

Управление производственным процессом разработки программного обеспечения

Введение

Кто читает этот курс

Лектор

Анойкин Денис Александрович

d.anoykin@g.nsu.ru

aden@ngs.ru

Семинарист групп ФИТ

Исмагилов Тимур Зинферович

Семинарист групп ВКИ

Кудинов Виталий Юрьевич

О чем этот курс

В этом курсе будет рассмотрены:

- Современные методологии разработки ПО
- Инструментарий, используемый при разработке ПО
- Распространенные инструменты
- Конфигурационное управление
- Управление большими данными

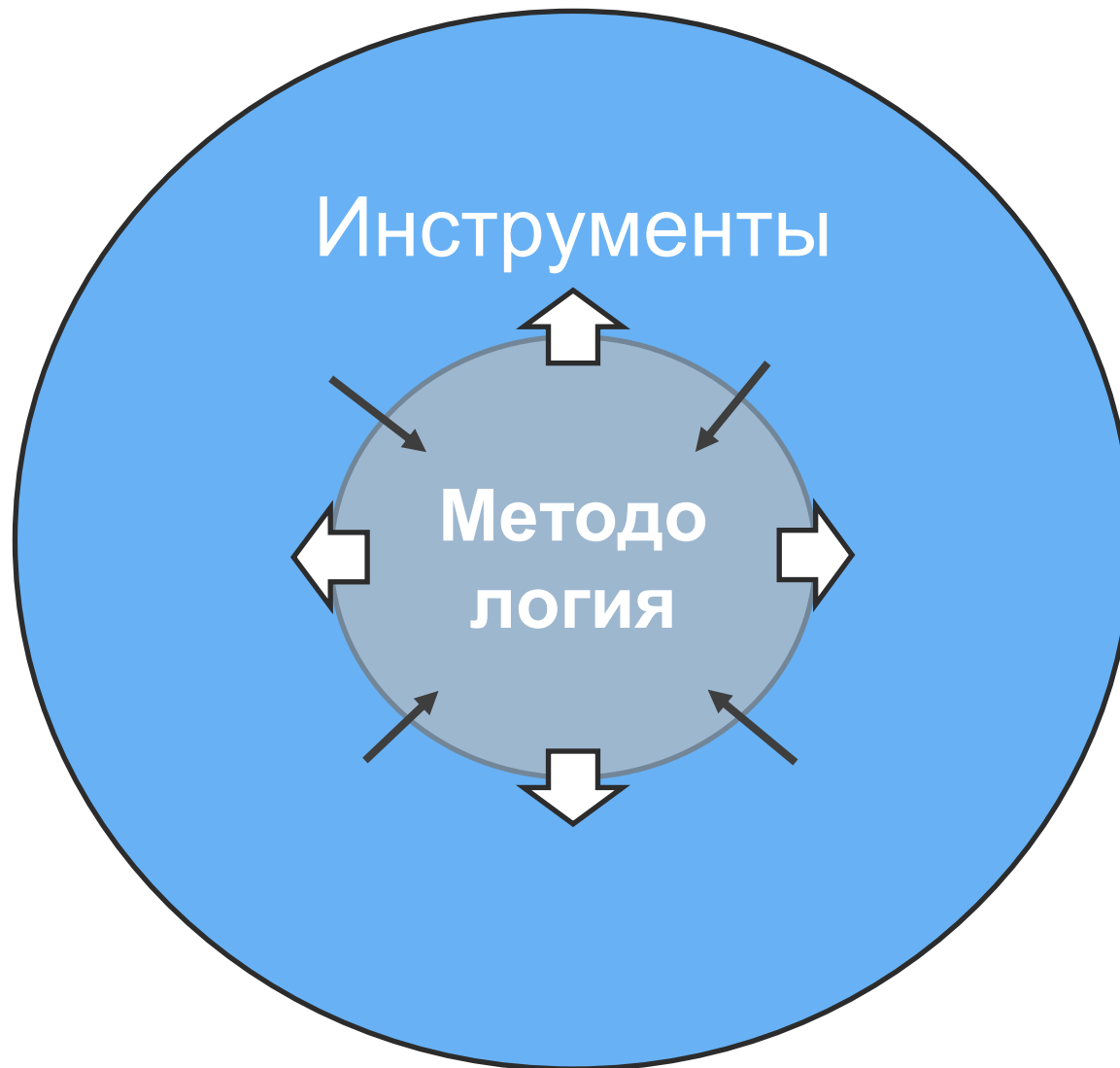
О курсе

Сайт

<http://ccfit.nsu.ru/~den>

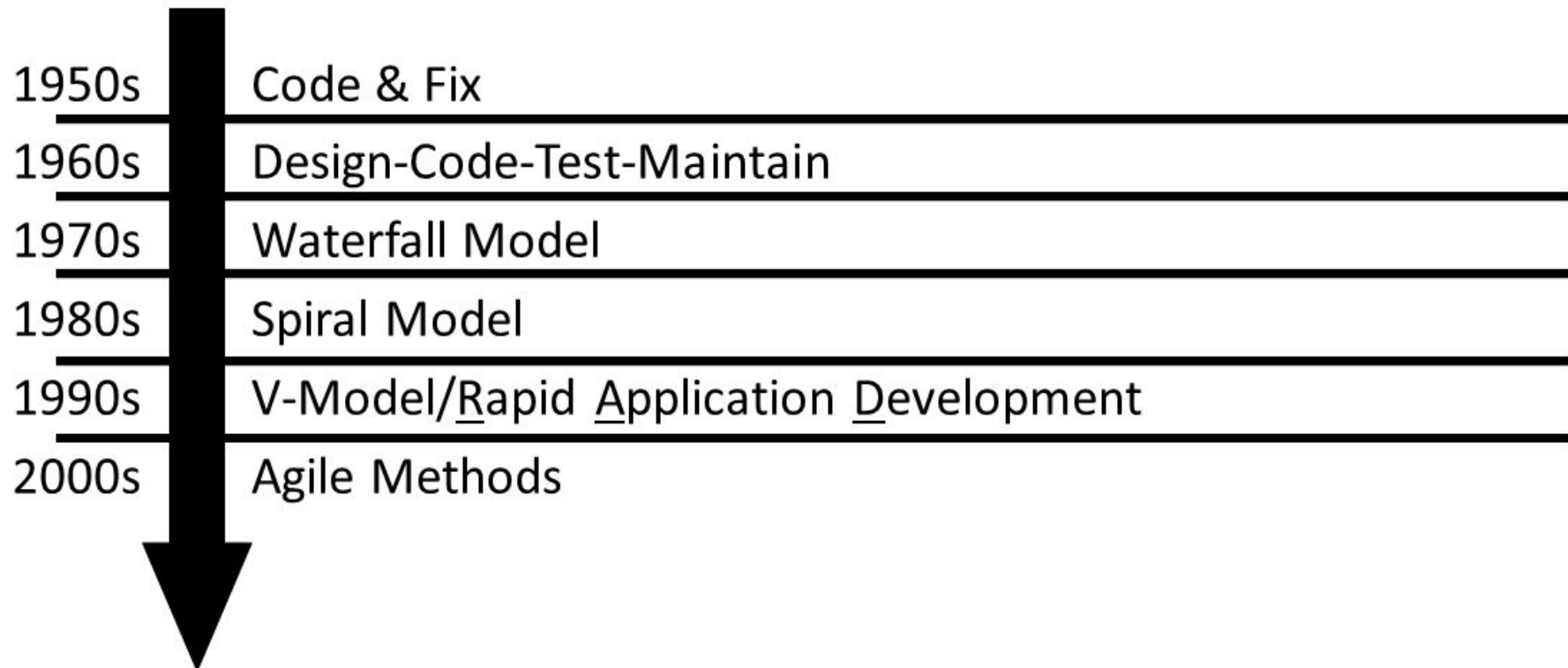
Контроль

1. Дифференцированный зачет
2. Оценка по практическим занятиям (учебный проект)
3. Учитывается посещение лекций
4. В спорных ситуациях экзаменует и ставит оценку лектор

