


ЖИЗНЕННЫЙ ЦИКЛ СОЗДАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ



Постоев Дмитрий
postoev@okbsapr.ru

Часть 2:

Модели и методологии разработки

Определения

Модель (парадигма) разработки – абстрактная концепция, определяющая порядок стадий разработки, а также критерии перехода от одной стадии к другой. ^[1]

Методология разработки – способ разделения процесса разработки на стадии, сфокусированный на том, как должен выполняться каждый из этапов (какие данные использует, как разделяются функции, как организовано управление, задаются требования и т.д.). ^[2]

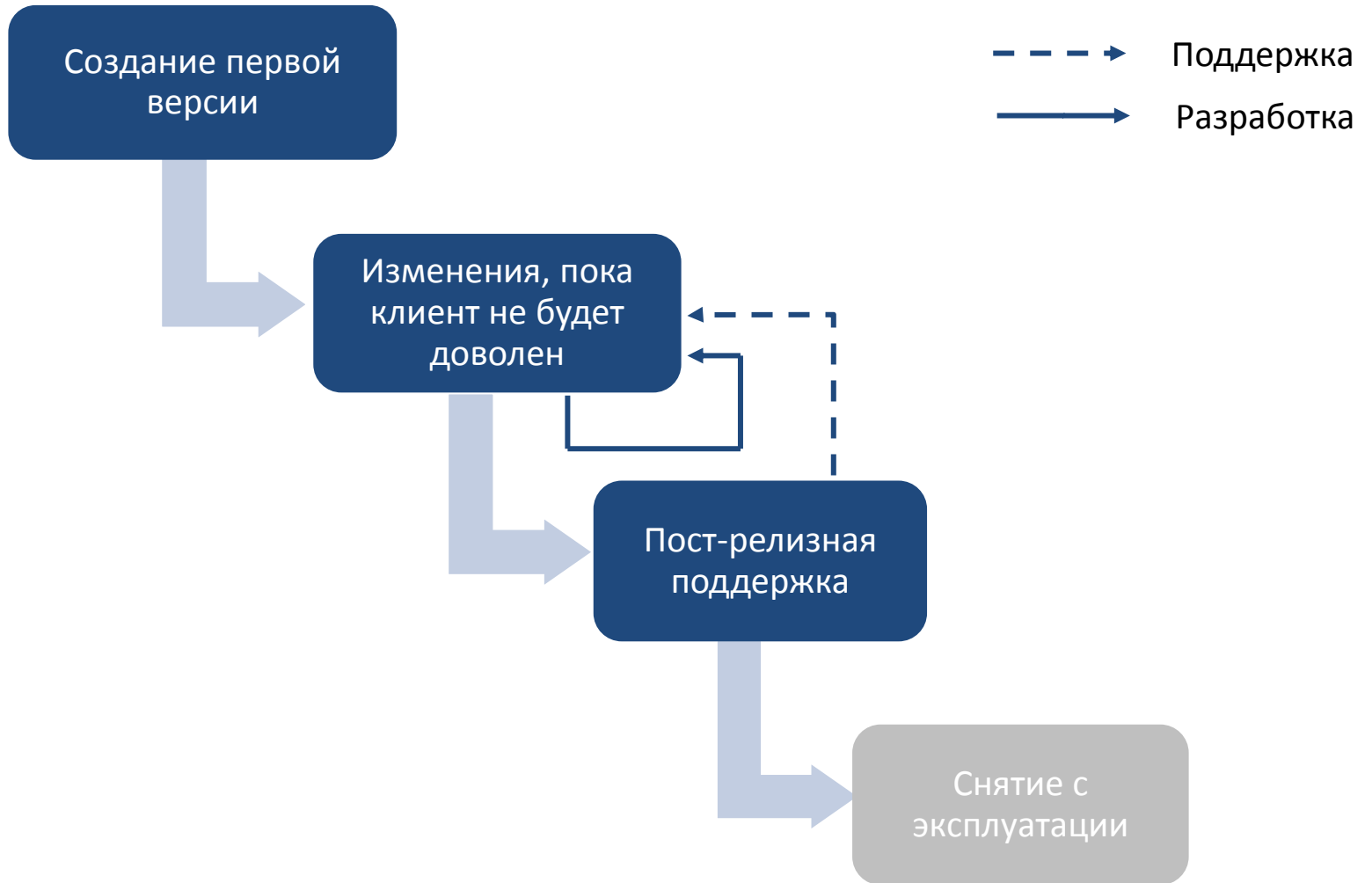
[1] Centers for Medicare & Medicaid Services (CMS) Office of Information Service (2008). Selecting a development approach

