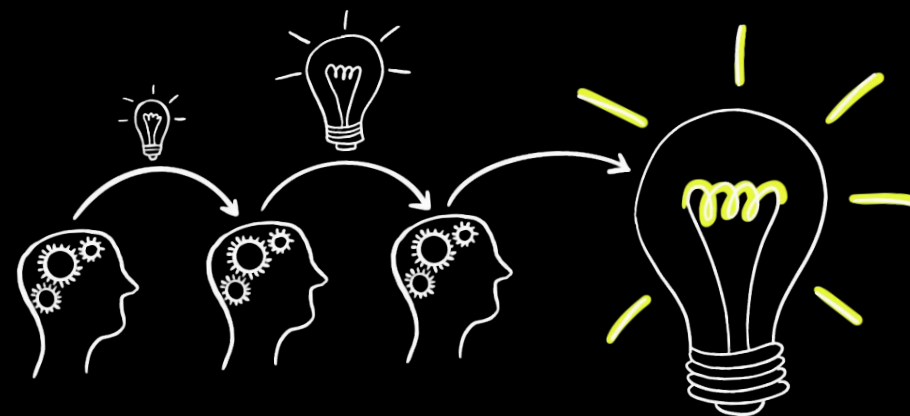
# 4 центральных идеи Agile Manifesto





## 4 центральных идеи Agile Manifesto

1. Люди и взаимодействие **важнее**, чем процессы и инструменты.
2. Работающее ПО **важнее**, чем исчерпывающая документация.
3. Сотрудничество с заказчиком **важнее**, чем согласование условий контракта.
4. Готовность к изменениям **важнее**, чем следование первоначальному плану.