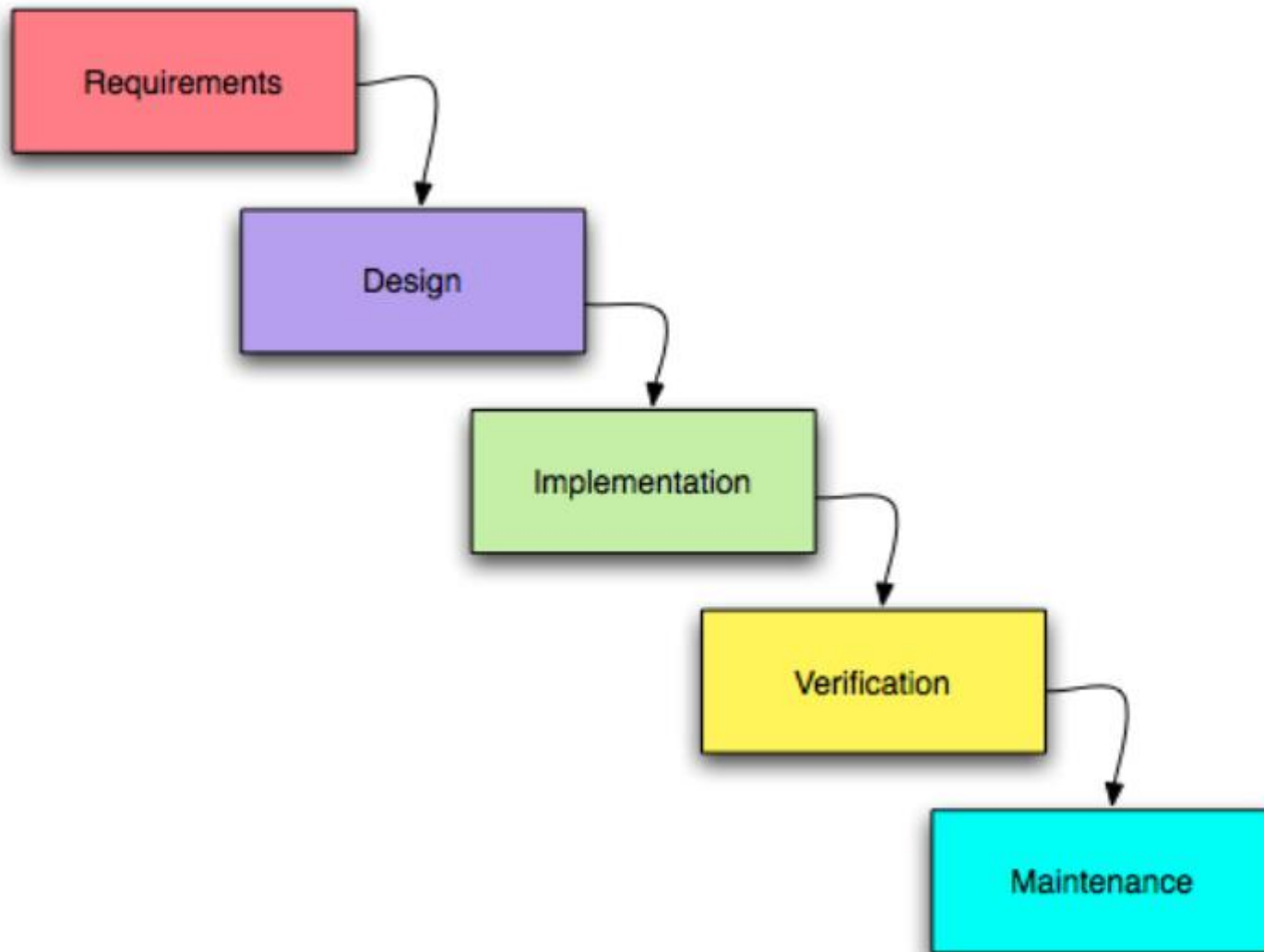


Методологии разработки

- Каскадная модель (waterfall)
- Спиральная модель
- Итеративная модель, RUP (Rational Unified Process)
- Гибкая разработка (Agile manifesto): Scrum, Kanban



