



ОНЛАЙН-ОБРАЗОВАНИЕ

Меня хорошо слышно && видно?



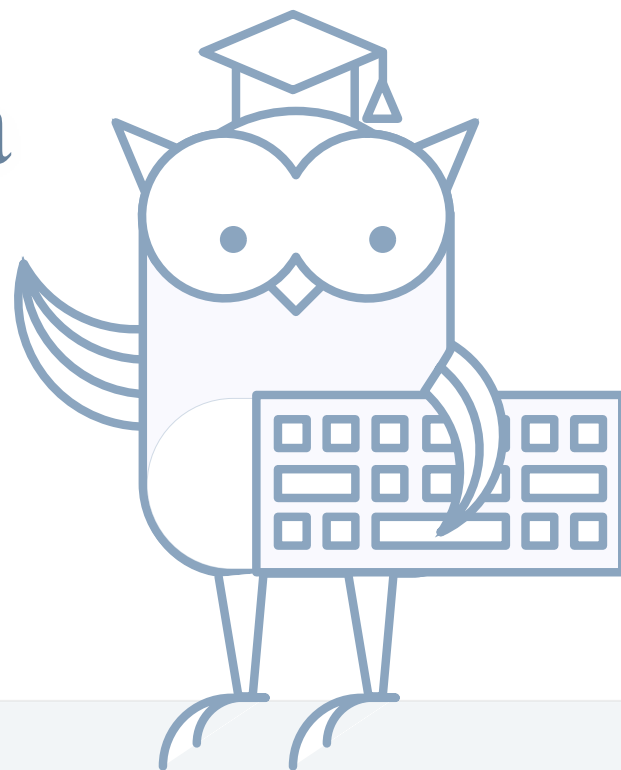
Напишите в чат, если есть проблемы!

Ставьте  если все хорошо



Командная разработка

PHP. Модуль 4



- Поговорить про процессы
- Обсудить методологии и фреймворки
- Рассмотреть полезные инструменты

01

Организация совещаний/встреч

We will continue having
lots of meetings until
we find out why no
work is getting done.



someecards
user card

- Запланированная
- Есть координатор
- Есть четкий результат
- Помогает команде достичь целей

- Обозначить цели встречи
- Приготовить повестку
- Разослать повестку всем участникам

- Распределить роли
- Придерживаться повестки
- Записывать

- Разослать краткую сводку всем участникам
- Назначит ответственных и дедлайны для всех задач

02

Процессы

- Требования
- Анализ
- Дизайн
- Создание
- Тестирование

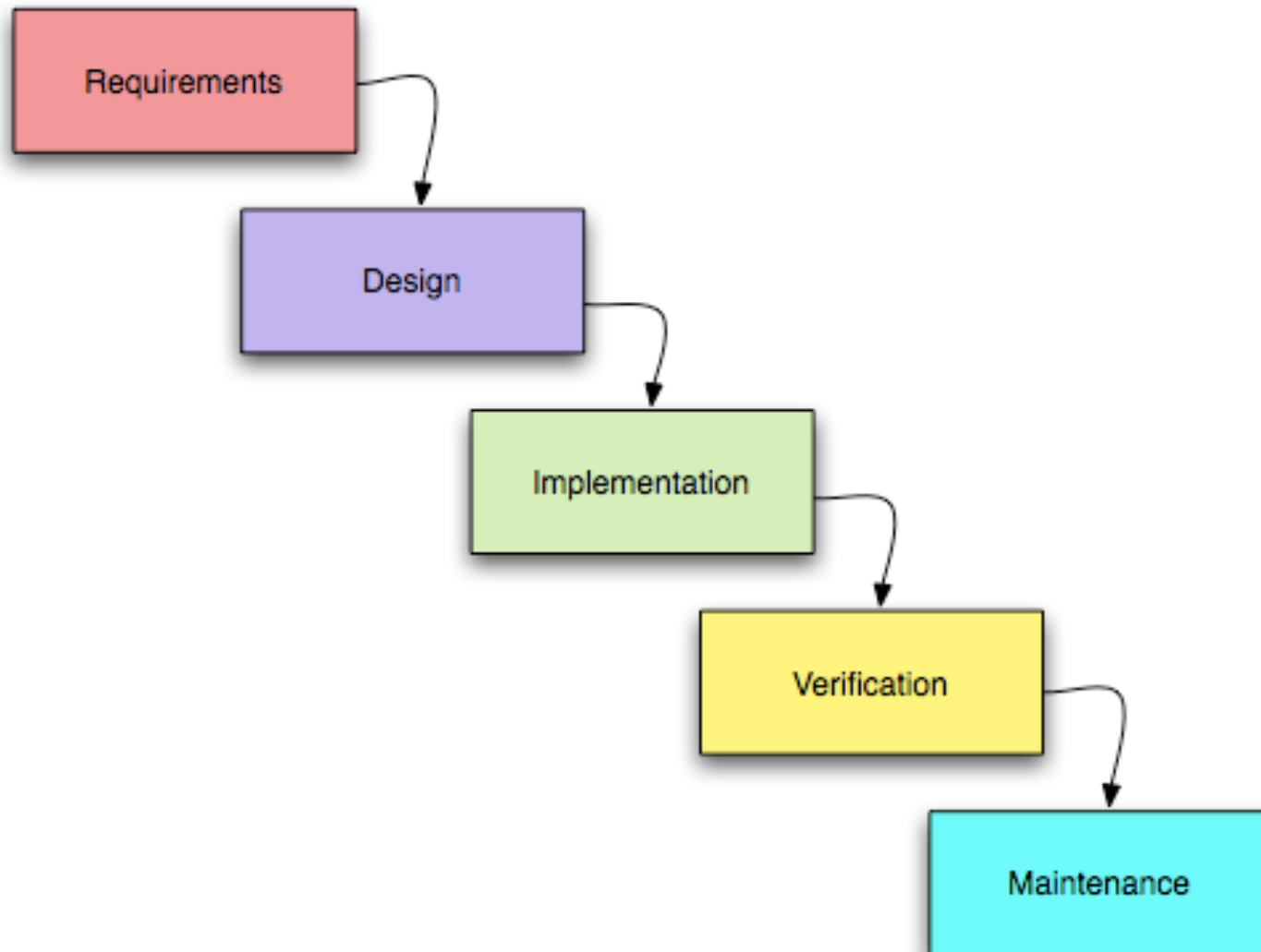
- Жизненный цикл
- Фреймворк
- Инстанцирование

