



AGILE

это семейство гибких методологий создания проектов. Agile минимизирует риски посредством коротких циклов (итераций) разработки.

Используется во всех ТОП ИТ-компаниях Украины



Input from End-Users,
Customers, Team and
Other Stakeholders



Product Owner

1	
2	
3	
4	
5	
6	
7	FEATURES
8	
9	
10	
11	
12	



Team

Team Selects
How Much To
Commit To Do
By Sprint's End

Sprint Planning
Meeting
(Parts One and Two)

Product
Backlog
Refinement



Sprint
Backlog

ScrumMaster



Product
Backlog

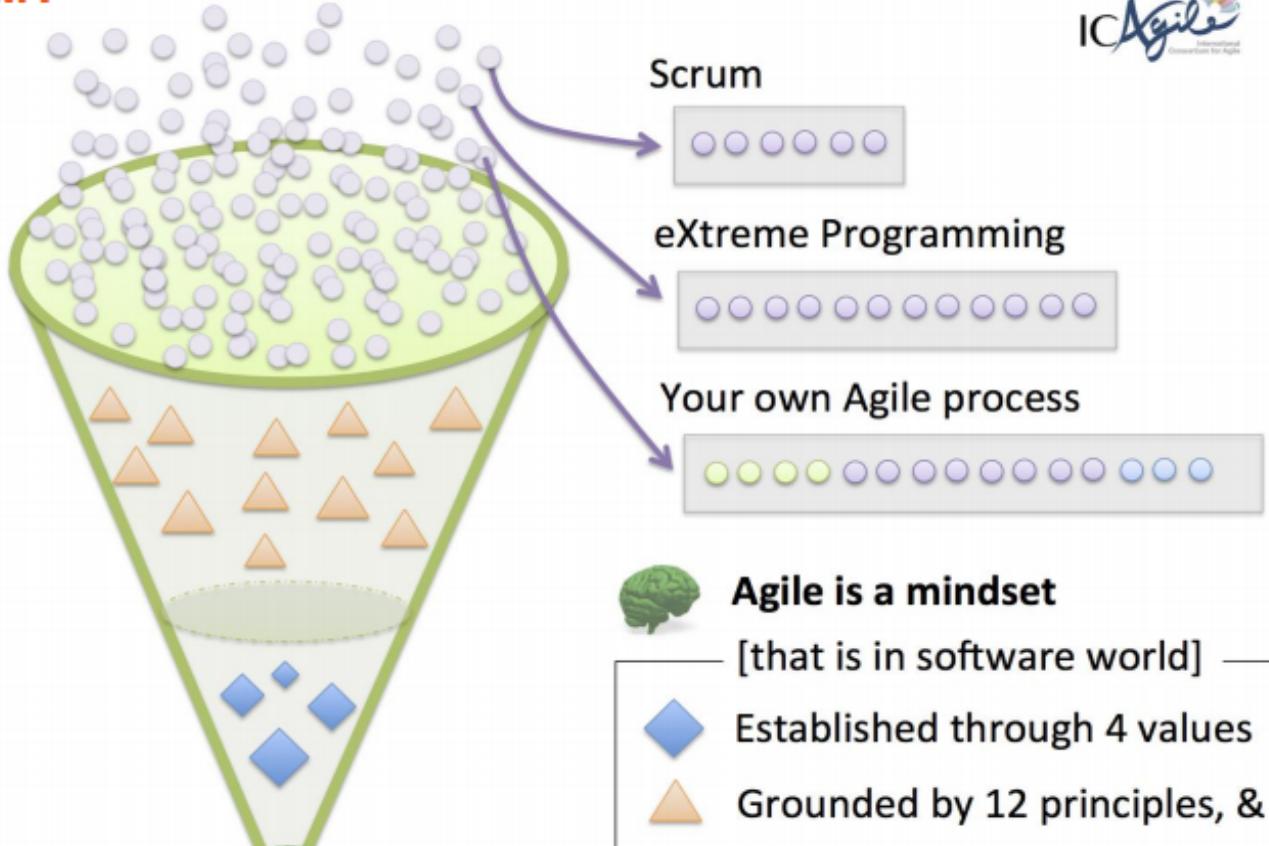
Что такое Agile и Скрам?