[2] A Spiral Model of Software Development and Enhancement Barry W. Boehm, TRW Defense Systems Group

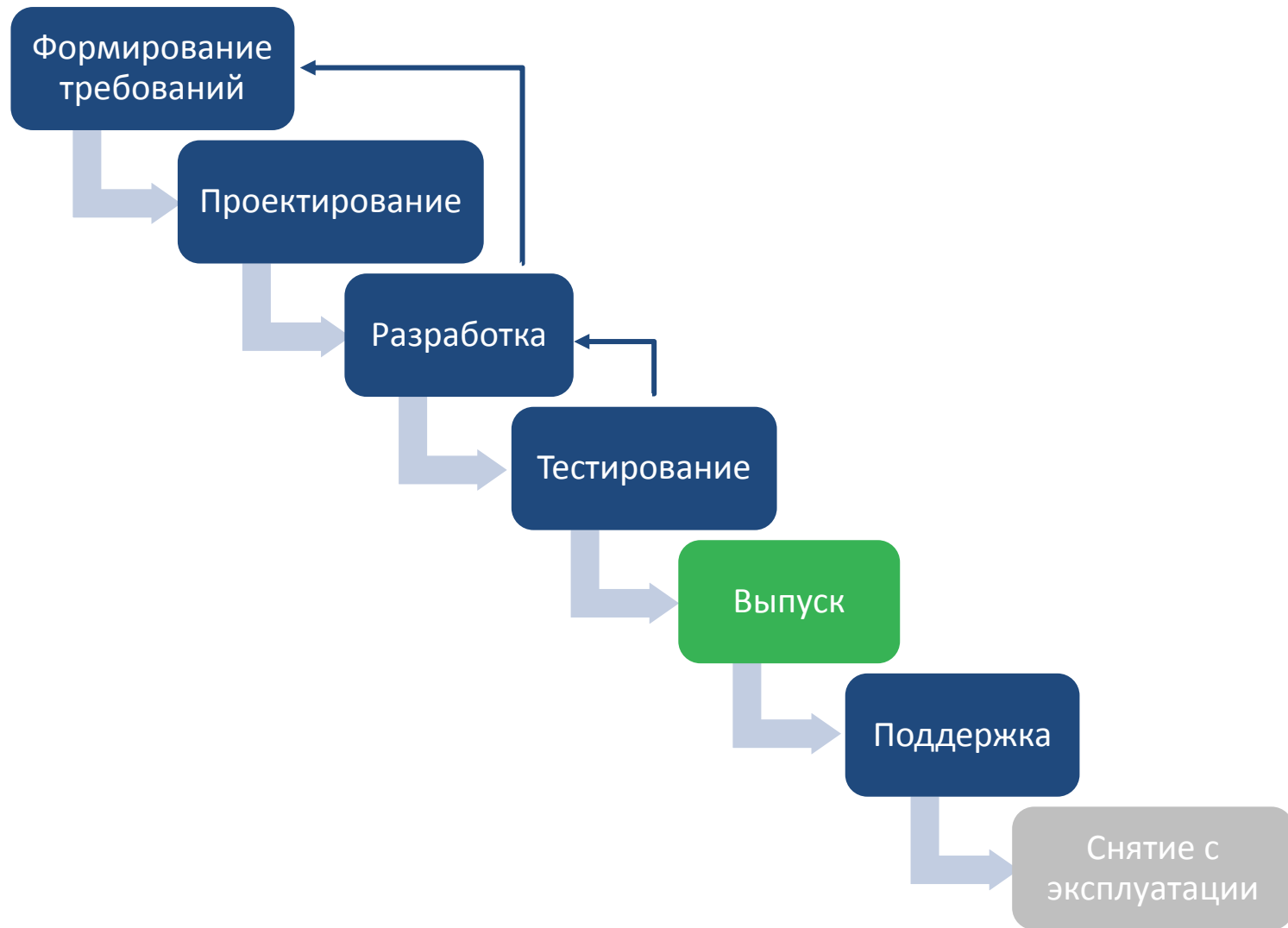
Основные модели разработки

1. Code-and-Fix
2. Каскадная
3. Модели постепенных улучшений
 - a) Инкрементальная
 - b) Итеративная
 - c) Итеративно-инкрементальная
4. Спиральная
5. Agile (гибкие методологии)
 - a) Scrum
 - b) Kanban
 - c) Экстремальное программирование (XP)
 - d) Lean Startup