- Водопад
- Итеративный
- Agile
- Etc.

- V-model
- Kanban
- Scrum
- Etc.

03

Водопад



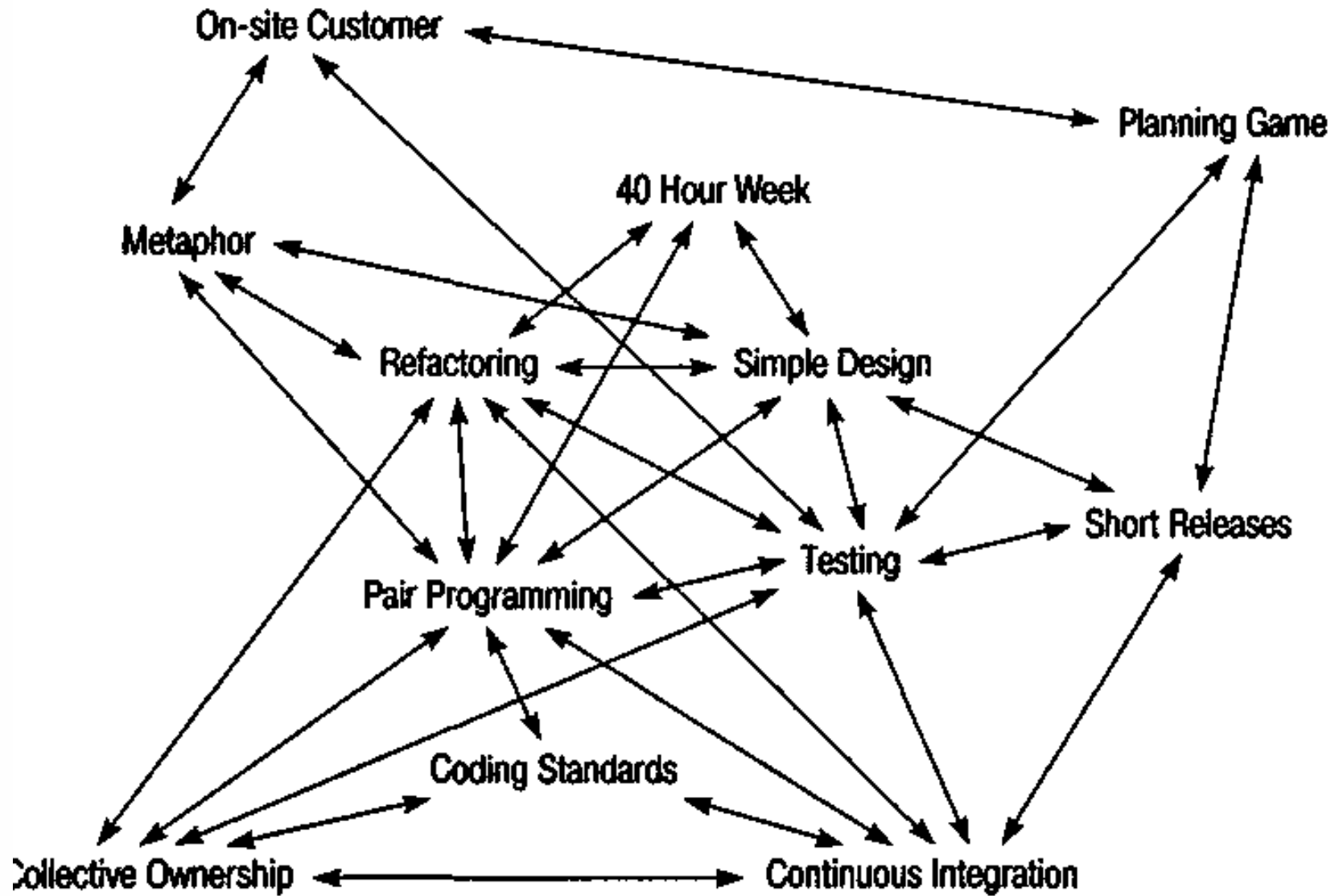
04

Agile

- Люди и их взаимодействие **важнее** процессов и инструментов;
- Готовый продукт **важнее** документации по нему;
- Сотрудничество с заказчиком **важнее** жестких контрактных ограничений;
- Реакция на изменения **важнее** следования плану

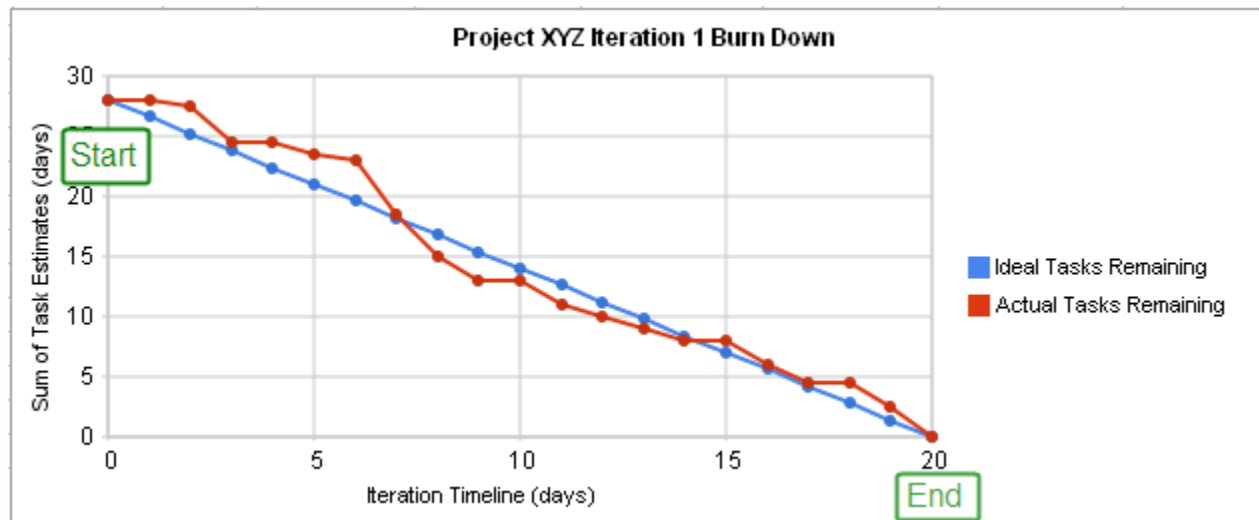
- Коммуникация
- Простота
- Быстрая обратная связь
- Смелость (менять код)