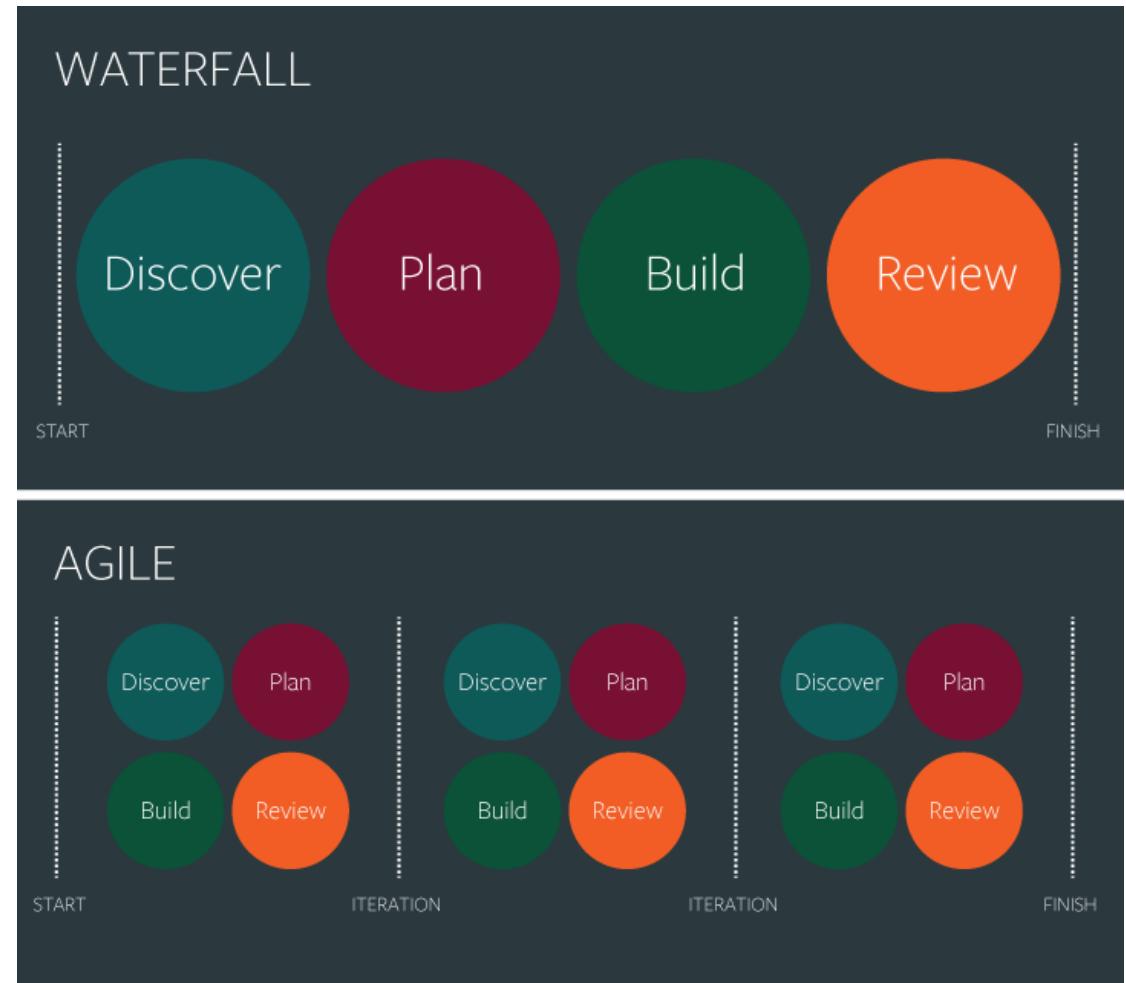
Agile практики

Agile манифест

Agile принципы

Agile ценности





Agile Manifesto



Люди и
взаимодействие > Процессов и
инструментов



Сотрудничество с
заказчиком > Контрактных
обязательств

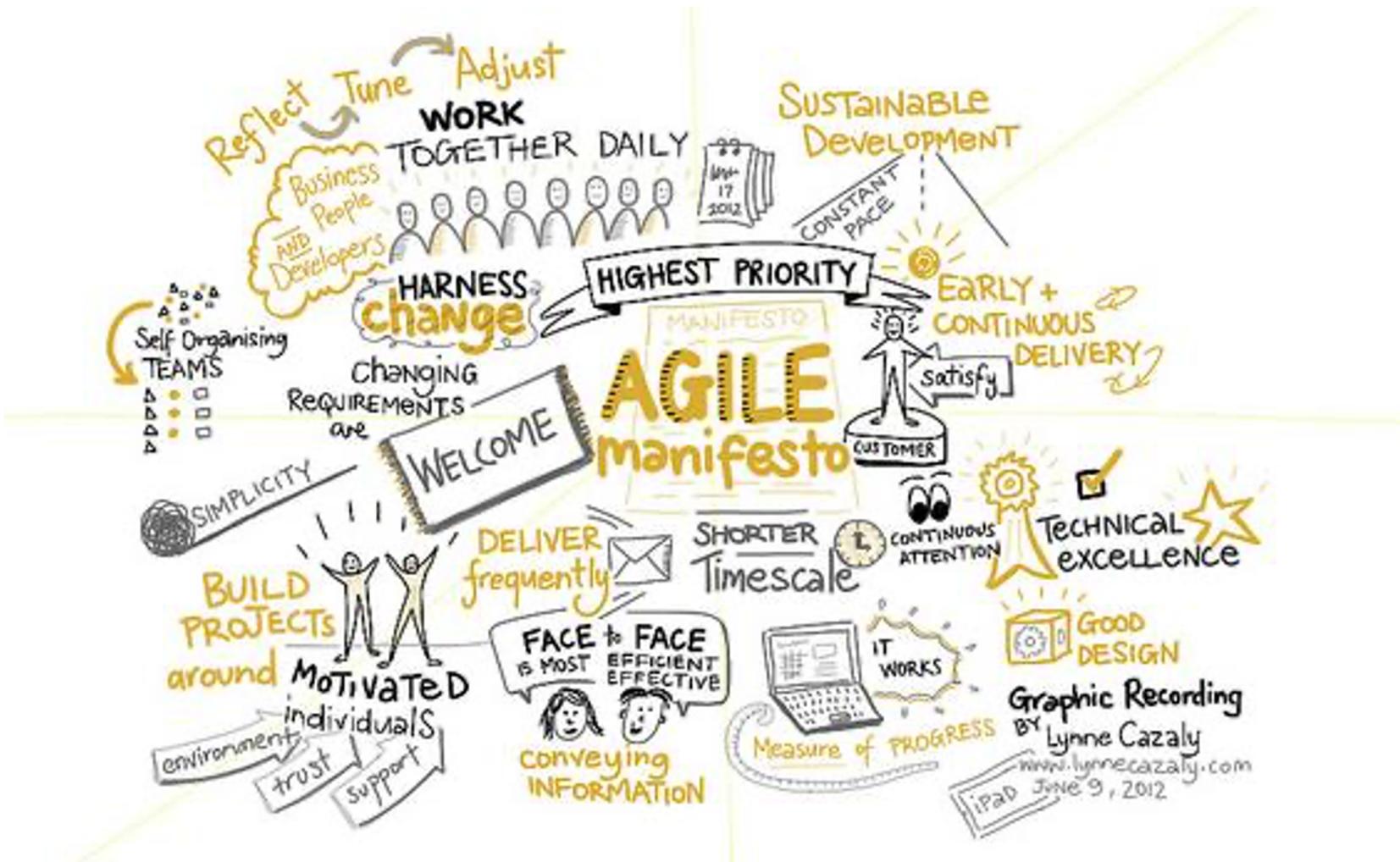


Работающий
продукт > Исчерпывающей
документации



Реагирование на
изменения > Следования плану

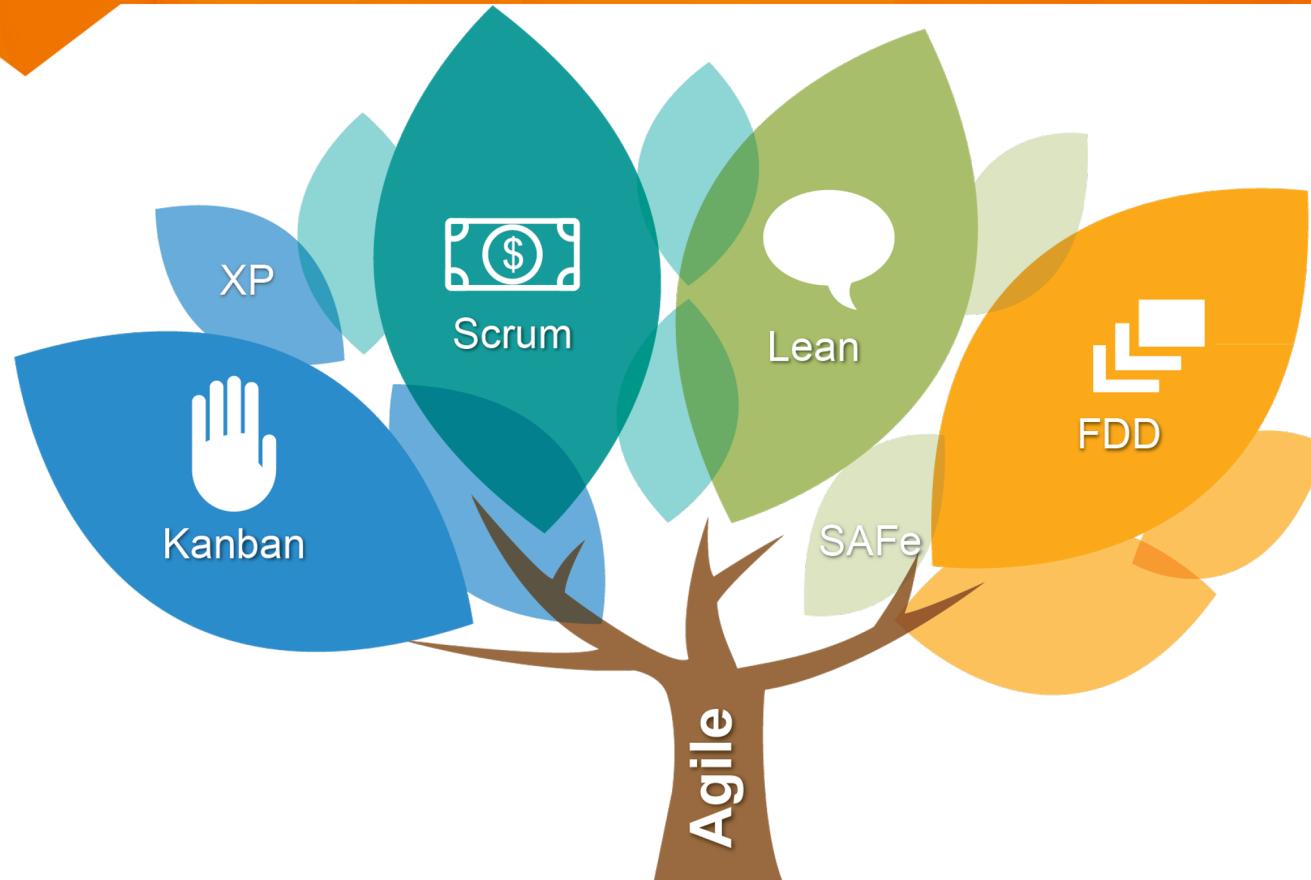




Agile principles

1. Наивысшим приоритетом для нас является удовлетворение потребностей заказчика, благодаря **регулярной и ранней поставке** ценного программного обеспечения.