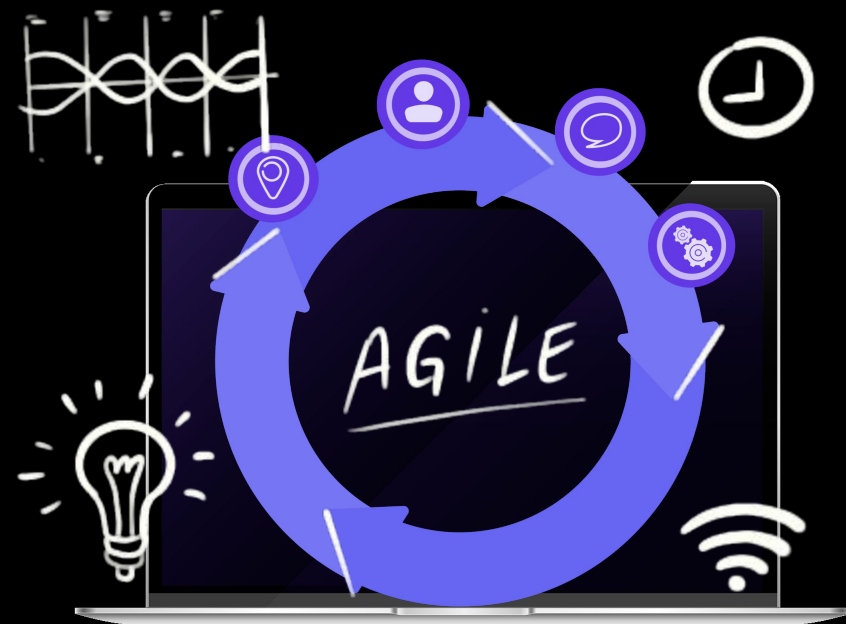


# **12 принципов Agile**



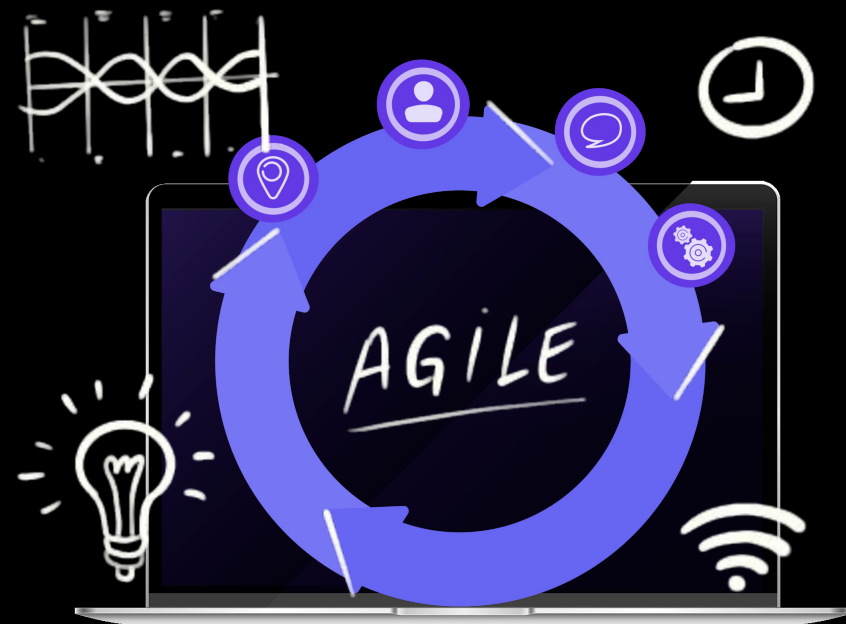
# 12 принципов Agile

1. **Наивысшая ценность** – это удовлетворение потребностей заказчика, благодаря регулярной и максимально ранней поставке ценного для него ПО.



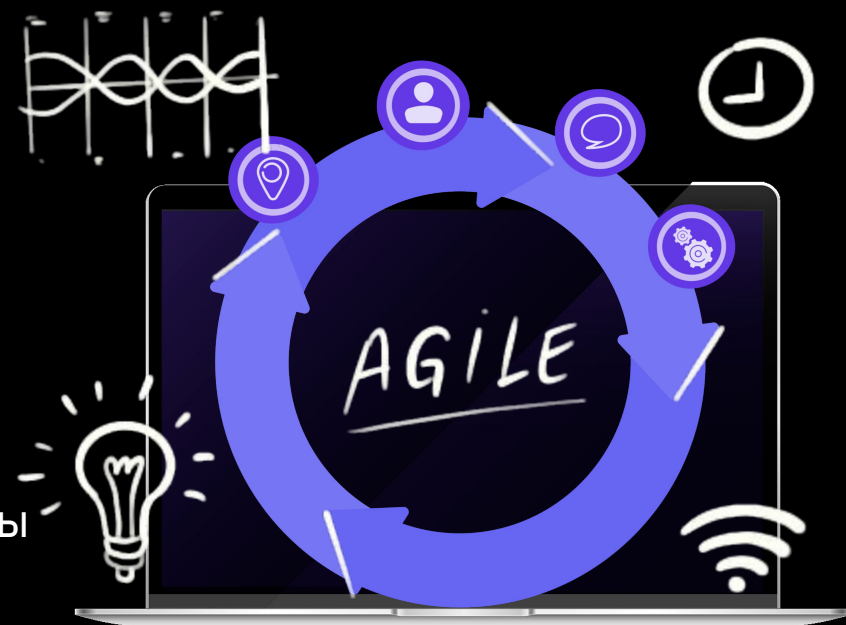
## 12 принципов Agile

1. **Наивысшая ценность** – это удовлетворение потребностей заказчика, благодаря регулярной и максимально ранней поставке ценного для него ПО.
2. Мы всегда готовы изменять требования, даже на поздних стадиях проекта, если узнаем что-то новое. Таким образом, мы создаем бизнесу или внешнему заказчику конкурентное преимущество.