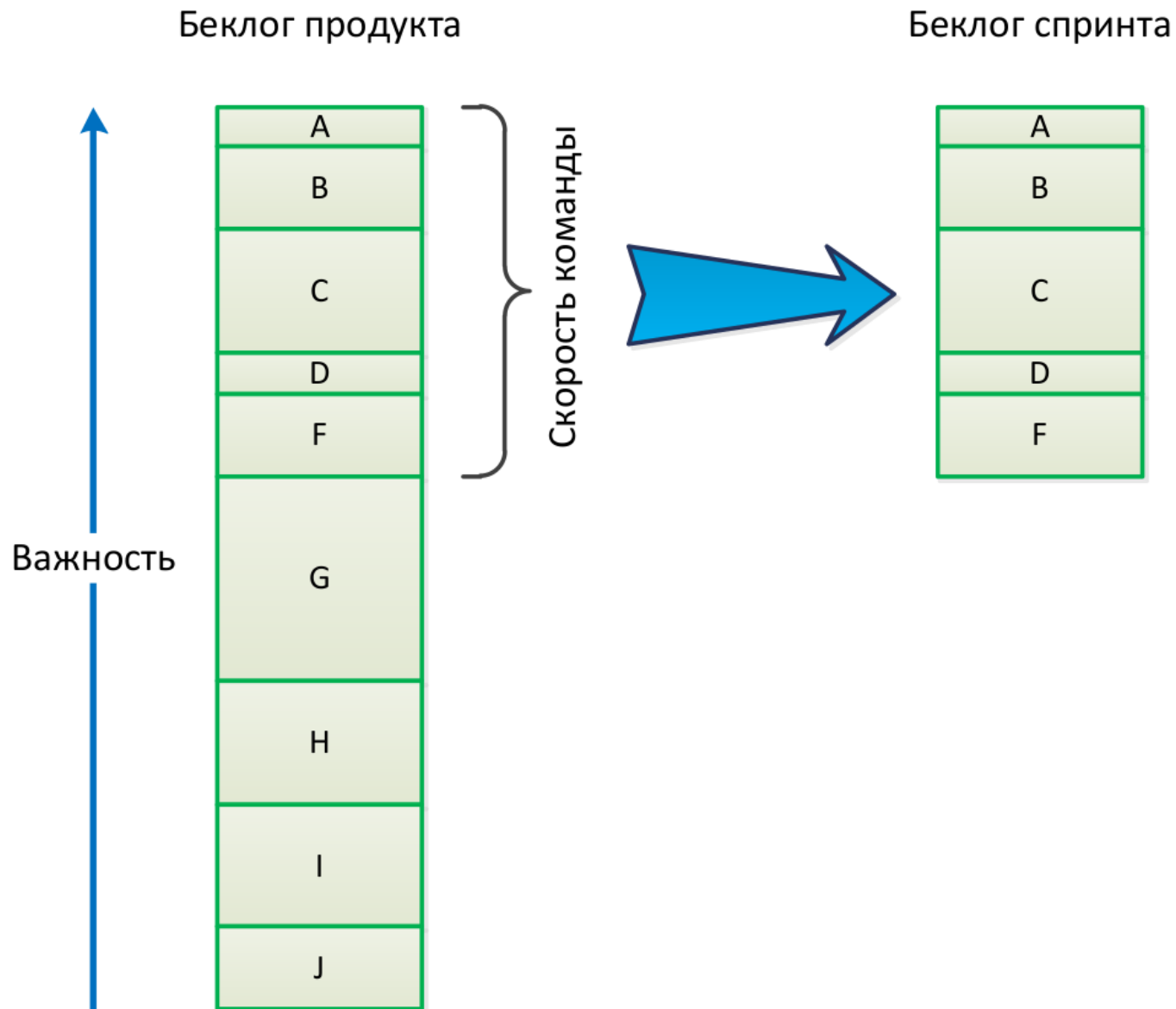
- TDD
- CI
- Парное программирование
- Рефакторинг