2. **Изменение требований приветствуется**, даже на поздних стадиях разработки. Agile-процессы позволяют использовать изменения для обеспечения заказчику конкурентного преимущества.
3. Работающий продукт следует выпускать **как можно чаще**(от пары недель до пары месяцев).
4. На протяжении проекта **разработчики и представители бизнеса** должны ежедневно работать вместе.
5. Над проектом должны работать мотивированные профессионалы. Чтобы работа была сделана, создайте **условия**, обеспечьте **поддержку** и полностью **доверьтесь им**.
6. Непосредственное **общение** является наиболее практическим и эффективным способом обмена информацией как с самой командой, так и внутри команды.
7. Работающий **продукт** — основной показатель прогресса.
8. Инвесторы, разработчики и пользователи должны иметь возможность поддерживать **постоянный ритм** бесконечно. Agile помогает наладить такой устойчивый процесс разработки.
9. Постоянное внимание к **техническому совершенству** и **качеству проектирования** повышает гибкость
10. **Простота** — искусство минимизации лишней работы — крайне необходима.
- 11.Самые лучшие требования, архитектурные и технические решения рождаются у **самоорганизующихся команд**.
- 12.Команда должна систематически анализировать возможные **способы улучшения** эффективности и соответственно **корректировать** стиль своей работы.

