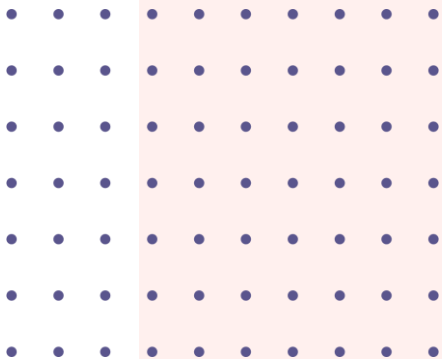


Урок 3. SDLC і STLC. Методології розробки ПЗ.

Структура заняття

1. Software Development Life Cycle (SDLC)
2. Software Testing Life Cycle (STLC)
3. Класичні методології розробки
4. Методології сімейства Agile





Як проходит розробка ПЗ?



Життєвий цикл розробки програмного забезпечення (Software Development Life Cycle)

— це структура, яка визначає кроки в розробці програмного забезпечення на кожному етапі й охоплює детальний план побудови, розгортання та обслуговування програмного забезпечення.

SDLC визначає повний цикл розробки, тобто всі завдання, пов'язані з плануванням, створенням, тестуванням та розгортанням програмного продукту.

Software Development Life Cycle (SDLC)





Які стадії тестування ви можете виділити?