- Ориентирован больше на менеджмент
- Хорошо сочетается с практиками XP

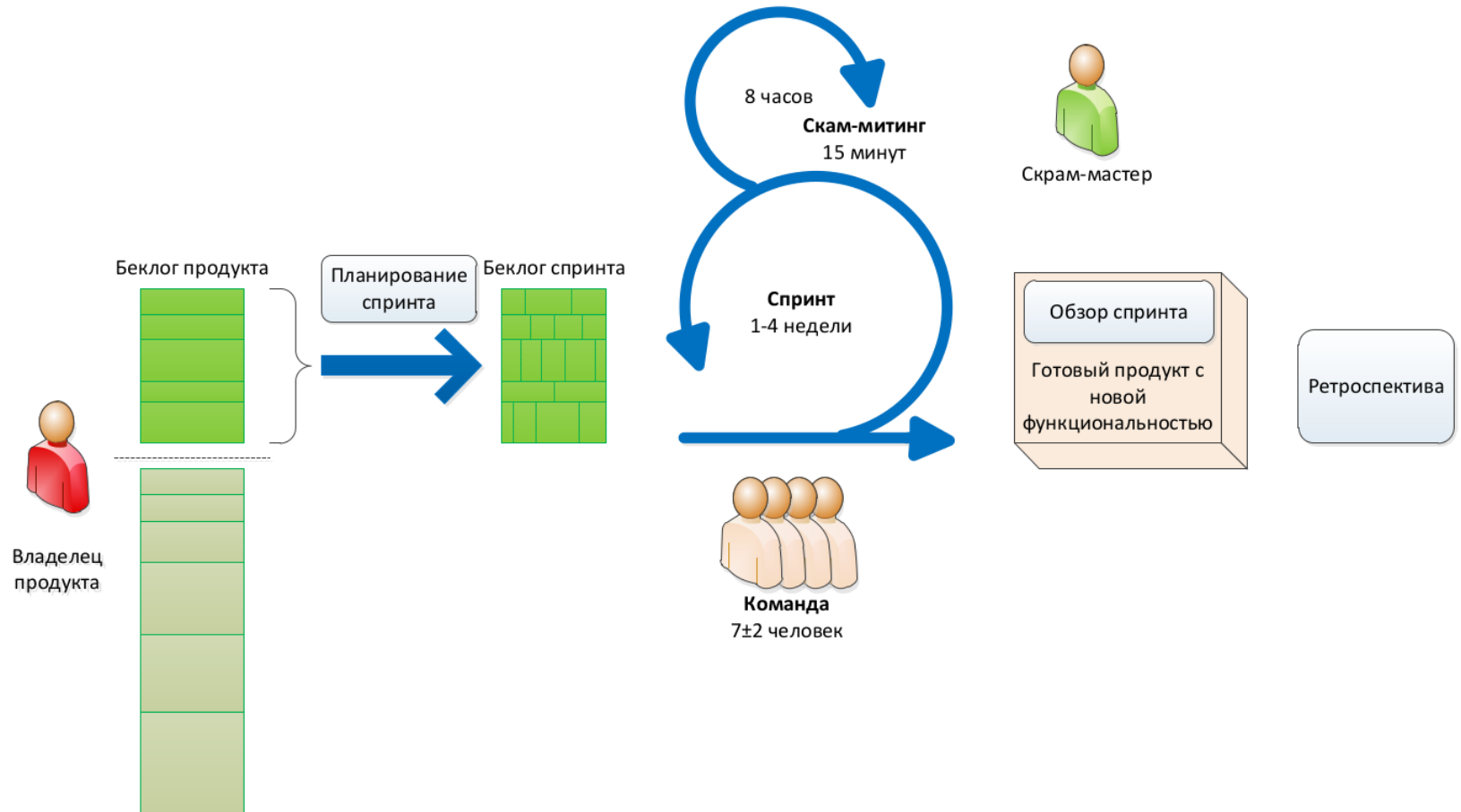
- Бэклог продукта
- Бэклог спринта
- Burn down chart