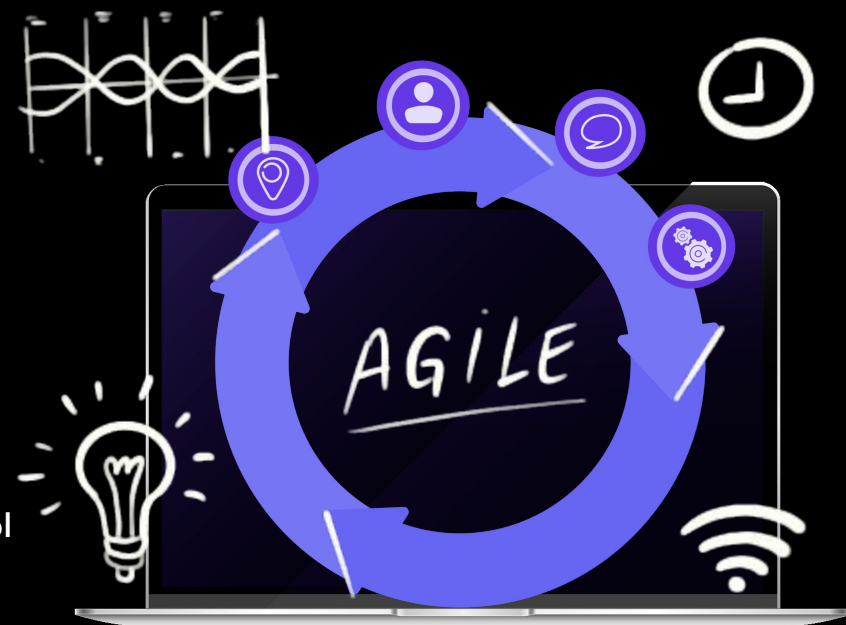
## 12 принципов Agile

1. **Наивысшая ценность** – это удовлетворение потребностей заказчика, благодаря регулярной и максимально ранней поставке ценного для него ПО.
2. Мы всегда готовы изменять требования, даже на поздних стадиях проекта, если узнаем что-то новое. Таким образом, мы создаем бизнесу или внешнему заказчику конкурентное преимущество.
3. При использовании Agile работающий продукт выпускают максимально часто. В манифесте прописаны сроки — от пары недель до пары месяцев.



## 12 принципов Agile

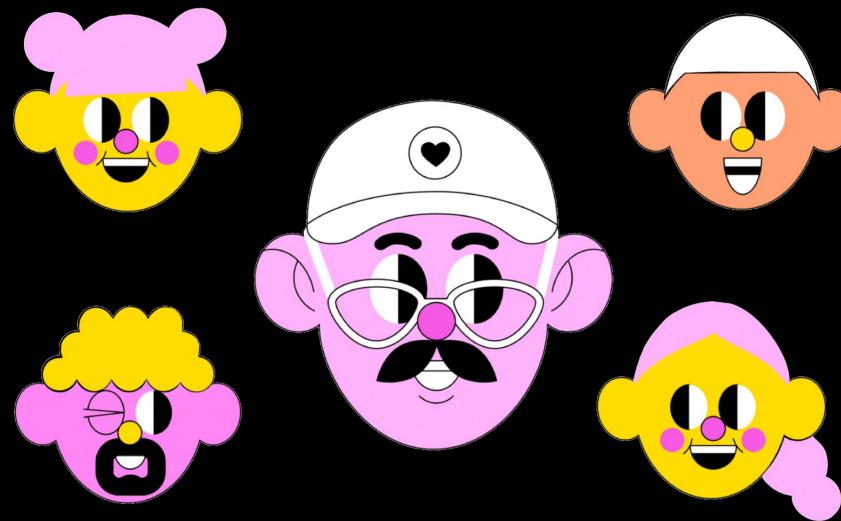
1. **Наивысшая ценность** – это удовлетворение потребностей заказчика, благодаря регулярной и максимально ранней поставке ценного для него ПО.
2. Мы всегда готовы изменять требования, даже на поздних стадиях проекта, если узнаем что-то новое. Таким образом, мы создаем бизнесу или внешнему заказчику конкурентное преимущество.
3. При использовании Agile работающий продукт выпускают максимально часто. В манифесте прописаны сроки — от пары недель до пары месяцев.
4. Бизнес обязательно должен работать вместе с программистами, помогать им понять специфику данного рынка



## 12 принципов Agile

5. Команда – один из краеугольных камней Agile. Наилучших результатов достигает команда мотивированных профессионалов.

Руководитель прежде всего должен создавать условия для команды и обеспечивать всестороннюю поддержку, проводить коучинг, следить за атмосферой в коллективе





# 12 принципов Agile

6. Самый простой и эффективный способ узнать требования клиента, заказчика или пользователя – поговорить с ними.





# 12 принципов Agile

- 6. Самый простой и эффективный способ узнать требования клиента, заказчика или пользователя – поговорить с ними.
- 7. Работающий продукт.

Степень готовности проекта должна измеряться не словами о том, что ТЗ уже написано и 50% макетов нарисовано, а количеством функционала, выпущенного в production.