Каскадная модель



Спиральная модель



Инкрементная модель

Инкрементная модель



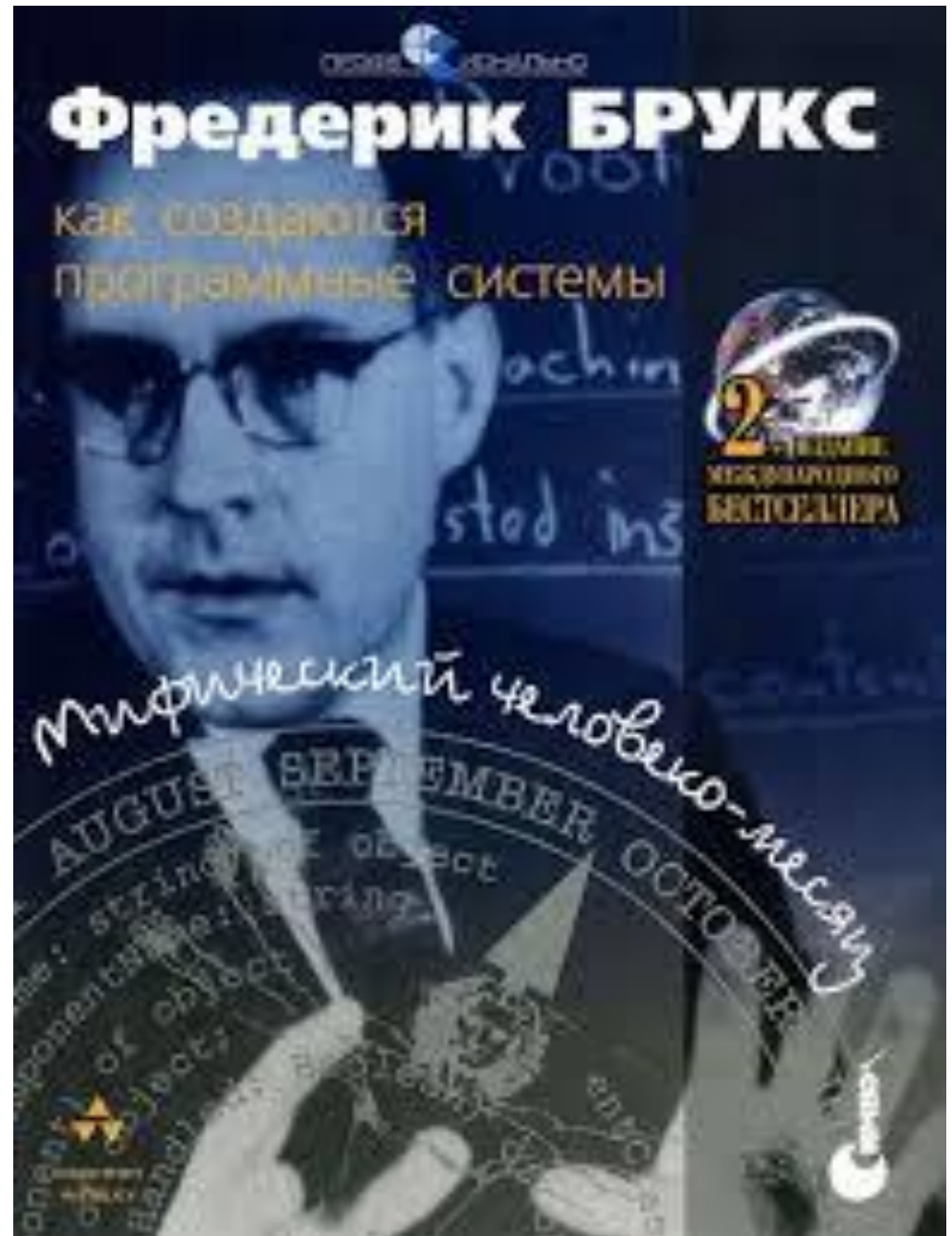
Итеративная модель



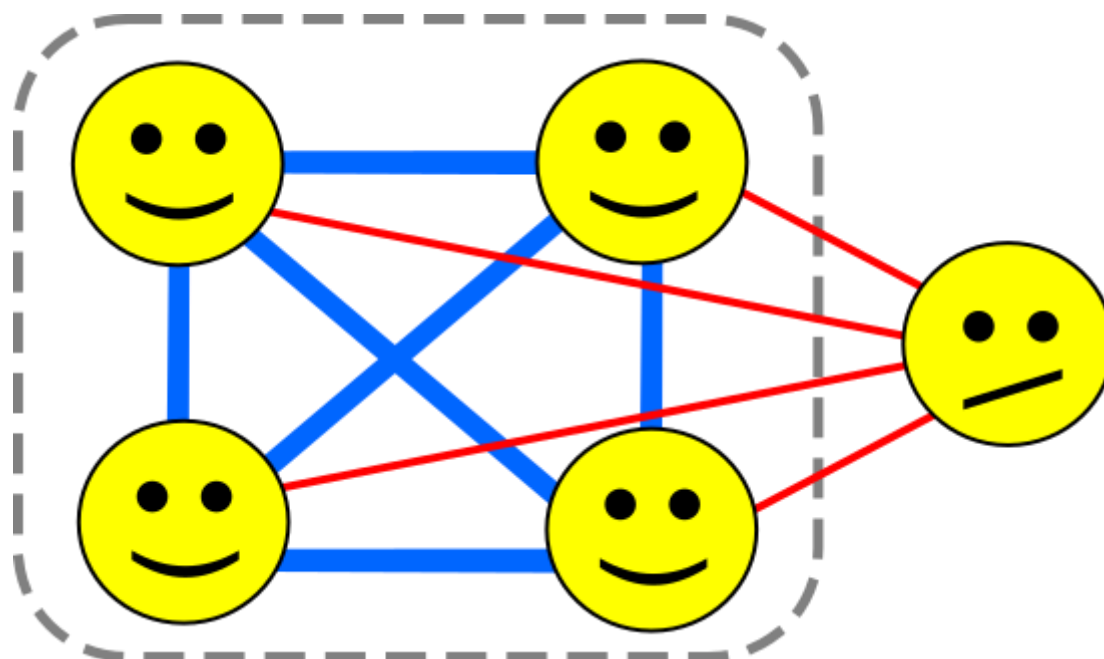
Закон Брукса

«Чтобы родить ребенка требуется девять месяцев независимо от того, сколько женщин привлечено к решению данной задачи. Многие задачи программирования относятся к этому типу, поскольку отладка по своей сути носит последовательный характер»

«Программные проекты чаще проваливаются из-за нехватки календарного времени, чем по всем остальным причинам вместе взятым.»