Code-and-Fix

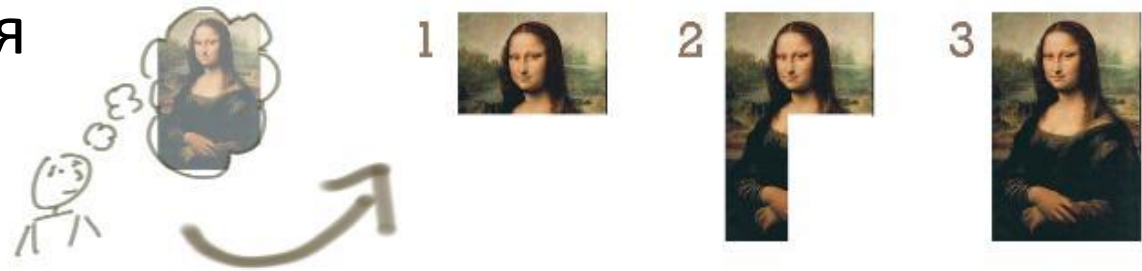


Каскадная модель



Модели постепенных улучшений

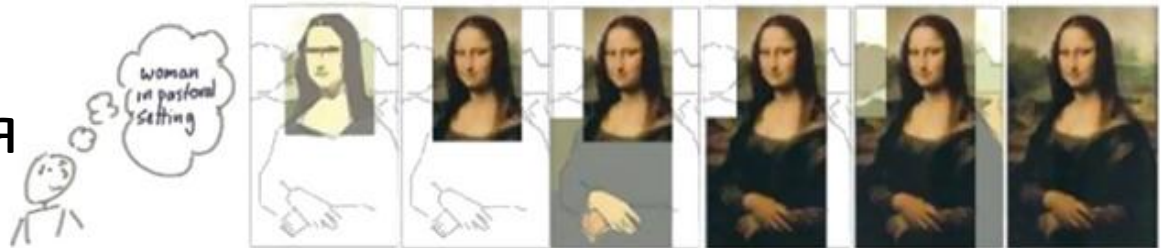
Инкрементальная



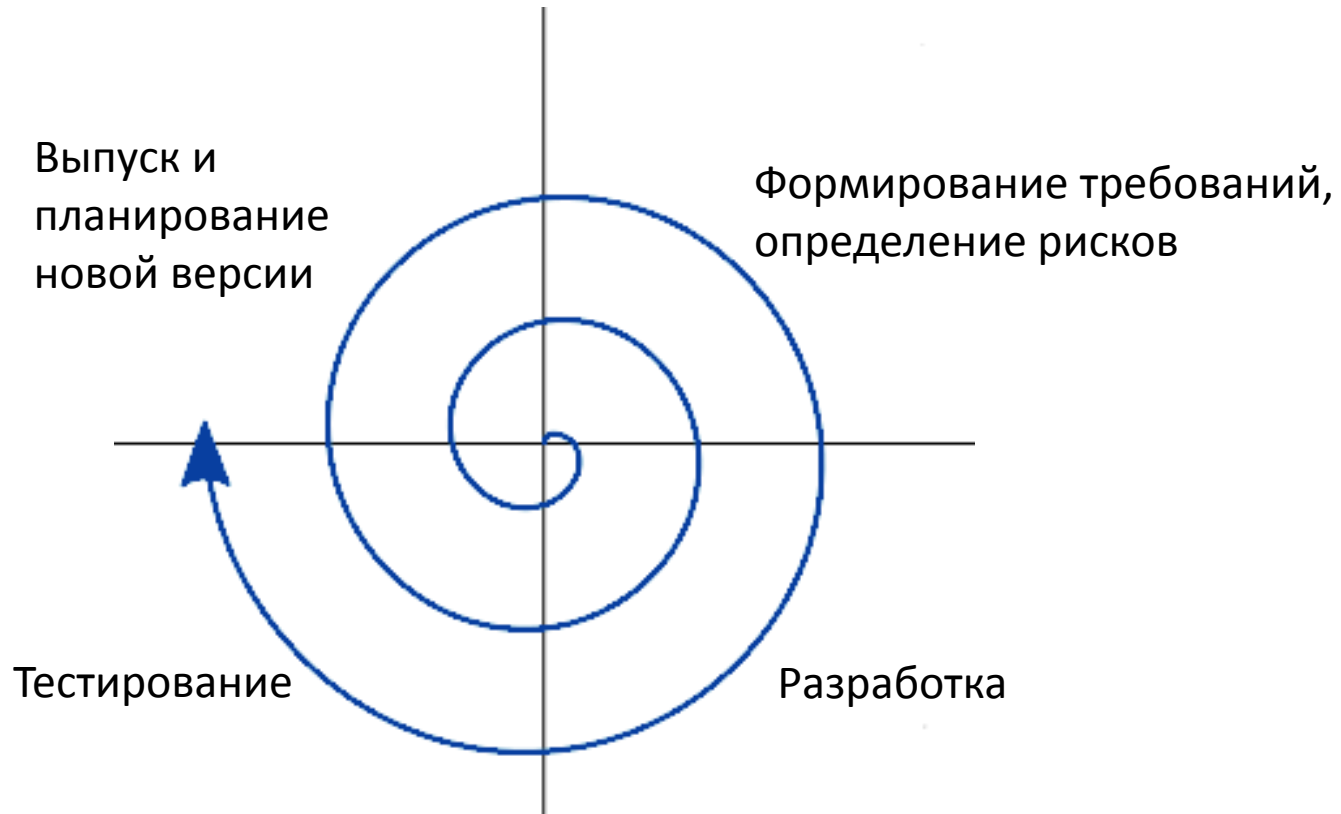
Итерационная



Итерационно-инкрементальная



Спиральная модель



Agile (гибкие методологии)

Гибкая методология разработки – серия подходов к разработке программного обеспечения, ориентированных на использование итеративной разработки, динамическое формирование требований и обеспечение их реализации в результате постоянного взаимодействия внутри самоорганизующихся рабочих групп, состоящих из специалистов различного профиля.^[1]



[1] “What is Agile Software Development?”. Agile Alliance

Agile Manifesto

1. **Люди и взаимодействие** важнее процессов и инструментов
2. **Работающий продукт** важнее исчерпывающей документации
3. **Сотрудничество с заказчиком** важнее согласования условий контракта
4. **Готовность к изменениям** важнее следования первоначальному плану

Принципы Agile

1. Удовлетворение клиента за счёт ранней и бесперебойной поставки ценного программного обеспечения.
2. Приветствуются изменения требований даже в конце разработки.
3. Частая поставка рабочего программного обеспечения.
4. Тесное, ежедневное общение заказчика с разработчиками на протяжении всего проекта.
5. Проектом занимаются мотивированные личности, которые обеспечены нужными условиями работы, поддержкой и доверием.
6. Рекомендуемый метод передачи информации — личный разговор (лицом к лицу).

Принципы Agile

7. Работающее программное обеспечение — лучший измеритель прогресса.
8. Спонсоры, разработчики и пользователи должны иметь возможность поддерживать постоянный темп на неопределённый срок.
9. Постоянное внимание улучшению технического мастерства и удобному дизайну.
10. Простота — искусство не делать лишней работы.
11. Лучшие технические требования, дизайн и архитектура получаются у самоорганизованной команды.
12. Постоянная адаптация к изменяющимся обстоятельствам.