AGILE



Input from End-Users,
Customers, Team and
Other Stakeholders



Product Owner

1	
2	
3	
4	
5	
6	
7	FEATURES
8	
9	
10	
11	
12	

Product
Backlog



Team

Team Selects
How Much To
Commit To Do
By Sprint's End

Sprint Planning
Meeting
(Parts One and Two)

Product
Backlog
Refinement



Sprint
Backlog

ScrumMaster



No Changes
in Duration or Goal



Daily Scrum
Meeting and
Artifacts Update



Review



Potentially
Shippable Product
Increment



Retrospective

Scrum-команда



Product Owner



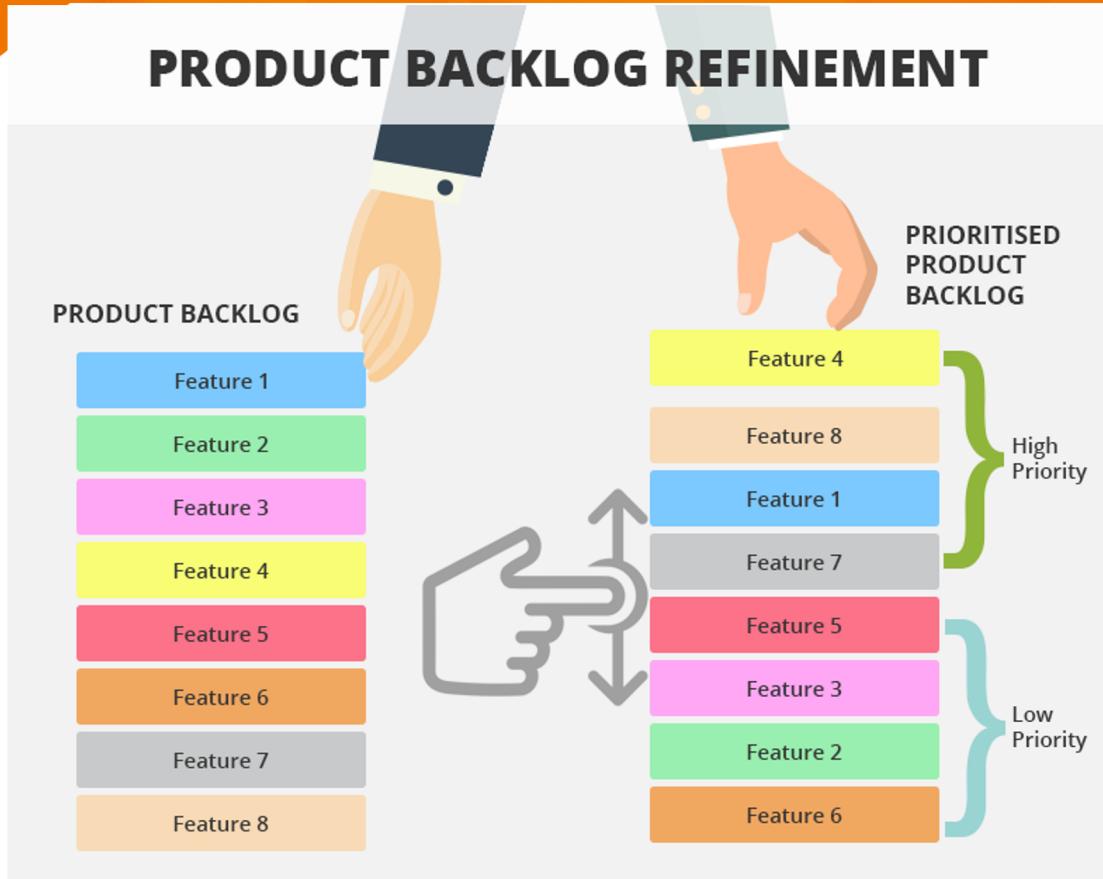
Scrum Master



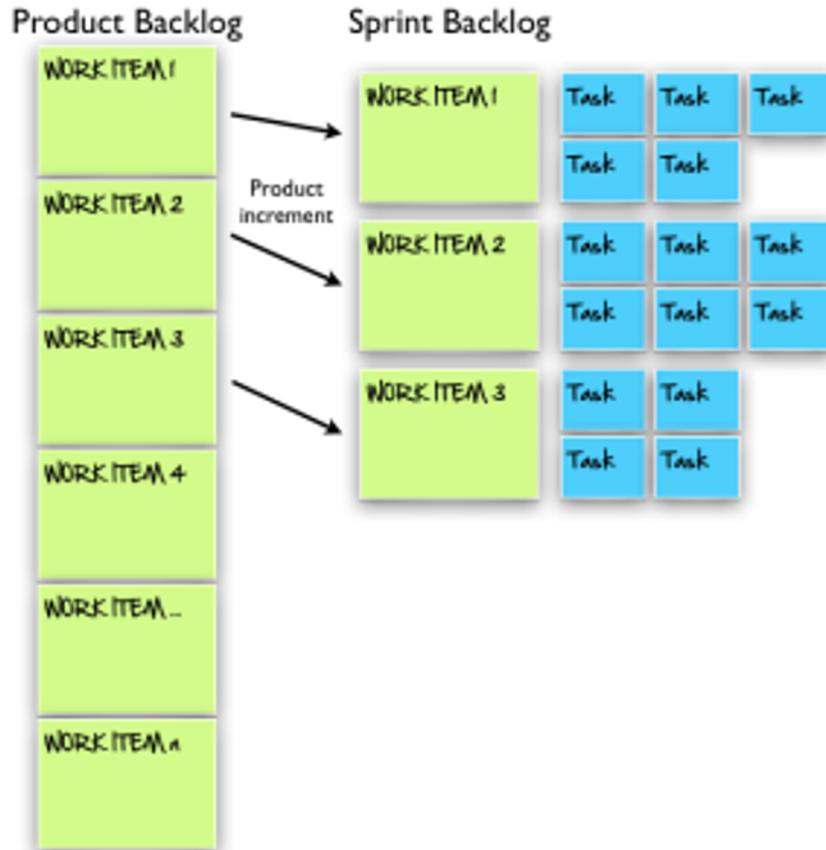
Development Team



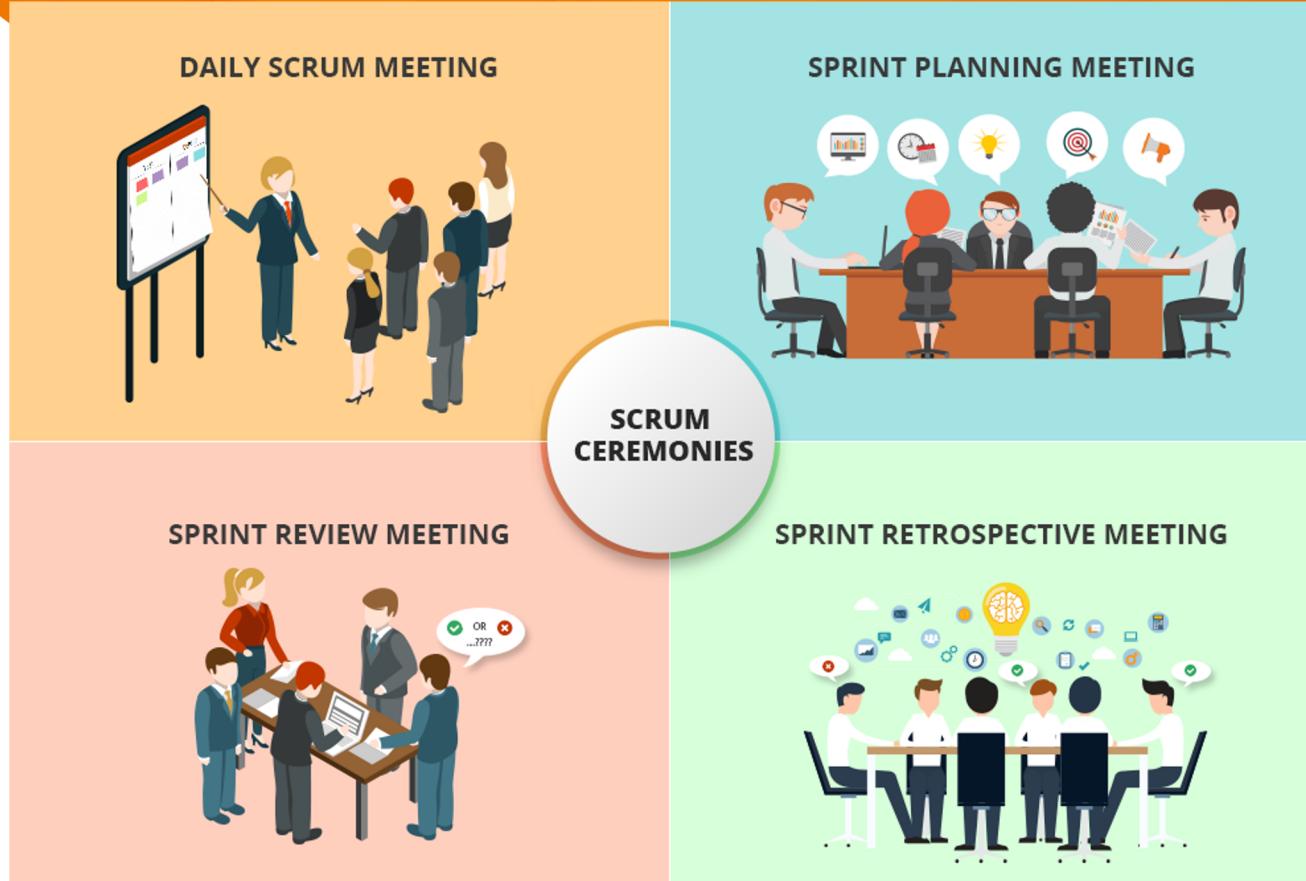
Product Backlog



Sprint Backlog



Scrum Ceremonies

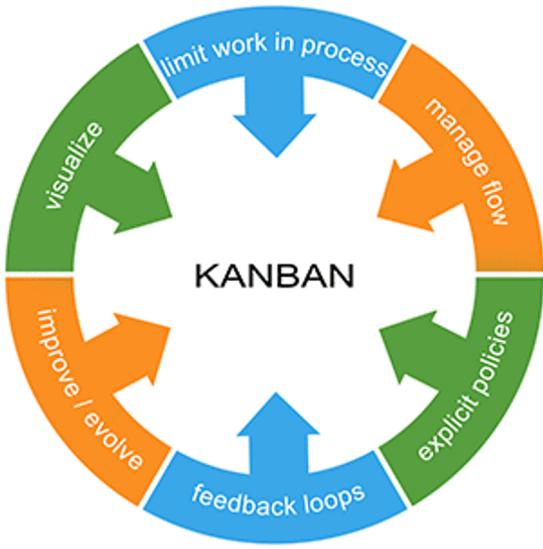


KANBAN

Канбан — метод управления разработкой, реализующий принцип “точно в срок” и способствующий равномерному распределению нагрузки между коллегами