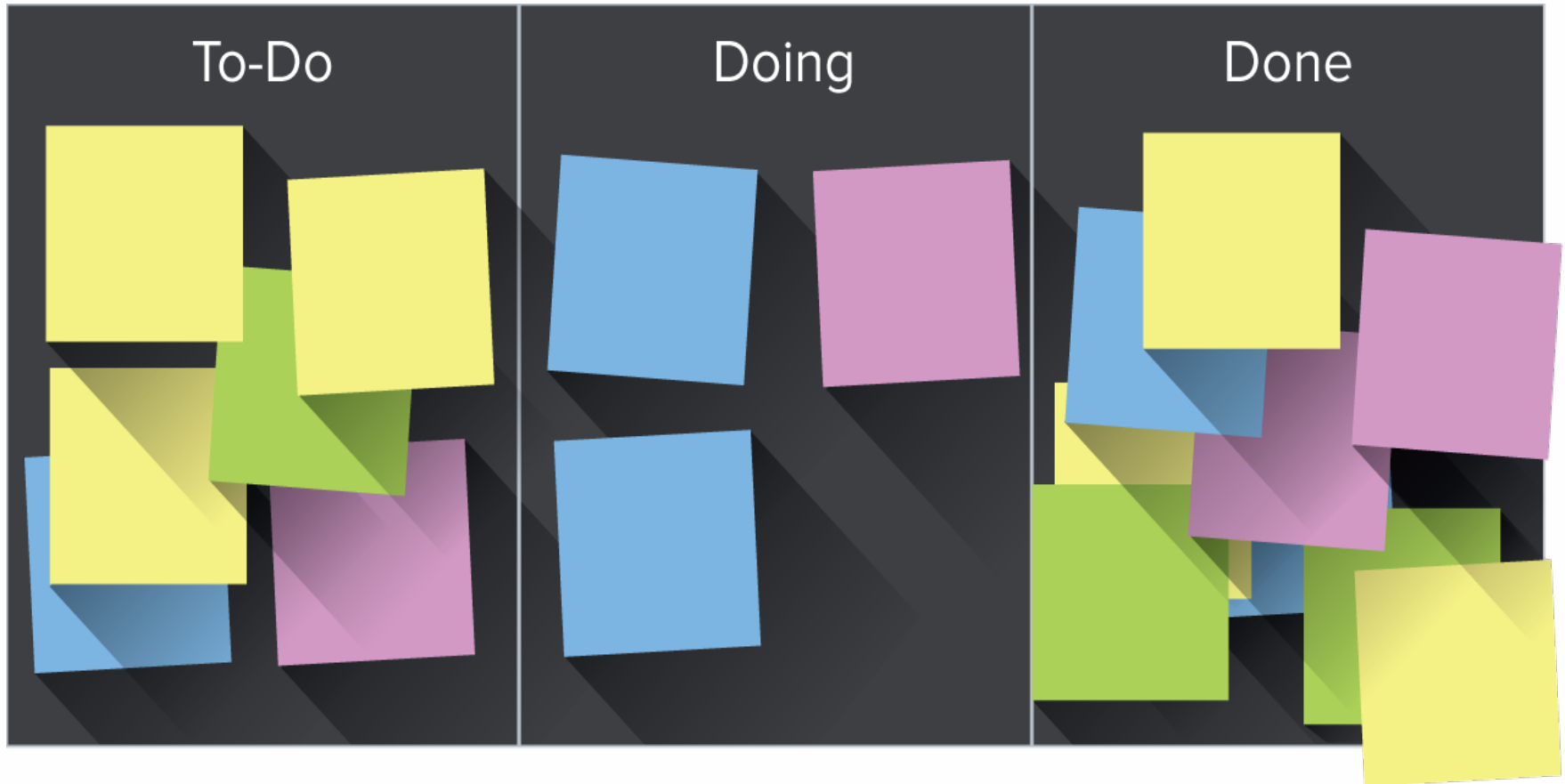




- Product owner
- Scrum master
- Команда