Принципы Scrum



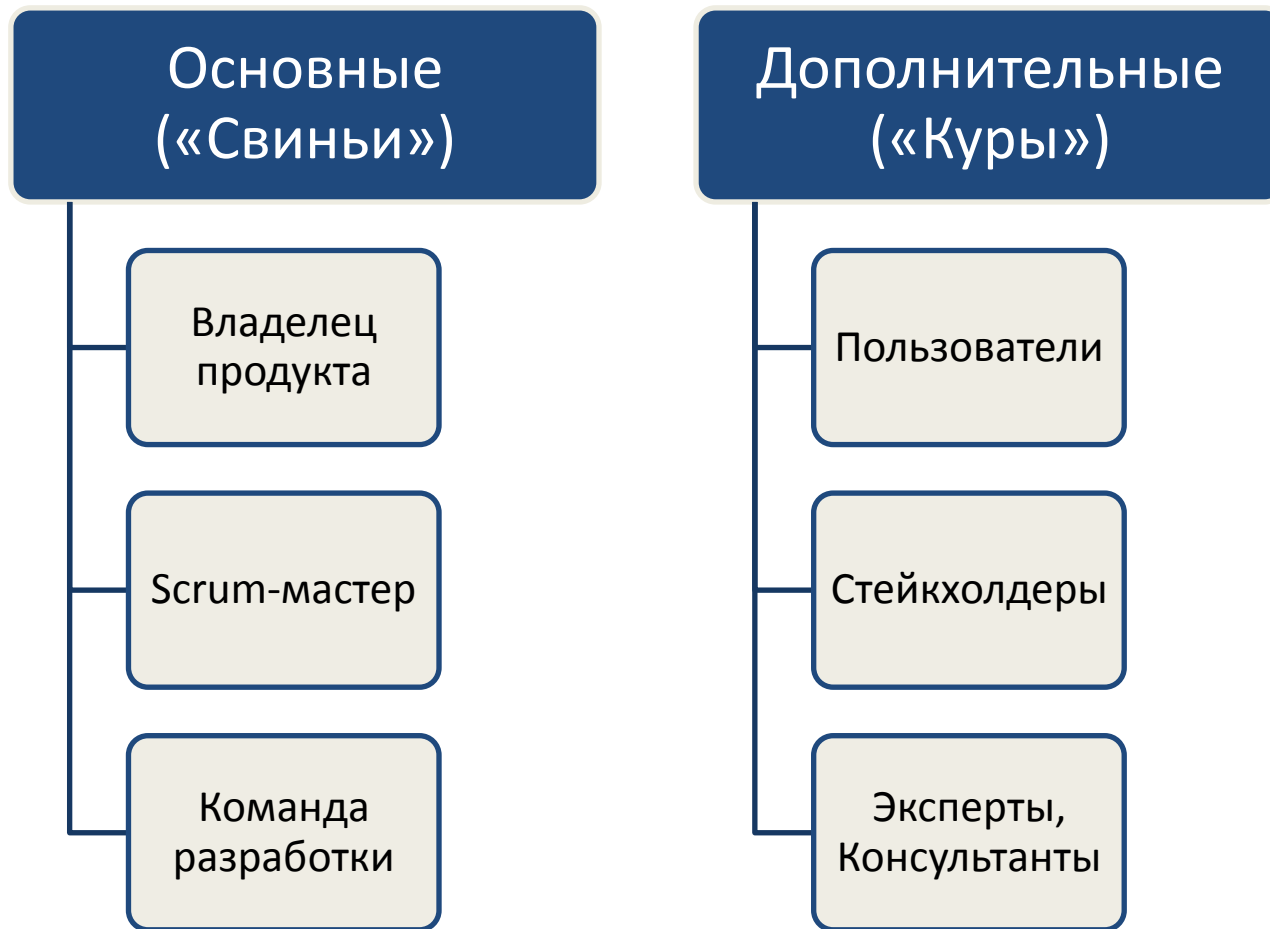
Терминология Scrum

Sprint (спринт) — базовая единица планирования процесса разработки (1 итерация), длительностью 2-4 недели.

Product Backlog (журнал пожеланий проекта) — это список требований