KANBAN: three rules



1. Визуализируй производство
2. Ограничь WIP (work in progress)
3. Измеряй среднее время на выполнение одной задачи

Отличия KANBAN от Scrum

- В Kanban нет таймбоксов ни на что (ни на задачи, ни на спринты)
- В Kanban задачи больше и их меньше
- В Kanban оценки сроков на задачу опциональные или вообще их нет
- В Kanban «скорость работы команды» отсутствует и считается только среднее время на полную реализацию задачи

KANBAN

XP

Экстремальное программирование (Extreme Programming, XP)

- гибкая методология
- идея -> новый «экстремальный» уровень для традиционных методов и практик разработки ПО.



XP

1. Короткий цикл обратной связи (Fine-scale feedback)

- Test-driven development
- Игра в планирование (Planning game)
- Заказчик всегда рядом (Whole team, Onsite customer)
- Pair programming

2. Непрерывный процесс

- Continuous integration
- Refactoring
- Частые небольшие релизы (Small releases)

3. Понимание, разделяемое всеми

- Простота проектирования (Simple design)
- Метафора системы
- Коллективное владение кодом (Collective code ownership)
- Стиль кода (Coding standard or Coding conventions)

4. Социальная защищённость программиста (Programmer welfare)

- 40-часовая рабочая неделя (Sustainable pace, Forty-hour week)

Input from End-Users,
Customers, Team and
Other Stakeholders



Product Owner

1	
2	
3	
4	
5	
6	
7	FEATURES
8	
9	
10	
11	
12	



Team

Team Selects
How Much To
Commit To Do
By Sprint's End

Sprint Planning
Meeting
(Parts One and Two)

Product
Backlog
Refinement



Sprint
Backlog

ScrumMaster



No Changes
in Duration or Goal



Daily Scrum
Meeting and
Artifacts Update



Review



Potentially
Shippable Product
Increment



Retrospective

Product
Backlog