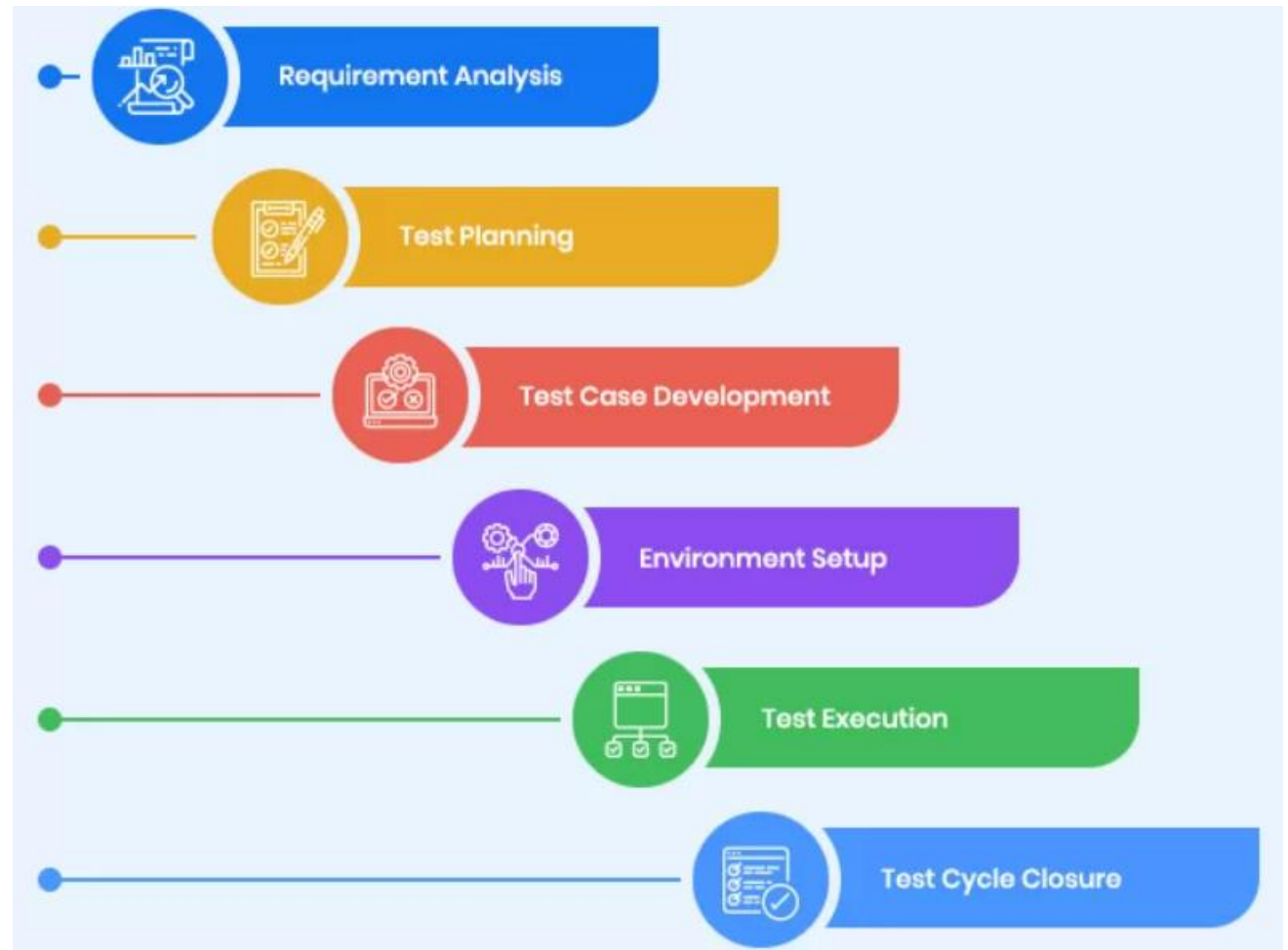


Життєвий цикл тестування програмного забезпечення (Software Testing Life Cycle)

— це послідовність конкретних дій, що проводяться під час процесу тестування для забезпечення досягнення цілей якості програмного забезпечення.

STLC включає як верифікацію, так і валідацію. Він складається з низки заходів, які проводяться методично, щоб допомогти сертифікувати програмний продукт

Фазы STLC



1. Аналіз вимог

1. Вивчення вимог до програмного забезпечення
2. Комунікація з бізнес аналітиком, архітектором та дизайн командою для уточнення та повного розуміння вимог
3. Визначення, чи можливо буде задокументовані вимоги протестувати



2. Планування тестування

1. Підготовка тестового плану та стратегії тестування
2. Вибір інструментів для проведення тестування
3. Планування ресурсів, ролей та відповідальності



3. Дизайн тестів

1. Вибір технік тест-дизайну
2. Написання тестових сценаріїв та скриптів автоматизації
(якщо це передбачено стратегією тестування)
3. Приоритезація тестових сценаріїв
4. Створення тестових даних



4. Налаштування тестового середовища

1. Визначення вимог до тестового середовища
2. Визначення кількості тестових середовищ
3. Безпосереднє налаштування тестових середовищ
4. Проведення валідації тестових середовищ



5. Виконання тестування

1. Запуск підготовлених тестових сценаріїв
2. Документування знайдених помилок
3. Повторне тестування після виправлення помилок
4. Слідування життєвим циклом дефектів та задач



6. Звітність і закриття тестування

1. Підготовка звіту про виконання тестування
2. Аналіз етапів тестування
3. Проведення ретроспективи виконаного тестування





Моніторинг та контроль тестування (Test monitoring and control)


— це процес спостереження за всіма показниками, потрібний для гарантії, що проєкт працює добре, за графіком і не виходить за рамки бюджету.

Процес збору, реєстрації та надання інформації про діяльність проєкту, яку необхідно знати менеджеру проєкту та замовнику.

Передбачає виконання дій, необхідних для досягнення цілей плану тестування, на основі даних моніторингу. Моніторинг та контроль відбуваються під час усіх етапів тестування



Чи є різниця між SDLC та STLC?