к функциональности, упорядоченный по их степени важности, подлежащих реализации.

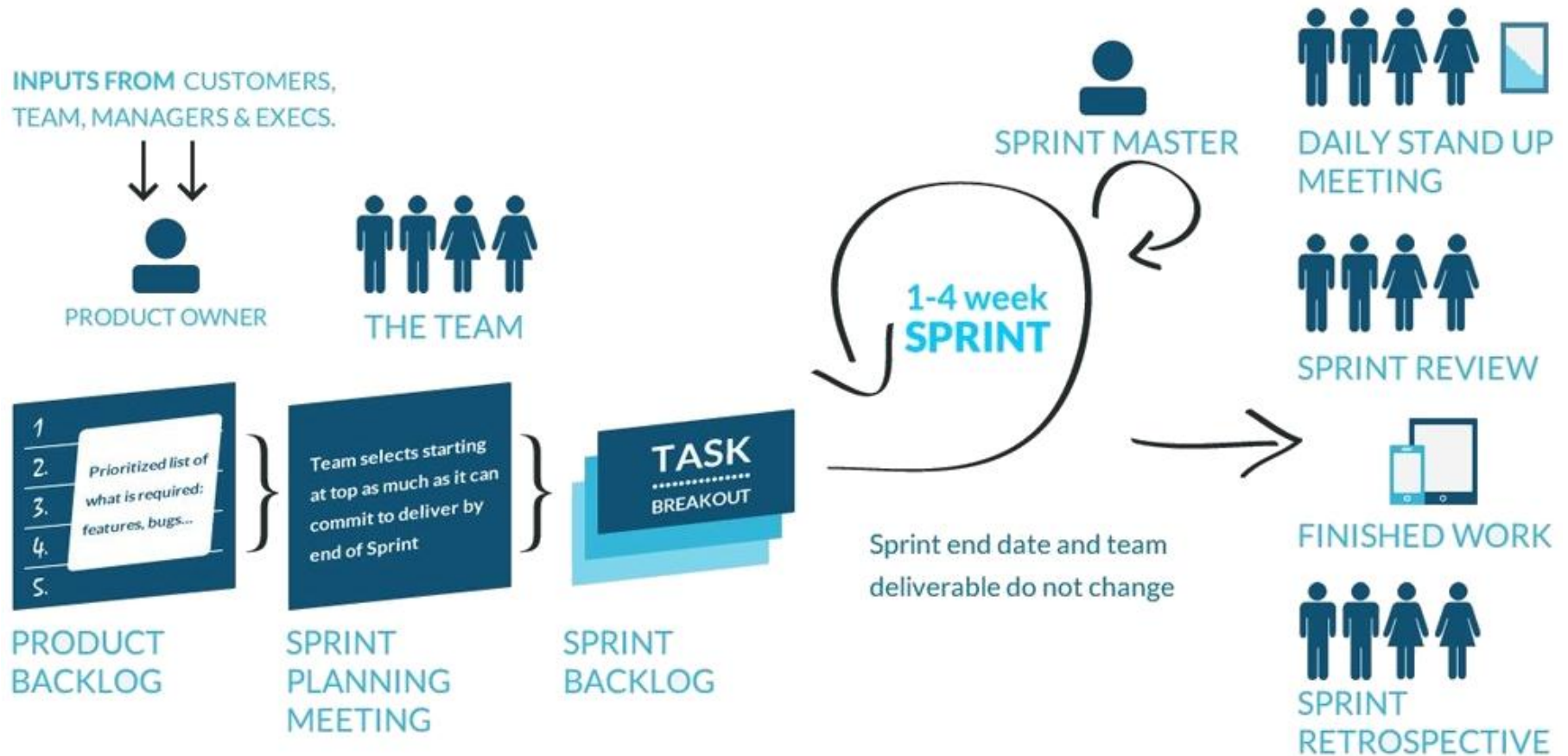
Sprint Backlog (журнал пожеланий спринта) — содержит функциональность, выбранную владельцем проекта из журнала пожеланий проекта.