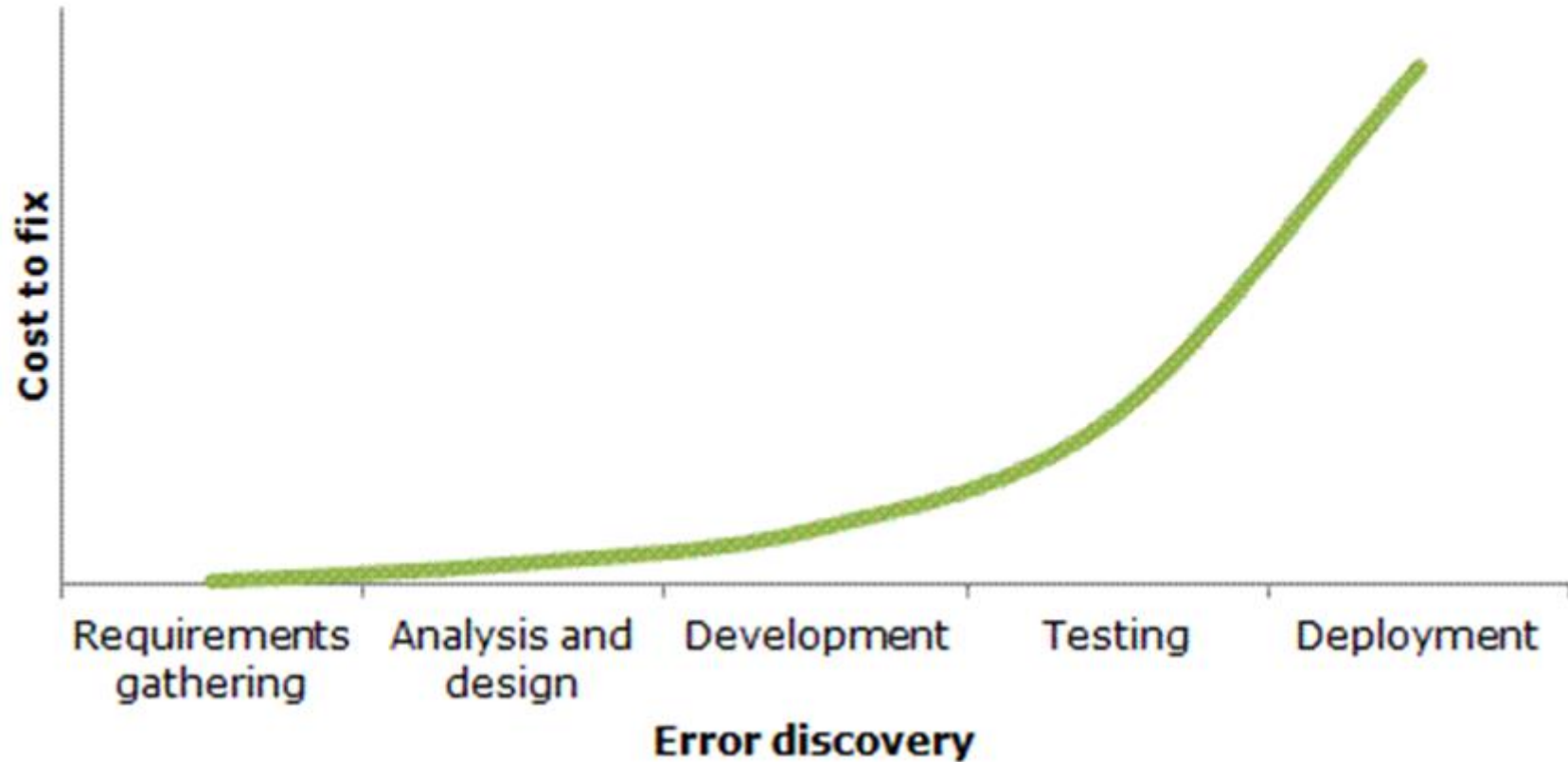


Закон Брукса #2: если проект не укладывается в сроки, то добавление рабочей силы задержит его ещё больше



Результат командной разработки ПО напрямую зависит от того, насколько эффективно участники команды взаимодействуют друг с другом

Закон Боэма

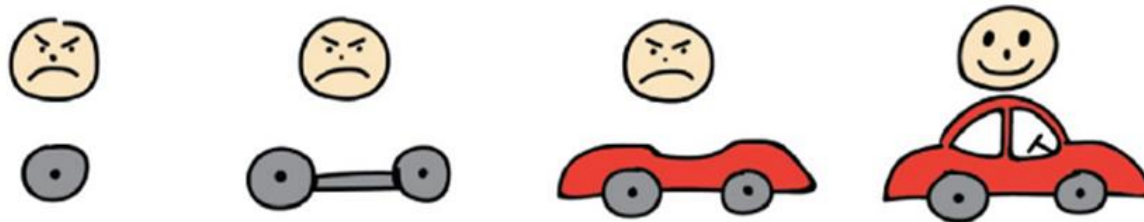


Чем быстрее мы обнаруживаем ошибки, тем эффективнее мы разрабатываем ПО

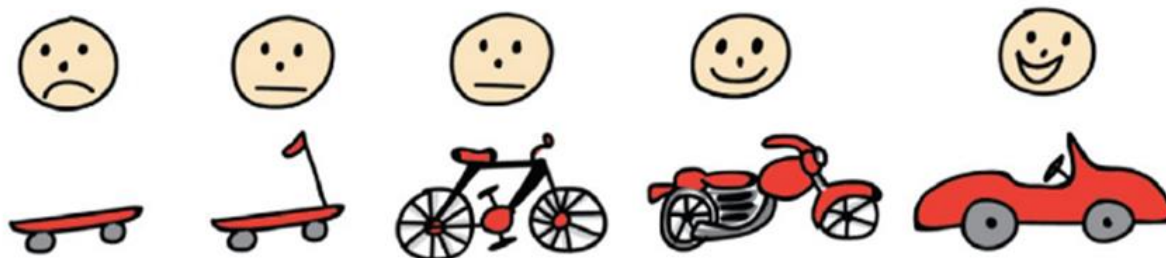
Закон Галла

Любая работающая сложная система развивается на базе работающей простой системы. Сложные системы, созданные с нуля, никогда не будут работать в реальном мире, поскольку в процессе разработки на них не влияли факторы отбора, присущие среде

Not like this...



Like this!