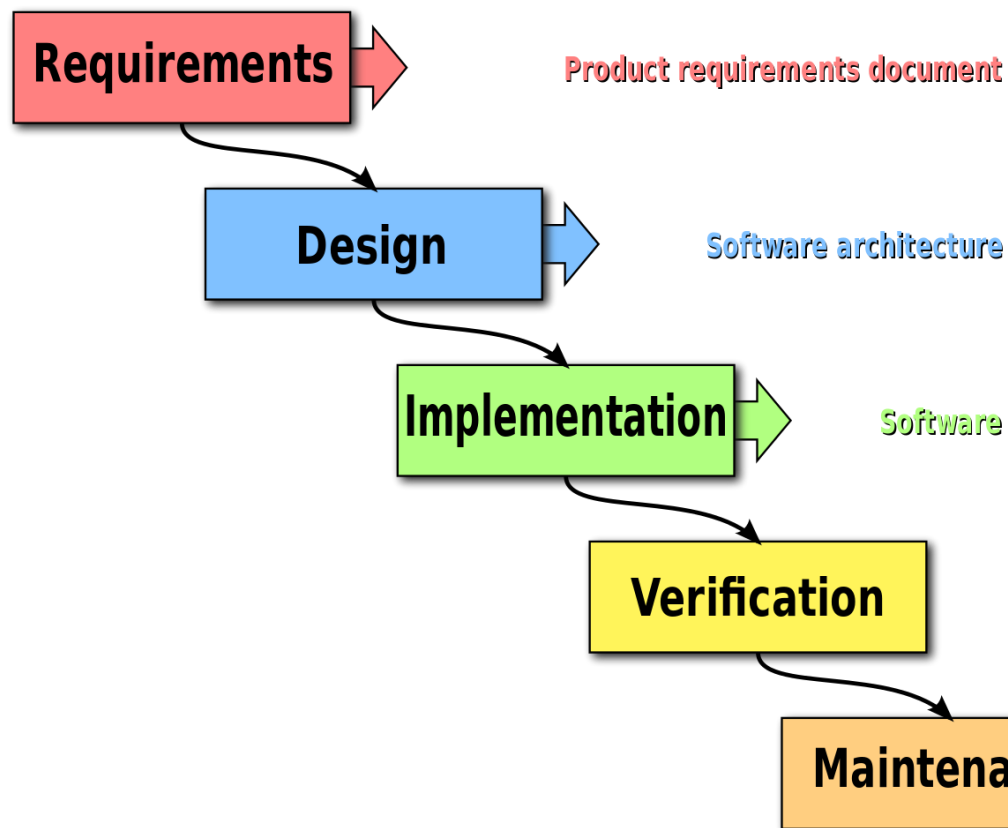


Методологія розробки – це система принципів, а також сукупність ідей, понять, методів, способів, засобів та практик, що визначають стиль розробки програмного забезпечення

- Waterfall
- V-model
- Spiral model
- Iterative model

Waterfall (каскадна, водоспадна)

одна з найстаріших методологій, що передбачає суворе послідовне виконання всіх етапів, кожен з яких повинен завершитися перед початком наступного.



Waterfall модель застосовують в авіабудуванні, військовій або космічній галузях, медицині та фінансовому секторі. Там Waterfall саме місце, тому що цим сферам потрібні чітко побудовані процеси та терміни, а це суть каскаду. Звідси й порівняння з водоспадом: кожен етап створення продукту, немов потік води, продовжує попередній і не може початися, доки минулий не завершився.