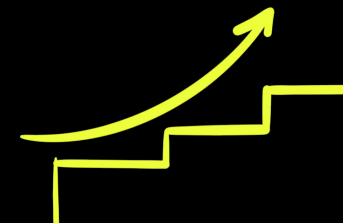
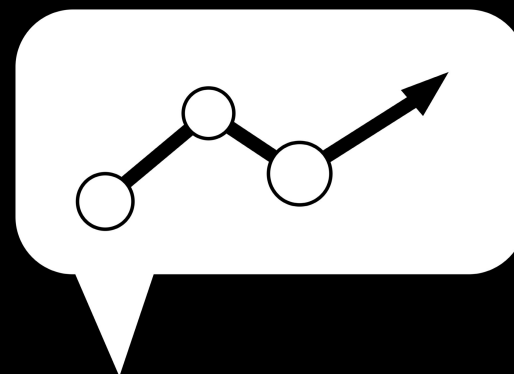
# 12 принципов Agile

8. В Agile важен ритм, постоянные улучшения.

Бизнес и программисты всегда должны располагать возможностью делать процесс устойчивым, постоянно его улучшать.

9. Agile вообще не будет работать, если вы написали кривой код.

У вас должна быть хорошая гибкая архитектура, в которую можно добавлять разные элементы и, при необходимости, легко их изменять.



## 12 принципов Agile

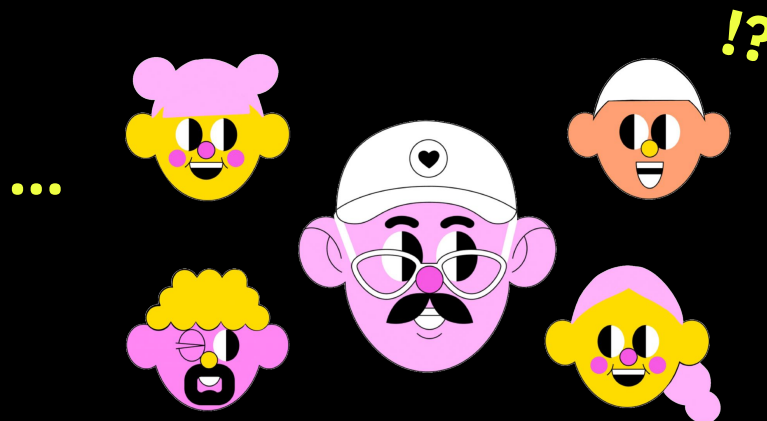
- 10.** Простота. Она проявляется в технической составляющей, в дизайне.

Простота очень важна также с точки зрения выпуска продукта: когда вы хотите «нарезать» того «слона», лучше начать с простой части.



## 12 принципов Agile

- 11.** Менеджер в команде меняет свою роль: он не столько занимается организацией процесса, сколько учит команду, поэтому команда должна быть самоорганизованной.
- 12.** Команда должна регулярно анализировать реализуемые процессы: что получилось, как они этого добились, и постоянно улучшать организацию работы.