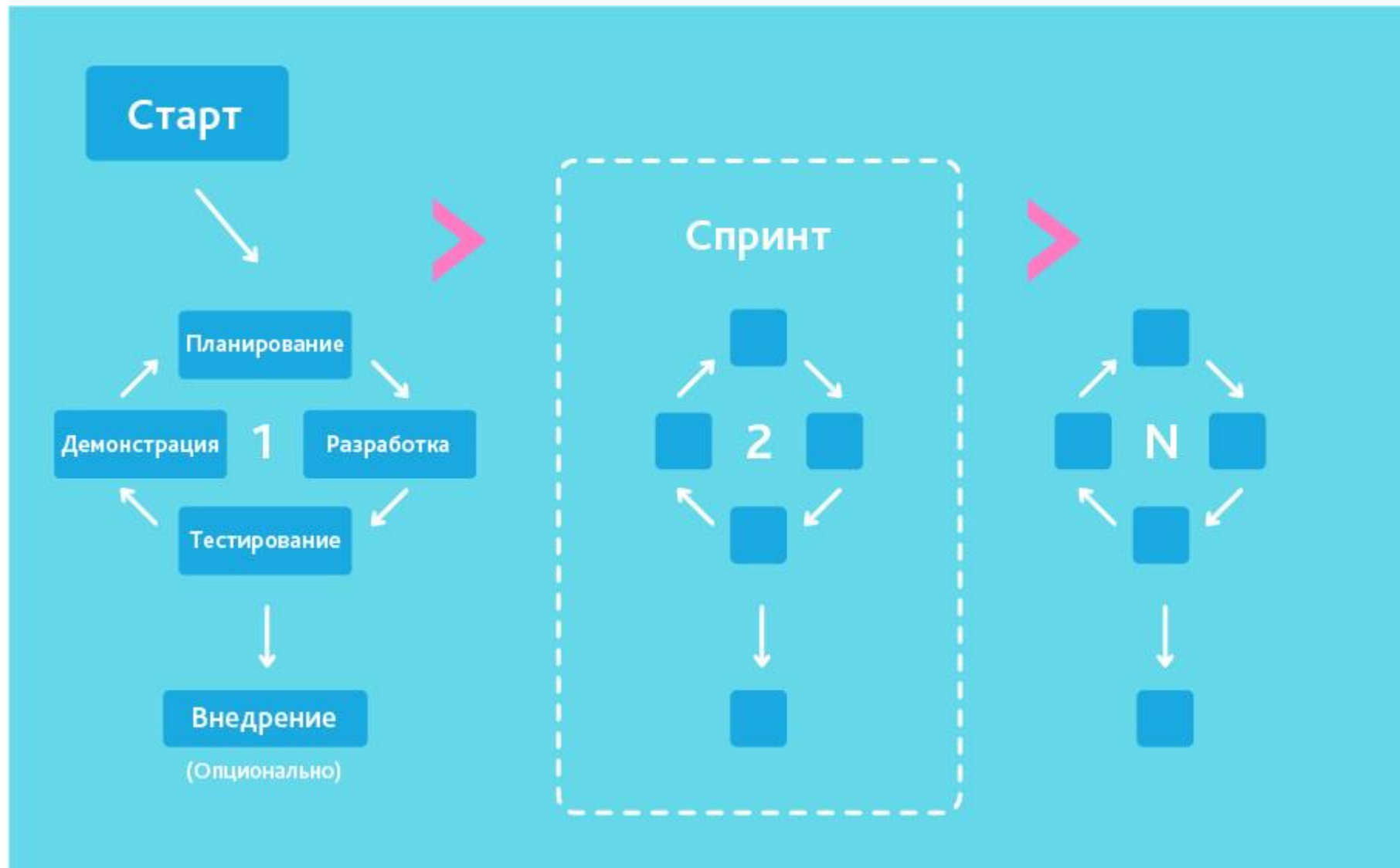


MVP

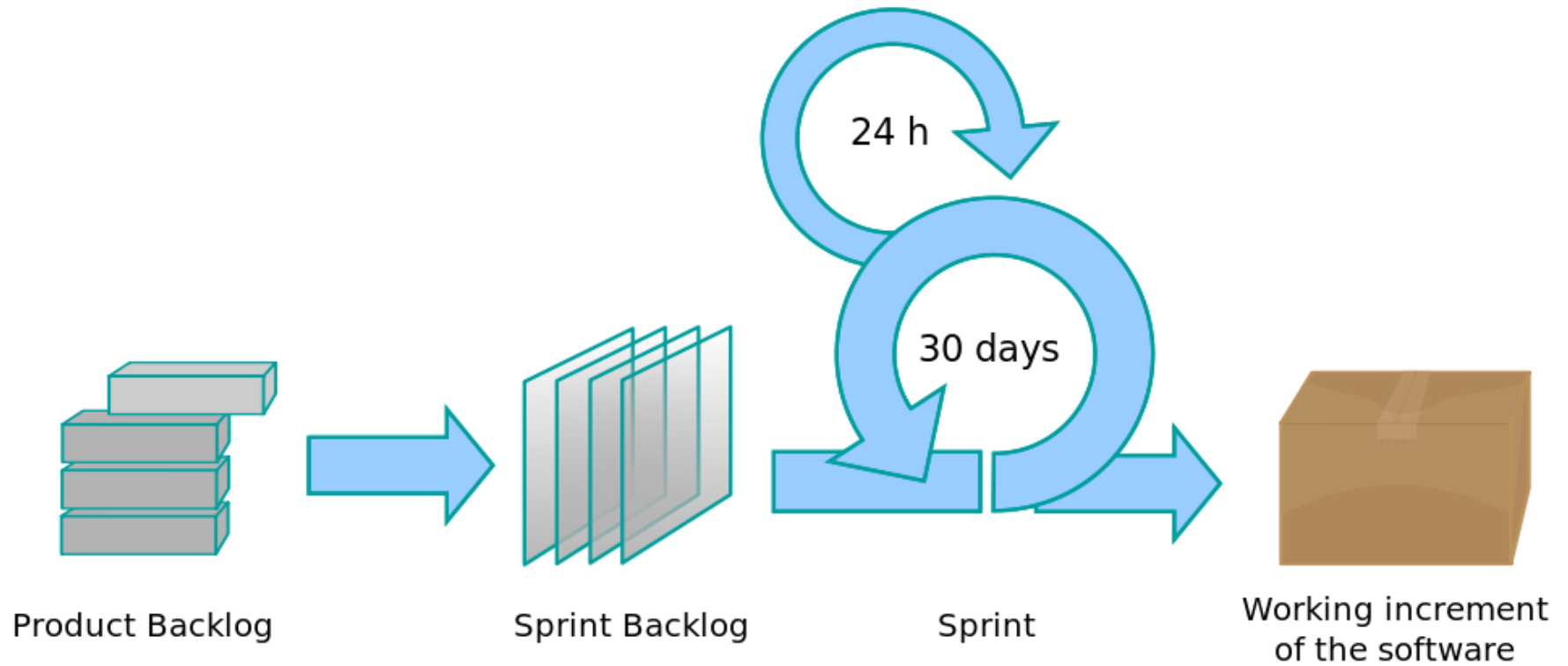
Старт разработки с **MVP** (minimum viable product) дает возможность оперативно протестировать идею и при необходимости изменить функциональность.

Например, первая версия сервиса Uber содержала только две простые функции: вызов водителя и оплата поездки банковской картой.

Гибкая разработка



Scrum



Agile манифест (2001)

Мы постоянно открываем для себя более совершенные методы разработки программного обеспечения, занимаясь разработкой непосредственно и помогая в этом другим.