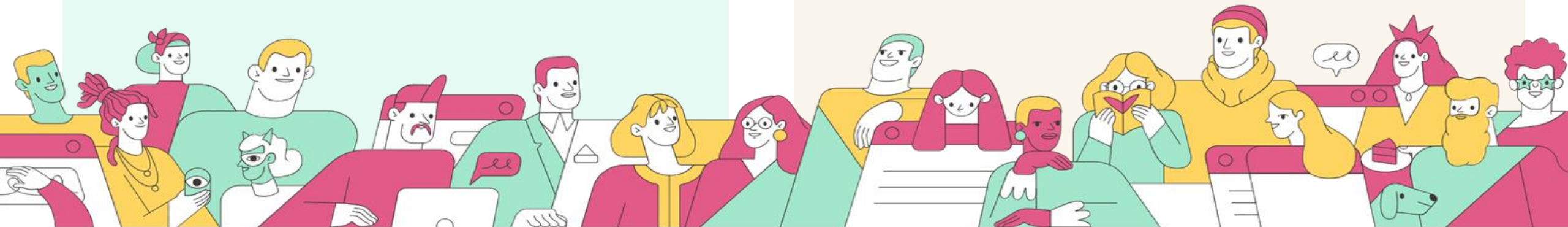
Waterfall модель

Переваги

- Висока прозорість розробки та фаз проекту
- Точна послідовність
- Стабільність вимог
- Суворий контроль за менеджментом проекту
- Полегшує роботу зі складання плану проекту та збору команди проекту
- Добре визначає процедуру контролю якості.

Недоліки

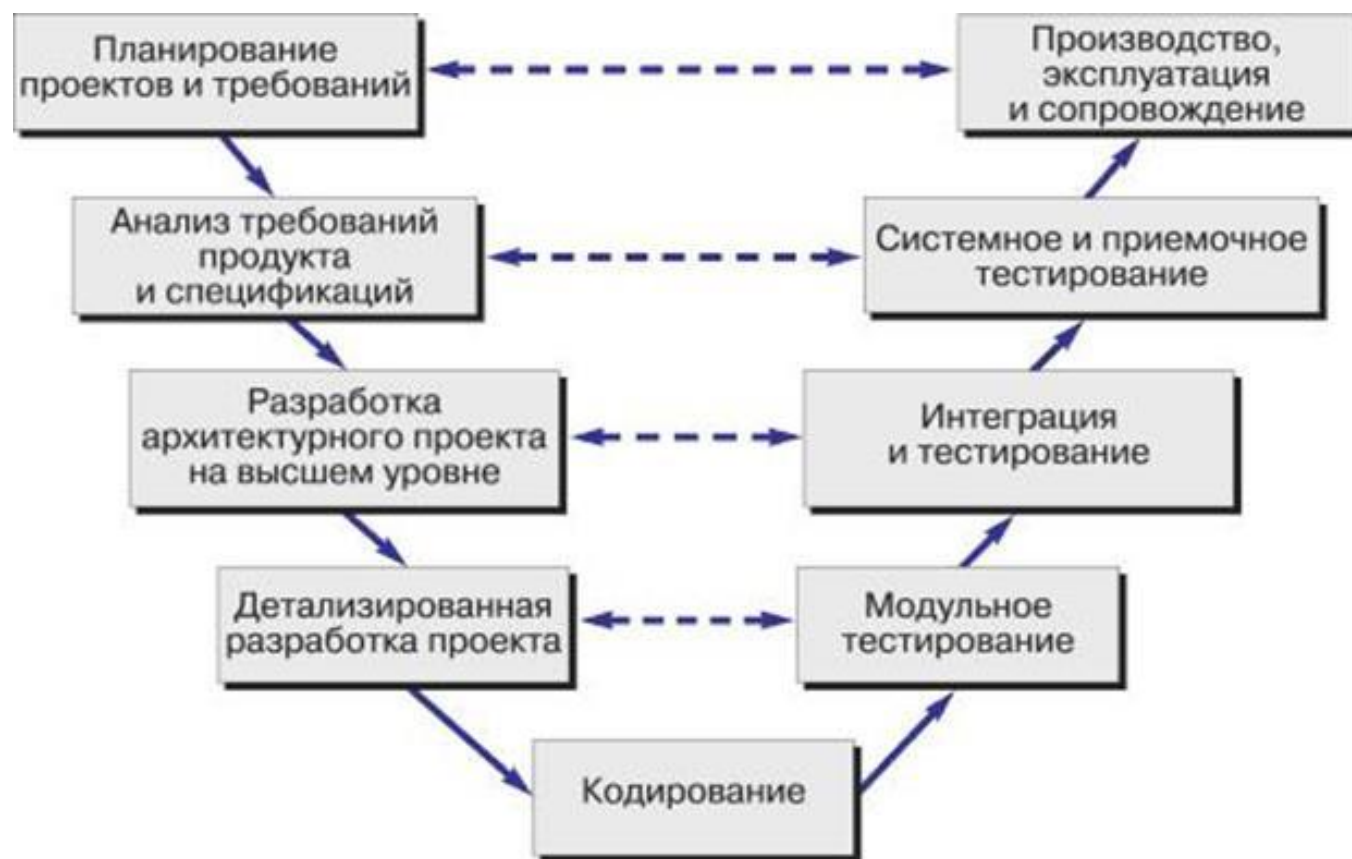
- Необхідність затвердження повного обсягу вимог до системи на першому етапі
- У разі потреби внесення змін вимог пізніше – повернення до першої стадії та переробка заново всієї виконаної роботи
- Збільшення витрат коштів та часу у разі потреби зміни вимог
- У замовника немає можливості ознайомитися із системою заздалегідь
- Користувач не має змоги звикати до продукту поступово.