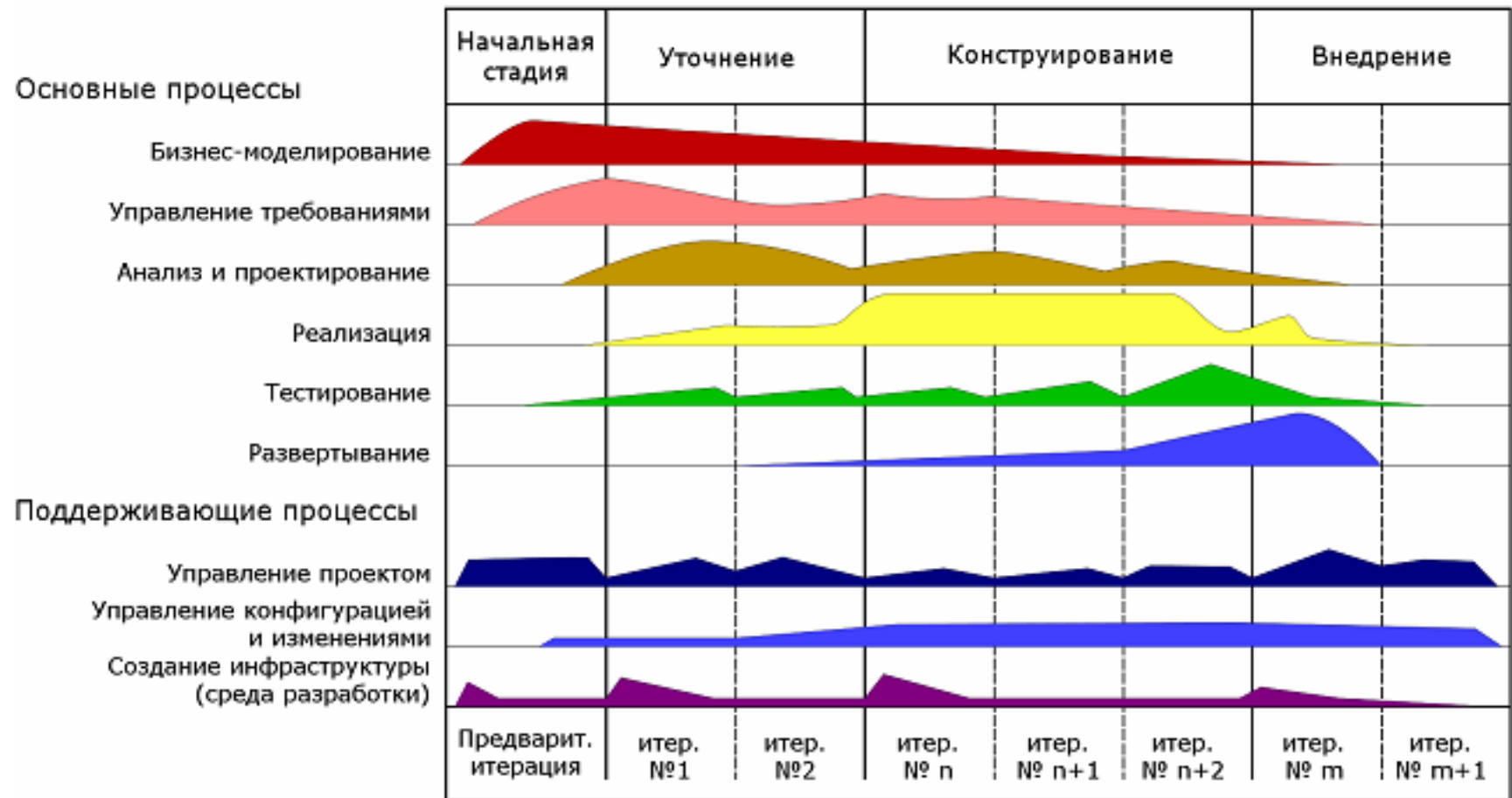
- Планирование
- Стендап
- Демо
- Ретроспектива