Роли Scrum



Ken Schwaber. Agile Project Management with Scrum. — Microsoft Press, 2004.

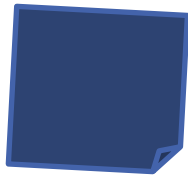
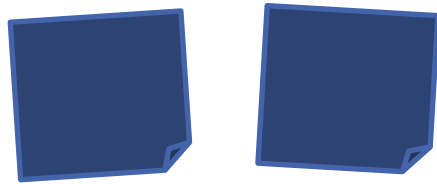
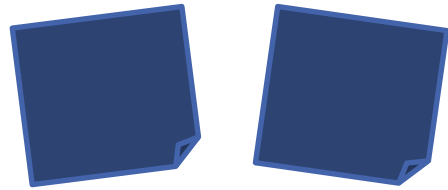
Scrum-процесс



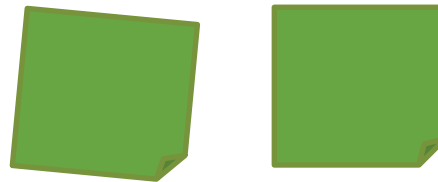
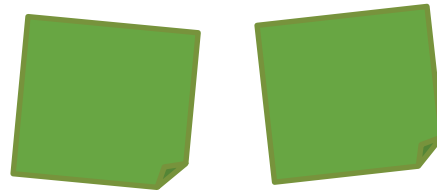
<http://www.augustinfotech.com/scrum-effective-way-implement-agile/>