V-модель (V-model)

модель, що описує процеси життєвого циклу розробки ПЗ з моменту складання специфікації вимог до етапу супроводу.

V модель показує інтеграцію процесів тестування в кожен фазу циклу розробки ПЗ.



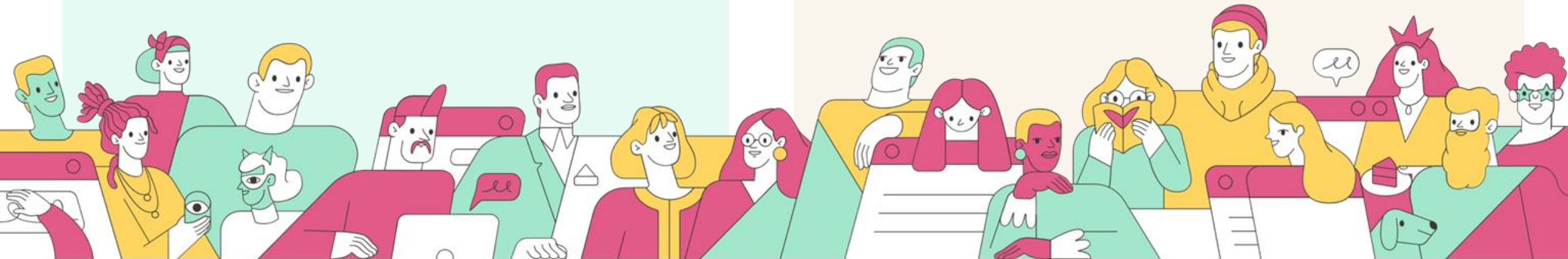
V модель

Переваги

- планування на ранніх стадіях розробки системи тестування
- забезпечення атестації та верифікації всіх проміжних результатів розробки
- спрощення (порівняно з каскадною моделлю) відстеження ходу процесу розробки, можливість більш реального використання графіка проекту
- простота у використанні.