Благодаря проделанной работе мы смогли осознать, что:

Люди и взаимодействие важнее процессов и инструментов

Работающий продукт важнее исчерпывающей документации

Сотрудничество с заказчиком важнее согласования условий контракта

Готовность к изменениям важнее следования первоначальному плану

То есть, не отрицая важности того, что справа, мы всё-таки ценим то, что слева.

12 принципов гибкой разработки ПО

1. Customer satisfaction by rapid delivery of useful software
2. Welcome changing requirements, even late in development
3. Working software is delivered frequently (weeks rather than months)
4. Close, daily cooperation between business people and developers
5. Projects are built around motivated individuals, who should be trusted
6. Face-to-face conversation is the best form of communication (co-location)
7. Working software is the principal measure of progress
8. Sustainable development, able to maintain a constant pace
9. Continuous attention to technical excellence and good design
10. Simplicity—the art of maximizing the amount of work not done—is essential
11. Self-organizing teams
12. Regular adaptation to changing circumstances

12 принципов гибкой разработки ПО

1. Наивысший приоритет — это удовлетворение заказчика при помощи частых и непрерывных поставок ценного для него программного обеспечения
2. Изменения в требованиях приветствуются даже на поздних этапах реализации проекта
3. Работающий продукт следует выпускать каждые несколько недель, в крайнем случае — каждые несколько месяцев
4. Наиболее эффективный и действенный способ передачи информации — это встреча членов команды разработки ПО
5. Представители бизнеса и команда разработки должны работать над проектом вместе
6. Проекты строятся вокруг мотивированных людей
7. Рабочее программное обеспечение — это главная мера прогресса проекта
8. Непрерывающаяся деятельность, держать постоянную скорость разработки
9. Постоянное внимание к техническому совершенству и качественной архитектуре способствует гибкости
10. Простота — это искусство не делать лишней работы
11. Лучшая архитектура, требования и дизайн создаются в самоорганизующихся командах
12. Команда постоянно ищет способы стать более эффективной путем настройки и коррекции своих действий

Компания с высокой концентрацией таланта – это место, где хотят работать все. Звезды любят, когда их окружают другие звезды.

Разделите все должности в организации на технические («операционные») и творческие («креативные»). Назначьте творческим сотрудникам максимально высокую зарплату. Возможно, для этого придется нанять одного профессионала высочайшей квалификации вместо десятка ничем не примечательных работников.