Kanban

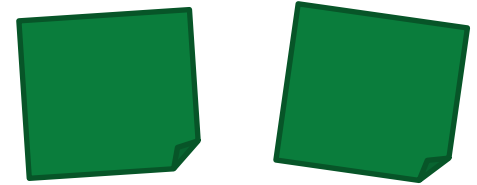
To Do



In Progress



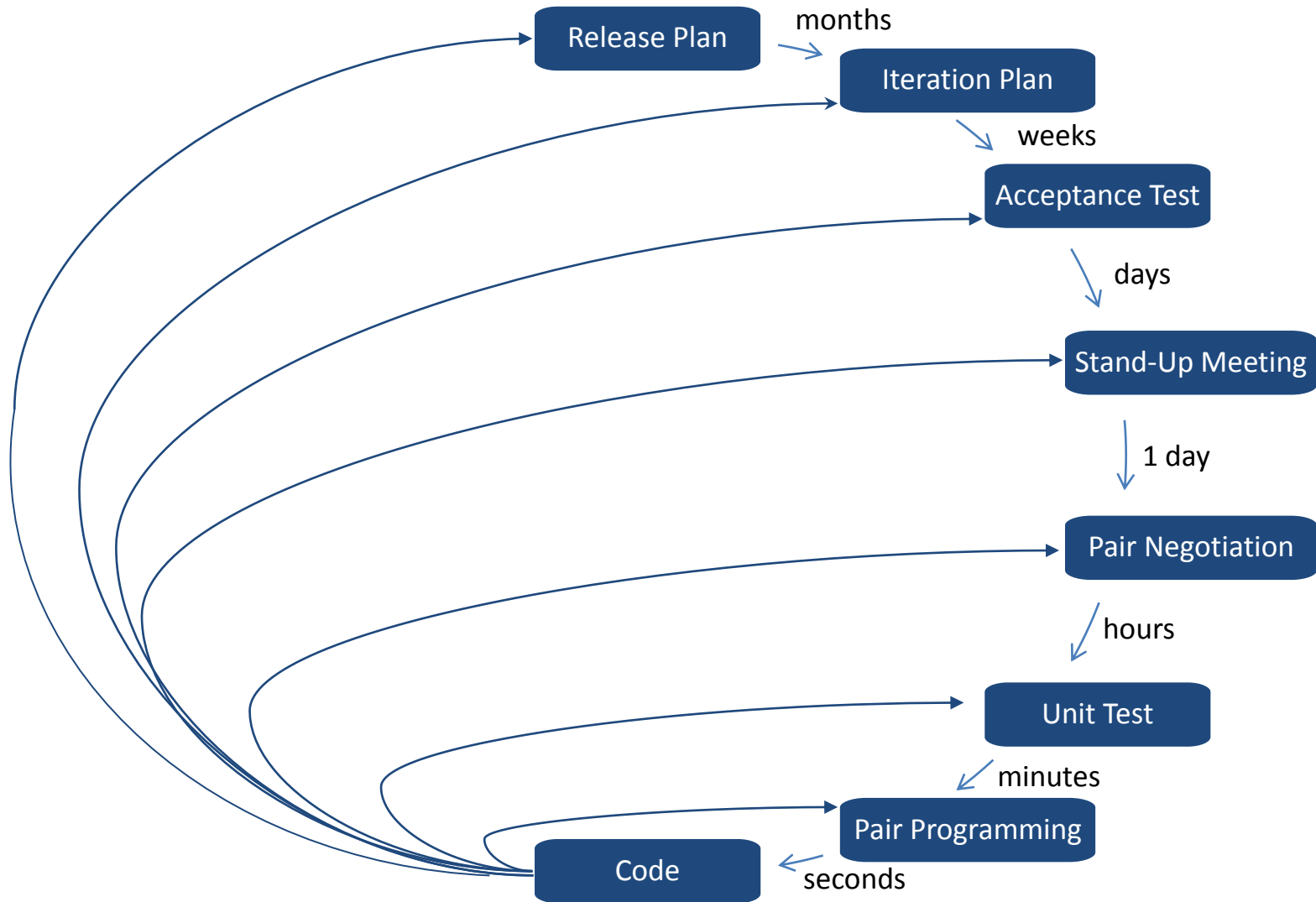
Done



Сравнение Kanban и Scrum

Признак	Kanban	Scrum
Роли	Нет четких заранее определенных ролей	Каждый член команды имеет четкую роль
Сроки и время выпуска	Непрерывный процесс выпуска, по требованиям бизнеса	Время выпуска определяется спринтом
Назначение задач	Члены команды выбирают задачи, количество задач в работе может быть ограничено	Члены команды выбирают задачи на время спринта
Модификации и изменения	Разрешены в течение всей жизни проекта	Разрешены только между спринтами, во время сильно нежелательны
Наиболее подходит	Проектам с неустоявшимися и более широкими приоритетами	Проектам с четким процессом создания продукта

Экстремальное программирование (XP)



Принципы XP

1. Клиенты (заказчики) определяют функции продукта
2. Частое формирование «легковесных релизов»
3. Использование единой системы именований и описаний
4. TDD
5. Написание простого кода, удовлетворяющего требованиям
6. Парное программирование
7. Коллективное владение кодом
8. Рефакторинг
9. Continuous Integration (ежедневные сборки)
10. Без переработок (недопустимы overnight)
11. Представитель клиента находится вместе с командой
12. Coding-style