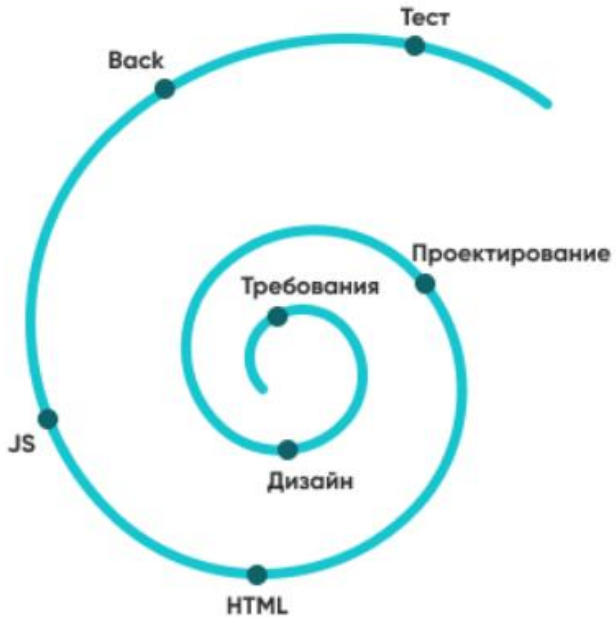
Недоліки

- Складність підтримки паралельних подій
- Непередбаченість ітерацій між фазами
- Неможливість внесення динамічних змін до вимог на різних етапах життєвого циклу
- Пізні терміни тестування вимог у життєвому циклі, що істотно впливає на графік виконання проекту при необхідності виконати їх зміни
- Відсутність у моделі дій, створені задля аналіз ризиків.

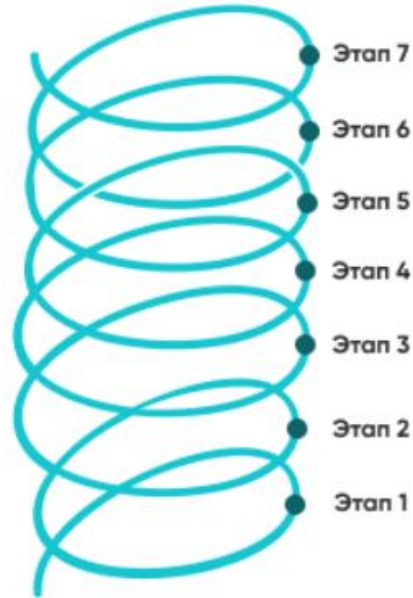


Спиральна модель

модель розробки ПЗ, що поєднує в собі ідеї ітеративної і каскадної моделей. У спіральній моделі життєвий шлях продукту зображується у вигляді спіралі, яка починається з етапу планування і розкручується з проходженням кожного наступного кроку. Модель заснована на прототипі: на виході з чергового витка потрібно отримати готовий протестований прототип, який доповнює існуючий білд. Прототип, відповідний до всіх вимог — готовий до релізу. Головна особливість спіральної моделі — концентрація на можливих ризиках.



Как работает изнутри



Как видит клиент

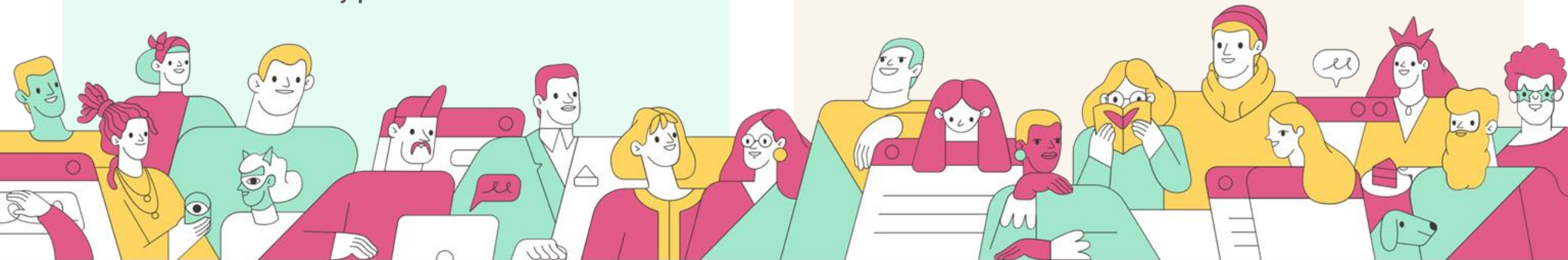
Спіральна модель

Переваги

- Моніторинг ризиків полегшує управління великим, складним і дорогим проектом
- Діюча версія продукту на ранніх стадіях життєвого циклу ПЗ
- Зміни можна внести на пізніх стадіях розробки
- Проект може бути поділений на кілька частин і найризикованіші можуть бути реалізовані на ранніх стадіях.
- Суворий контроль за документацією, як результат постійного аналізу ризиків