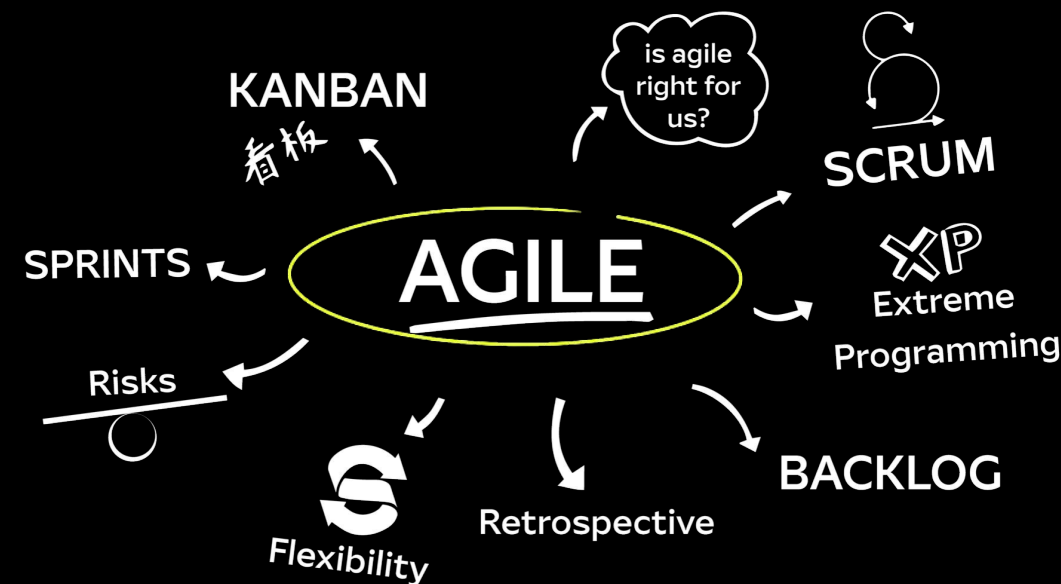


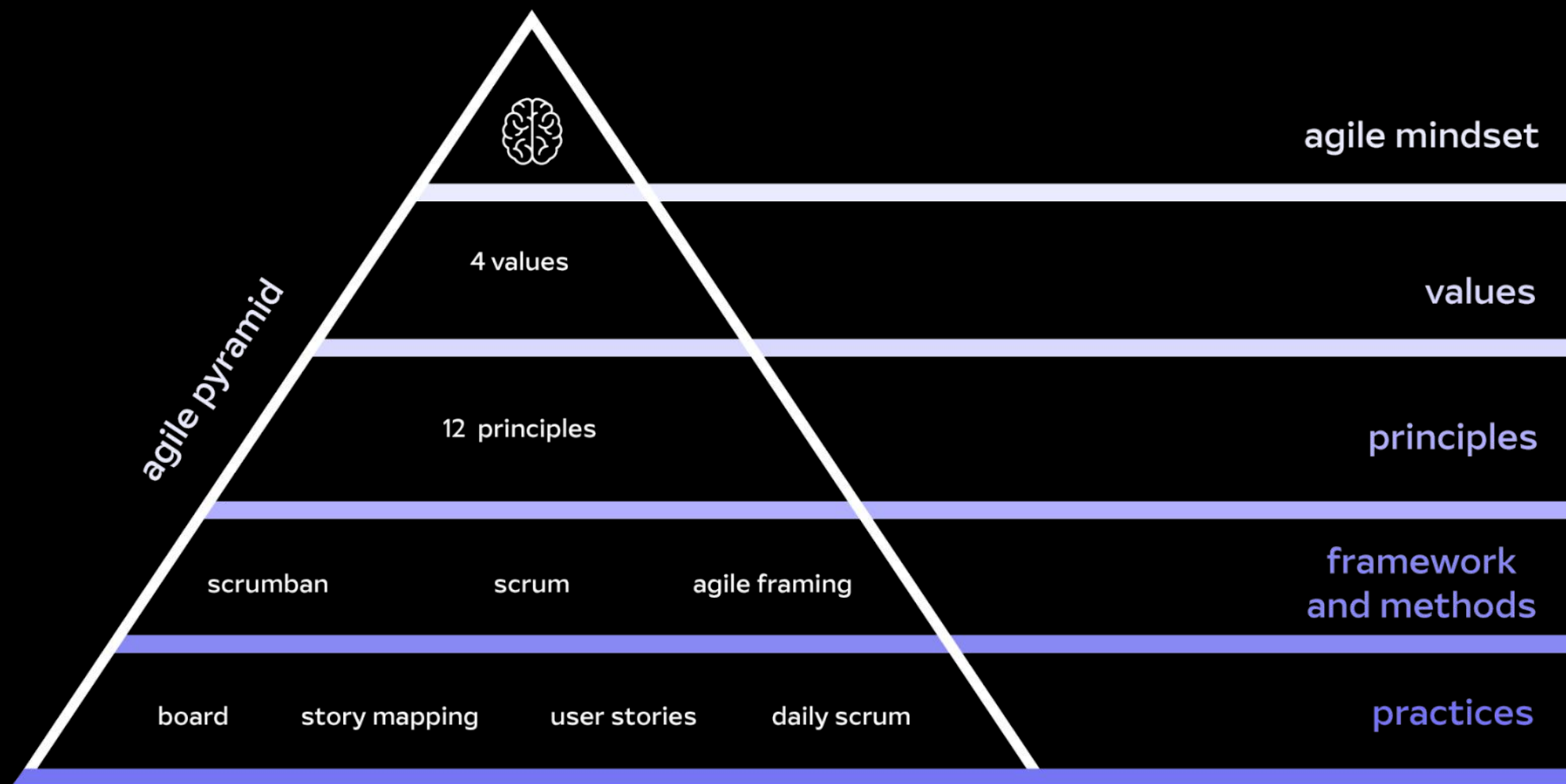
# ВЫВОДЫ



## Выводы

- ✓ Руководствуясь только этими идеями и принципами, выстроить рабочие процессы нельзя.
- ✓ **Agile** – это целая серия подходов к разработке ИТ-продуктов путем непрерывной и быстрой поставки ценного рабочего функционала самоорганизованной командой профессионалов в сотрудничестве с заказчиком.





# Практики Agile








# ВЫВОДЫ

# Выводы

Одна из ценностей Agile гласит:



**Мы должны выстраивать рабочий процесс, налаживая взаимодействие и коммуникацию между людьми.**



# Пример



# Пример

- ✓ Утром команда разработчиков проводит совещание и синхронизирует свою деятельность на предстоящий день.
- ✓ Совещание = стендап



## Пример

- ✓ Утром команда разработчиков проводит совещание и синхронизирует свою деятельность на предстоящий день.
- ✓ Совещание = стендап

**Можно вместо стендапов каждому писать отчет о проделанной вчера работе, но это не Agile.**

Почему?

Потому что возникает поток малозначимой документации.



# Часто используемые практики Agile



# Часто используемые практики Agile

## 1. Стендапы

Каждый день в определенное время команда собирается и синхронизирует свою деятельность



# Часто используемые практики Agile

## 1. Стендапы

Каждый день в определенное время команда собирается и синхронизирует свою деятельность