Рабочие процессы

Стадии

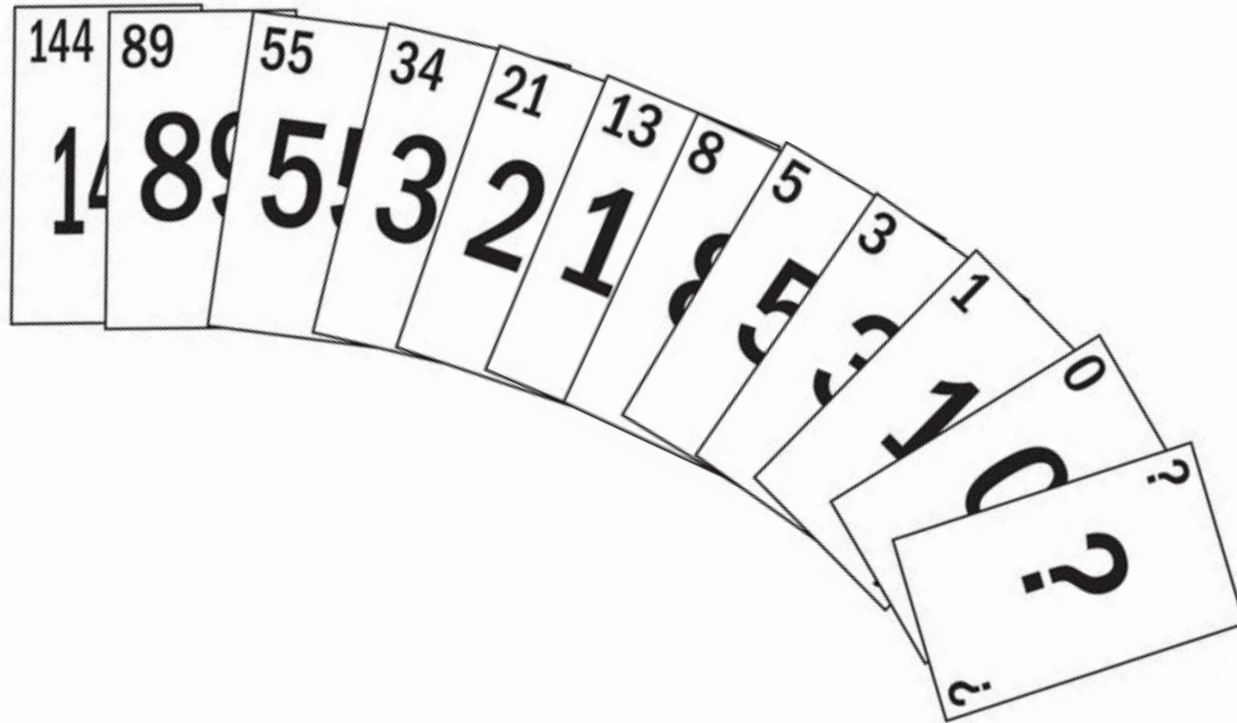


Итерации

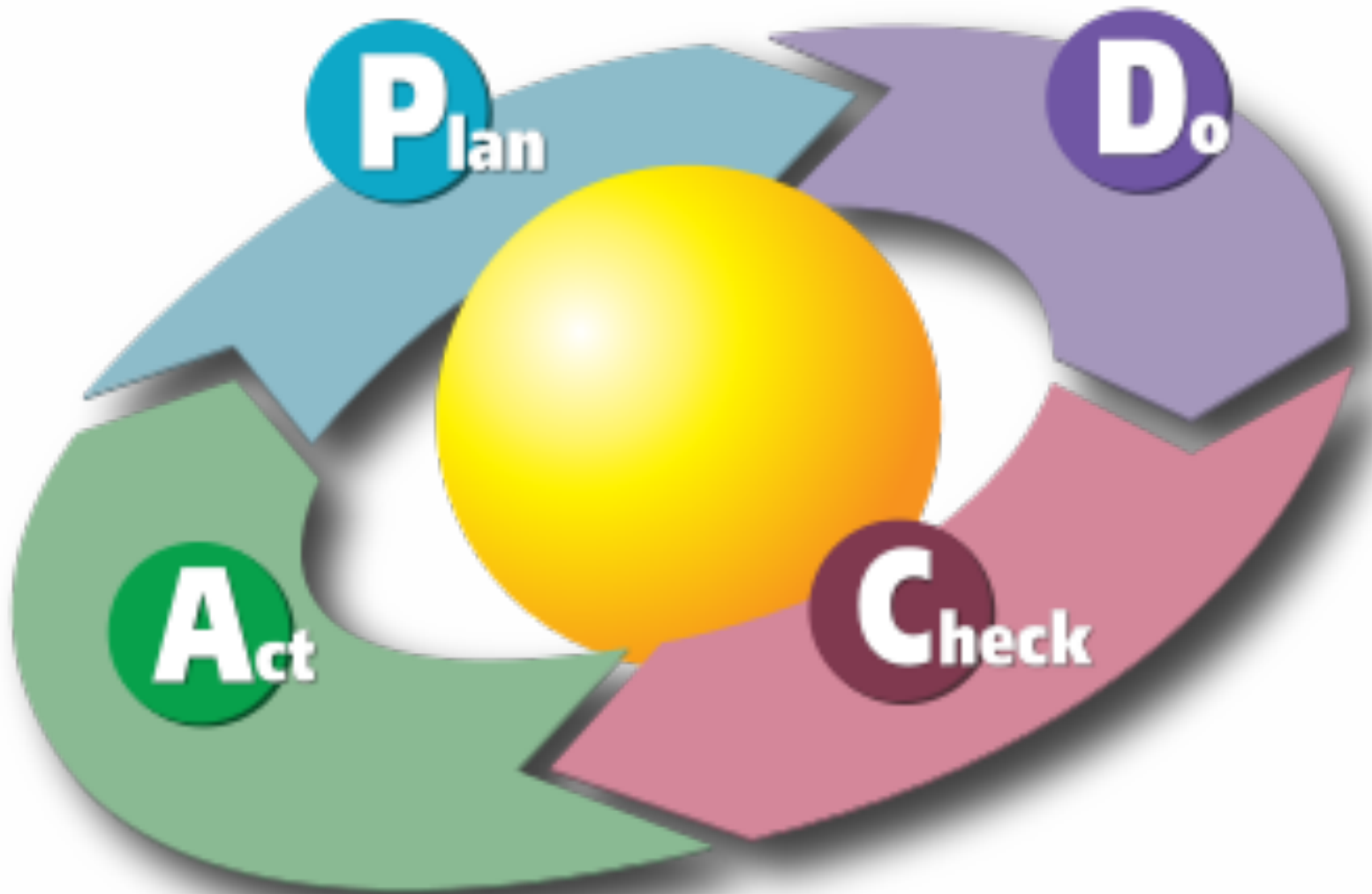
- 26 ролей
- Use cases
- UML
- Архитектура – часть процесса

05

Инструменты







06

Работа с VCS

- После каждого коммита система в рабочем состоянии
- Реверт коммита должен оставить систему в рабочем состоянии
- Коммит должен делать только одну вещь

<https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>

- A develop branch is created from master
- A release branch is created from develop
- Feature branches are created from develop
- When a feature is complete it is merged into the develop branch
- When the release branch is done it is merged into develop and master
- If an issue in master is detected a hotfix branch is created from master
- Once the hotfix is complete it is merged to both develop and master



Игорь Саханков

Email: sakhankov@gmail.com

Telegram: [@isakhankov](https://www.telegram.me/isakhankov)