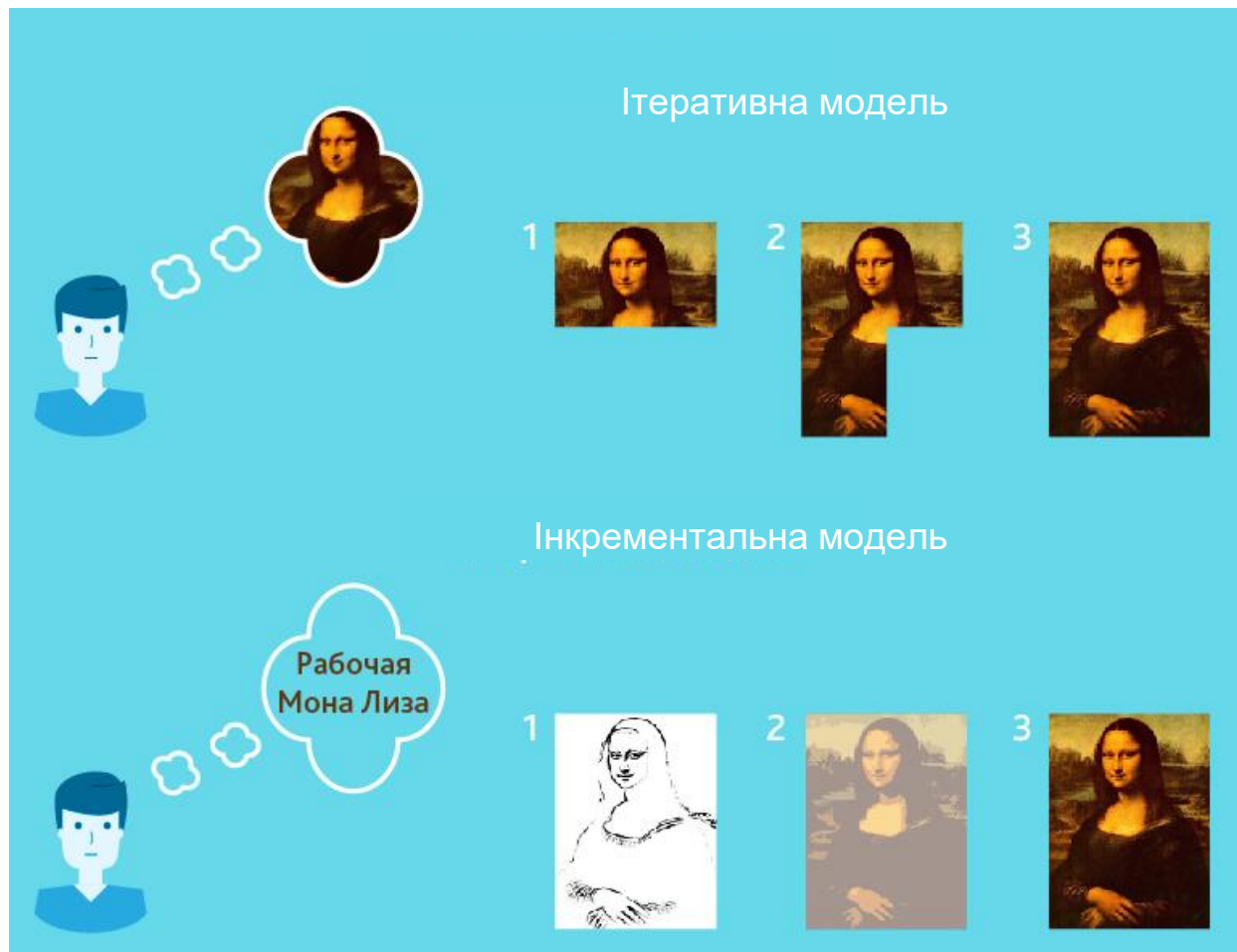
Недоліки

- Моніторинг ризиків вимагає додаткових ресурсів, отже, ця модель може бути дуже затратною
- Велика кількість проміжних стадій розробки. Як наслідок – великий обсяг документації
- На ранніх стадіях дата завершення роботи над проектом може бути невідома, що також ускладнює контроль над процесом розробки