## 2. Планирование итераций

Команда планирует объем работ на ближайшую итерацию







# Часто используемые практики Agile

## 1. Стендапы

Каждый день в определенное время команда собирается и синхронизирует свою деятельность

## 2. Планирование итераций

Команда планирует объем работ на ближайшую итерацию

## 3. Unit-тестирование

Проверка на корректность отдельных единиц работы исходного кода программы. Минимальными единицами работы для нас являются процедуры и функции.





# Часто используемые практики Agile

## 1. Стендапы

Каждый день в определенное время команда собирается и синхронизирует свою деятельность

## 2. Планирование итераций

Команда планирует объем работ на ближайшую итерацию

## 3. Unit-тестирование

Проверка на корректность отдельных единиц работы исходного кода программы. Минимальными единицами работы для нас являются процедуры и функции.

## 4. Планирование релизов

Планирование большими кусками – что и когда будем выпускать в свет

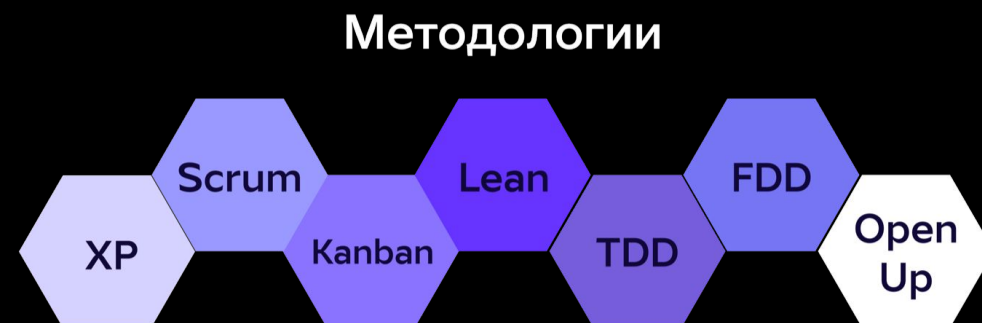


# Agile методологии



# Agile методология

- ✓ Scrum (самая большая и часто применяемая методология)
- ✓ Экстремальное программирование (XP)
- ✓ DSDM (Dynamic Systems Development Method)
- ✓ Crystal
- ✓ FDD (Feature Driven Development)
- ✓ Kanban