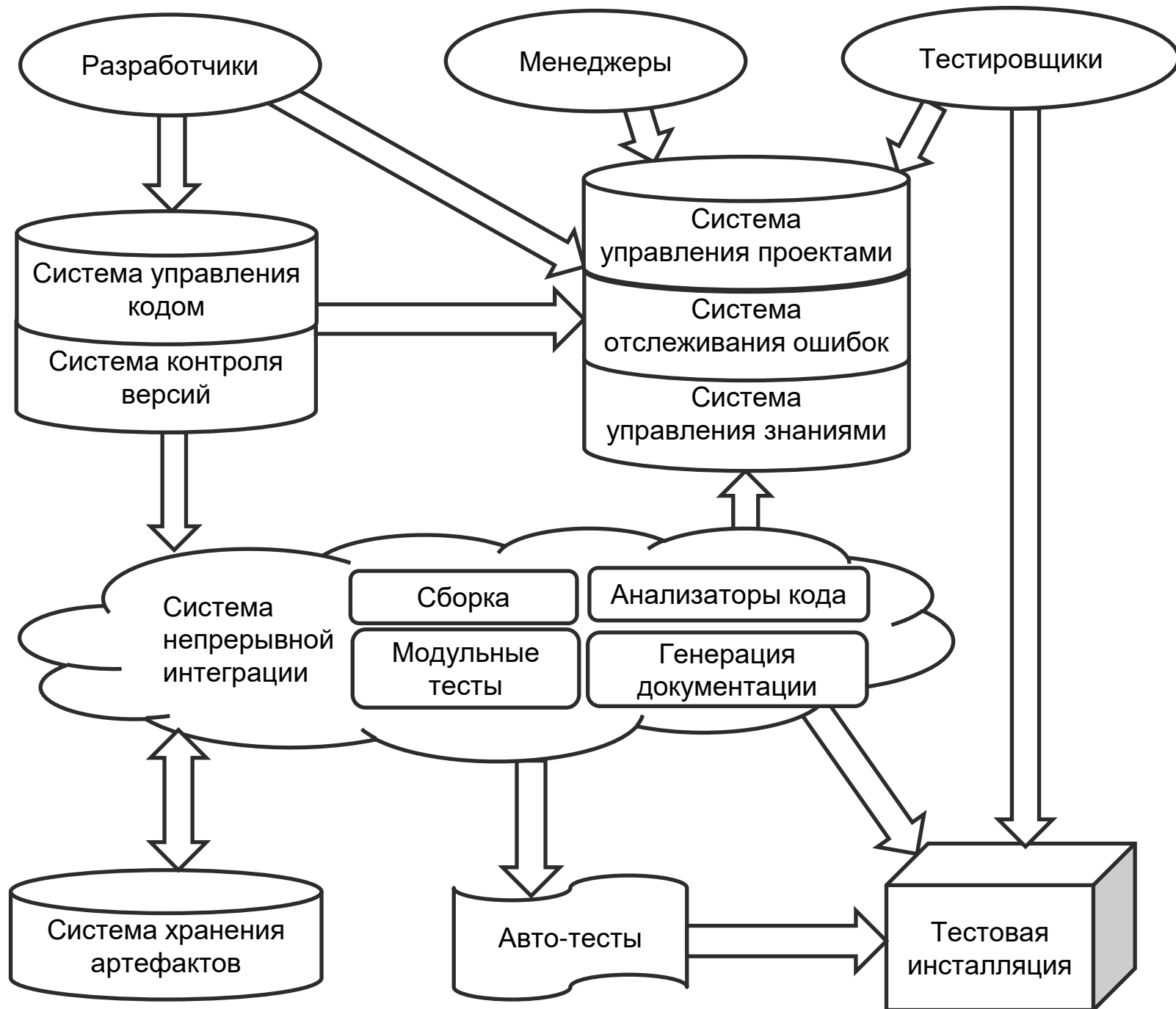


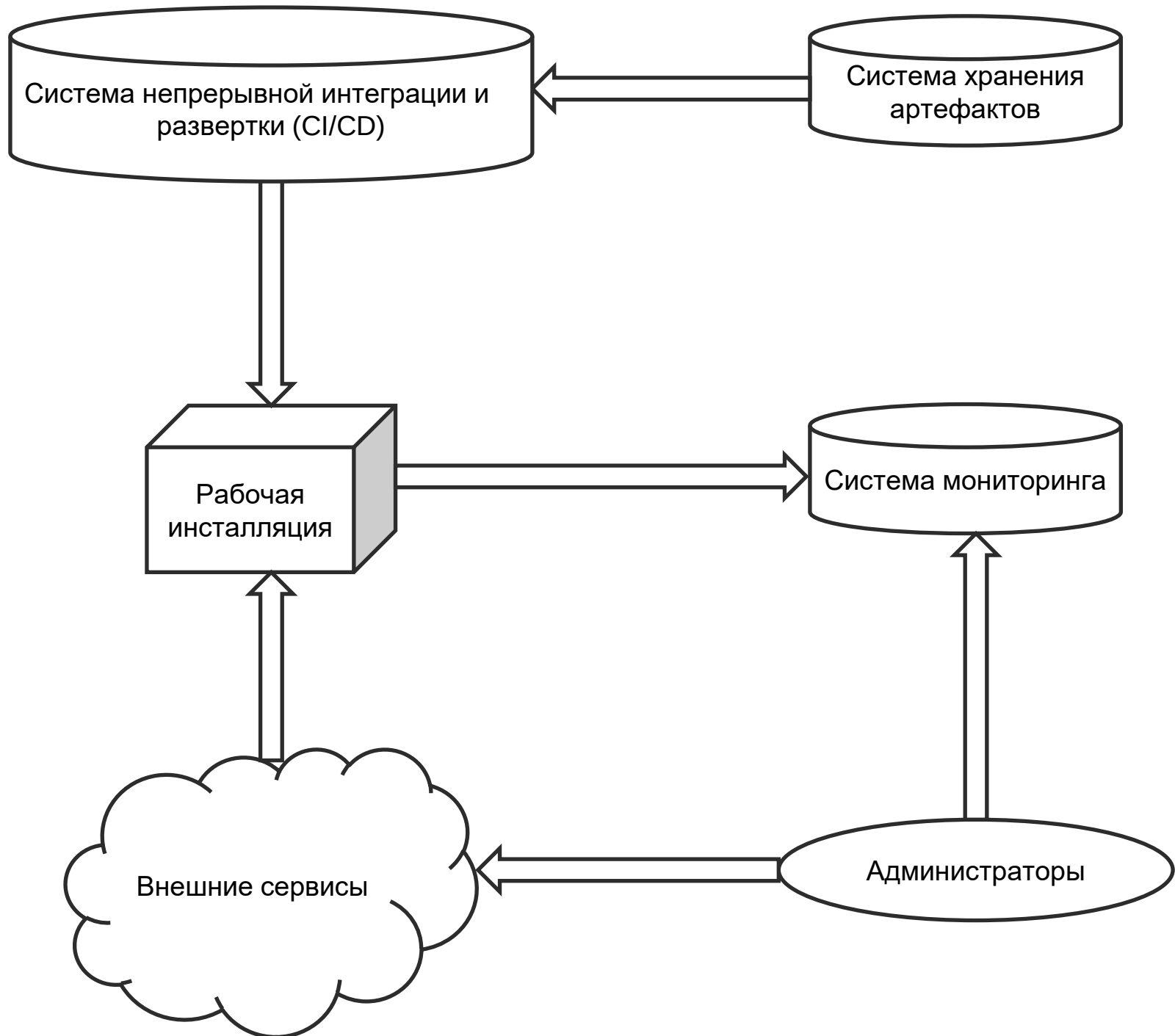
От графика отпусков мы
избавились в 2003 году

Теперь, когда вы повысили
концентрацию таланта и
устранили каждое слабое
звено, пора внедрять принцип
полной откровенности.

Мы представляли культуру,
которая ставила во главу
угла человека, а не метод.
Наш подход отдавал
приоритет инновациям, а не
эффективности и при этом
не был обременен
механизмами контроля.







Системы контроля версий



Системы отслеживания ошибок



YouTRACK

Системы сборки



make



maven



Непрерывная интеграция



Travis CI



TeamCity

Тестирование

JUnit



Управление знаниями



JavaDoc



Управление конфигурацией

ANSIBLE



 puppet

Виртуализация



Big Data



◁EROSPIKE-