Lean Startup

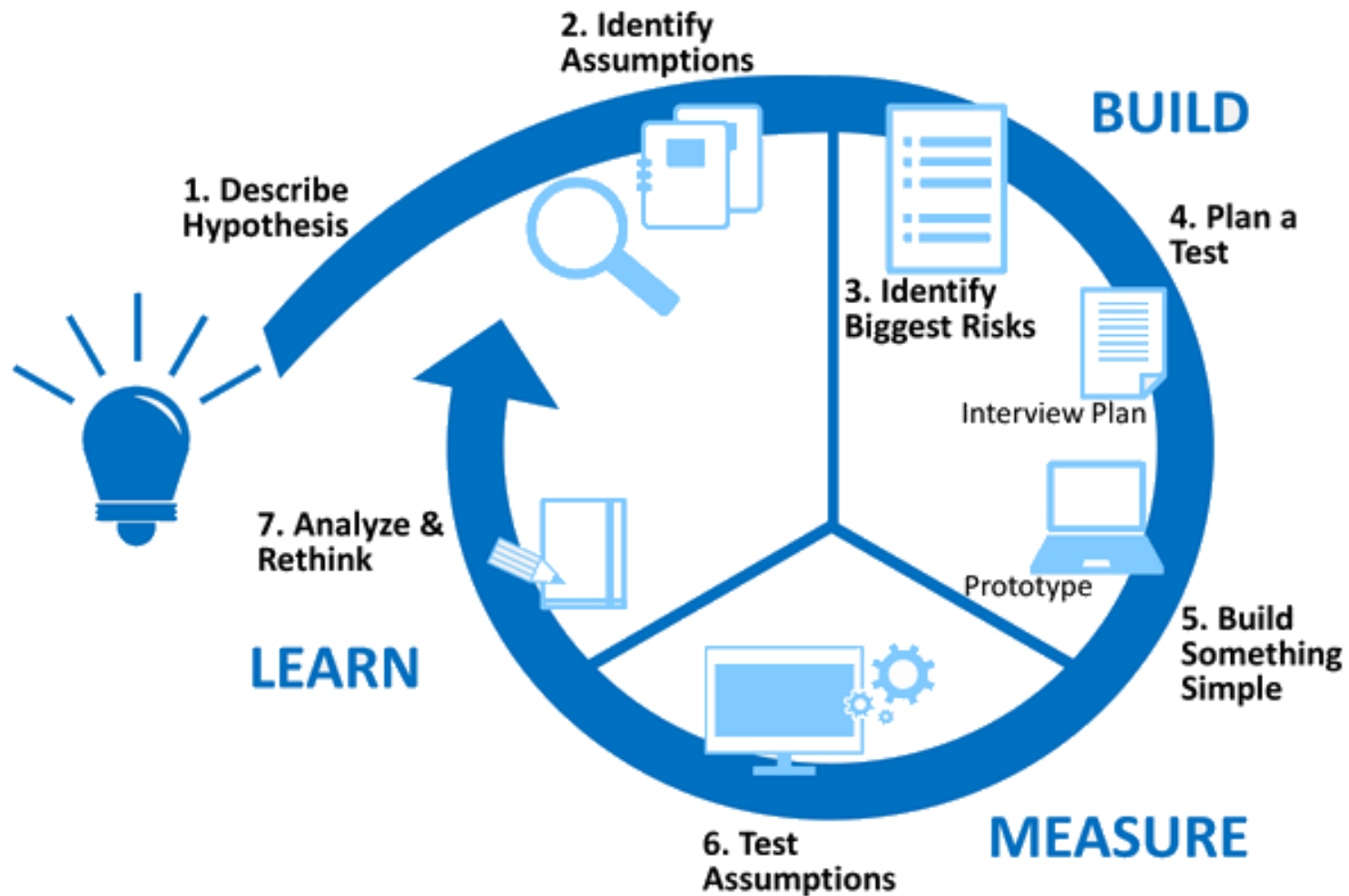
— концепция создания новых продуктов, основанная на научном подходе к управлению проектами: бережливом производстве, всеобщем управлении качеством, проведении экспериментов, итеративном выпуске продуктов для сокращения цикла разработки, измерении прогресса, и получение ценной обратной связи от клиентов. ^[1]

[1] Penenberg, Adam L. "Eric Ries is a Lean Startup machine". *Fast Company*.

Принципы Lean Startup

1. Обобщение гипотез в виде канвы бизнес-модели (схематическое описание того, как компания создает ценность для себя и потребителей)
2. Использование гибкой методологии разработки продукта.
3. Создание **Minimal Viable Product (MVP)** – простейшего продукта, для демонстрации его основной идеи, тестирования и получения быстрой обратной связи от клиентов.

Build-Measure-Learn



Выводы

1. Используйте то, что применимо конкретно к **вашему** проекту
2. Методологии **не решают инженерных задач**, они упрощают управление и контроль процесса разработки
3. Найдите **золотую середину** между планированием и гибкостью в вашем проекте

Литература

1. “Agile Project Management with Scrum, 2nd Edition”, Ken Schwaber
2. “The Scrum Guide”, Ken Schwaber and Jeff Sutherland
3. “The Mythical Man-Month: Essays on Software Engineering, 2nd Edition”, Frederick P. Brooks, Jr.
4. “Extreme Programming Explained: Embrace Change, 2nd Edition”, Kent Beck, Cynthia Andres
5. “The Lean Startup”, Eric Ries