# Преимущества и недостатки Agile



Преимущества



Недостатки

# Преимущества и недостатки Agile



## Преимущества

- ПО готово к использованию на самых ранних этапах его разработки, пусть и не с полной функциональностью
- Разработчики постоянно в контакте с заказчиком
- Нет жестких рамок = ПО постоянно изменяется и улучшается
- Заказчик и пользователь партнёры и идейные вдохновители
- Нет заранее и подробно сформулированного ТЗ = разработчик может решить задачу творчески



## Недостатки



# Преимущества и недостатки Agile



## Преимущества

- ПО готово к использованию на самых ранних этапах его разработки, пусть и не с полной функциональностью
- Разработчики постоянно в контакте с заказчиком
- Нет жестких рамок = ПО постоянно изменяется и улучшается
- Заказчик и пользователь партнёры и идейные вдохновители
- Нет заранее и подробно сформулированного ТЗ = разработчик может решить задачу творчески



## Недостатки

- Не факт, что программа когда-нибудь будет завершена
- Пользователь требует все и сразу
- Работа над проектом требует не только профессионализма разработчика, но и сознательности пользователя
- «Золотые пользователи»
- Строительство без чертежей
- Постоянная спешка



# Границы применимости Agile

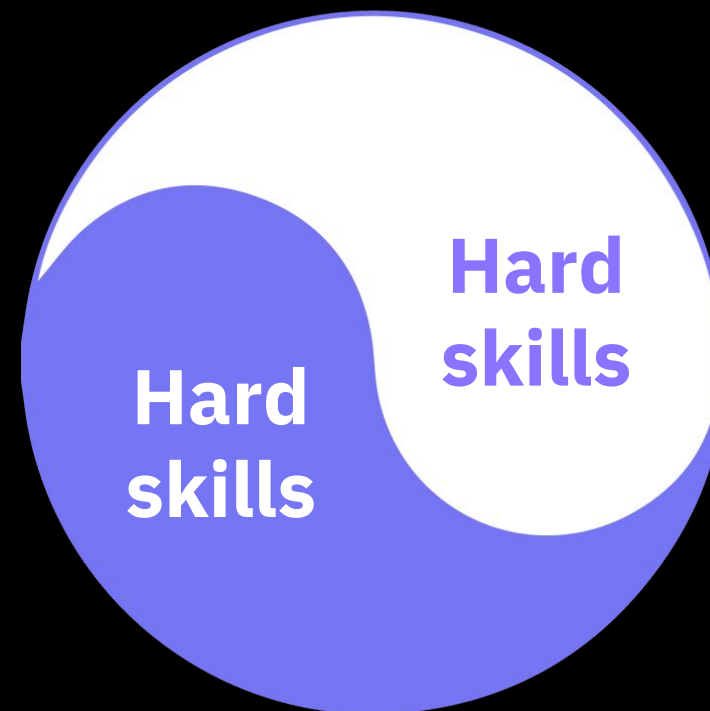
---





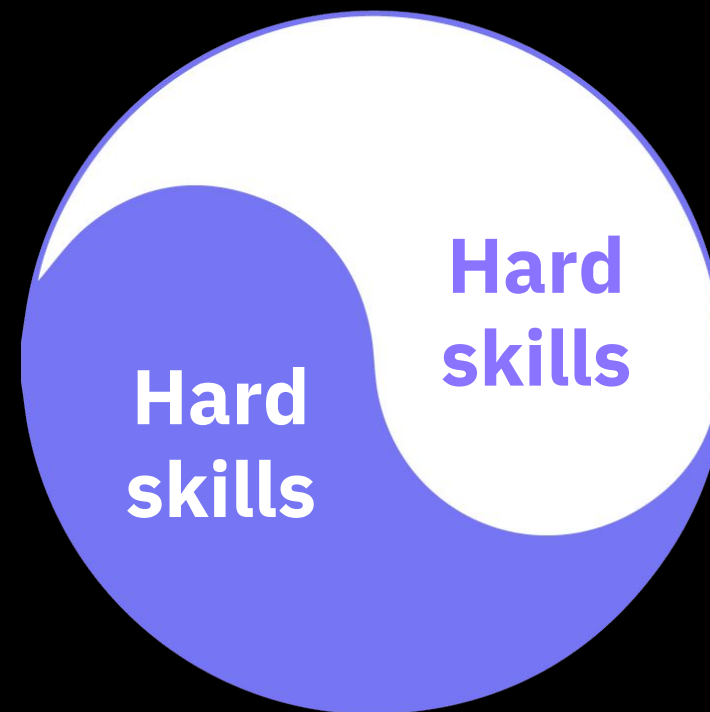
## Границы применимости Agile

- ✓ Agile предъявляют высокие требования к профессионализму, квалификации и настрою специалистов



## Границы применимости Agile

- ✓ Agile предъявляют высокие требования к профессионализму, квалификации и настрою специалистов
- ✓ Могут сделать рабочие процессы быстрыми и эффективными



## Границы применимости Agile

- ✓ Agile предъявляют высокие требования к профессионализму, квалификации и настрою специалистов
- ✓ Могут сделать рабочие процессы быстрыми и эффективными
- ✓ Могут мотивировать членов команды трудиться творчески и развиваться



## Границы применимости Agile

- ✓ Agile предъявляют высокие требования к профессионализму, квалификации и настрою специалистов
- ✓ Могут сделать рабочие процессы быстрыми и эффективными
- ✓ Могут мотивировать членов команды трудиться творчески и развиваться
- ✓ Не являются универсальным средством для всех проектов



# Agile хорошо применим





## Agile хорошо применим

- ✓ В вашей команде работают опытные и квалифицированные специалисты.
- ✓ У заказчика нет ясного представления, как должен выглядеть готовый итоговый продукт, но он готов участвовать в совместной работе, чтобы развивать и улучшать продукт.



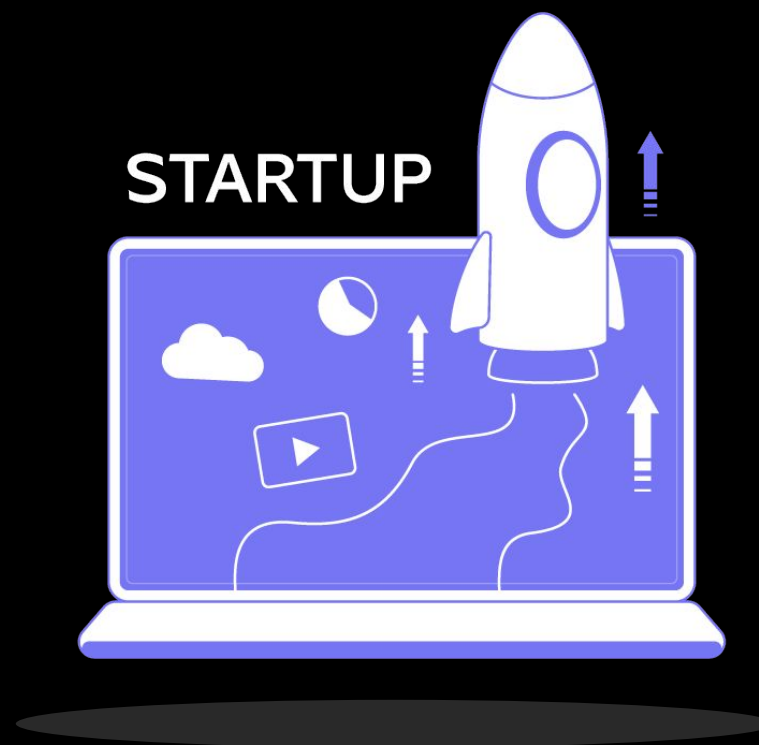
## Agile хорошо применим

- ✓ В вашей команде работают опытные и квалифицированные специалисты.
- ✓ У заказчика нет ясного представления, как должен выглядеть готовый итоговый продукт, но он готов участвовать в совместной работе, чтобы развивать и улучшать продукт.
- ✓ Сфера применения продукта постоянно меняется, часто возникают новые потребности и задачи.
- ✓ Нет конкретных сроков, к которым продукт должен быть завершен, и жестких ограничений бюджета.
- ✓ Работоспособную программу необходимо получить как можно скорее.



# Яркий пример удачного применения Agile

- Инновации в конкретной области
- Важно успеть занять нишу
- Быстрее выдать работающий продукт
- Нет долгосрочных прогнозов
- Не известно будущее продукта





Спасибо  
за внимание



@igor\_zuriev



i.zuriev@gmail.com