Ітеративні або інкрементальні моделі

моделі, які передбачають розбиття створюваної системи на складові, які розробляються за допомогою декількох послідовних проходів всіх або частини робіт



Який підхід кращий?



1



2



3



4



1



2



3



4



5

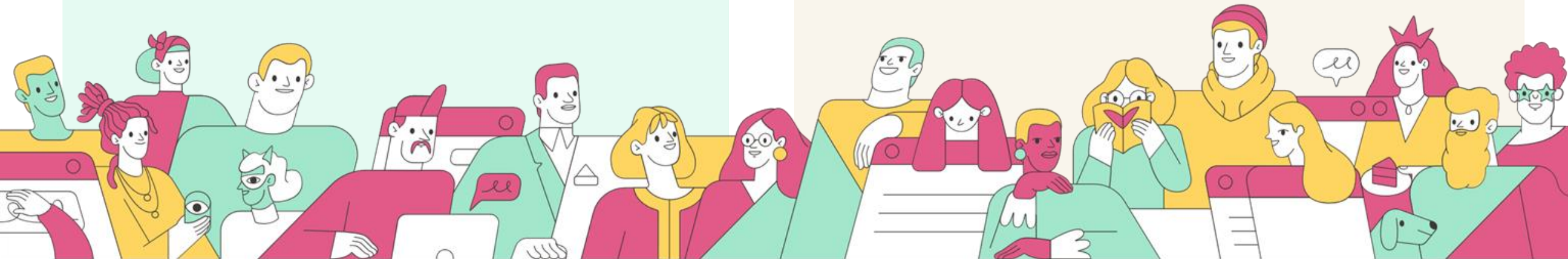
Інкрементальні моделі

Переваги

- Зменшення витрат на зміни вимог користувачів
- Значно скорочується , повторний аналіз та сукупність документації порівняно з водопадною (каскадною) моделлю.
- Швидке отримання відгуку від клієнта про виконану роботу
- У клієнта є можливість швидко отримати та освоїти створене програмне

Недоліки

- Прогрес не дуже добре простежується – менеджери повинні постійно вимірювати прогрес процесу
- Структура системи має тенденцію до погіршення при додаванні нових компонентів (частин) – постійні зміни порушують структуру системи. Щоб уникнути цього та підвищити якість програмного забезпечення потрібен додатковий час та гроші на рефакторинг. Погана структура робить програмне забезпечення складним та дорогим для подальших змін.



**На основі ітеративної моделі було
створено Agile — не модель і не
методологію, а скоріше підхід до
розробки.**

Agile

маніфест цінностей і принципів підходу до управління проєктами, який вирішує проблеми традиційного проєктного менеджменту. Ця методологія створена як протилежність традиційній лінійній методології «водоспаду», передбачаючи ітеративну і покрокову розробку ПЗ для мінімізації ризиків



Можна охарактеризувати як загальний підхід до розробки але в свою чергу він включає в себе різні фреймворки які більш детально описують підхід до розробки.

Agile Manifesto

**ЛЮДИ
И ОБЩЕНИЕ**

важнее

**ПРОЦЕССЫ
И ИНСТРУМЕНТЫ**

**РАБОТАЮЩИЙ
ПРОДУКТ**

важнее

**ИСЧЕРПЫВАЮЩАЯ
ДОКУМЕНТАЦИЯ**

**СОТРУДНИЧЕСТВО
С ЗАКАЗЧИКОМ**

важнее

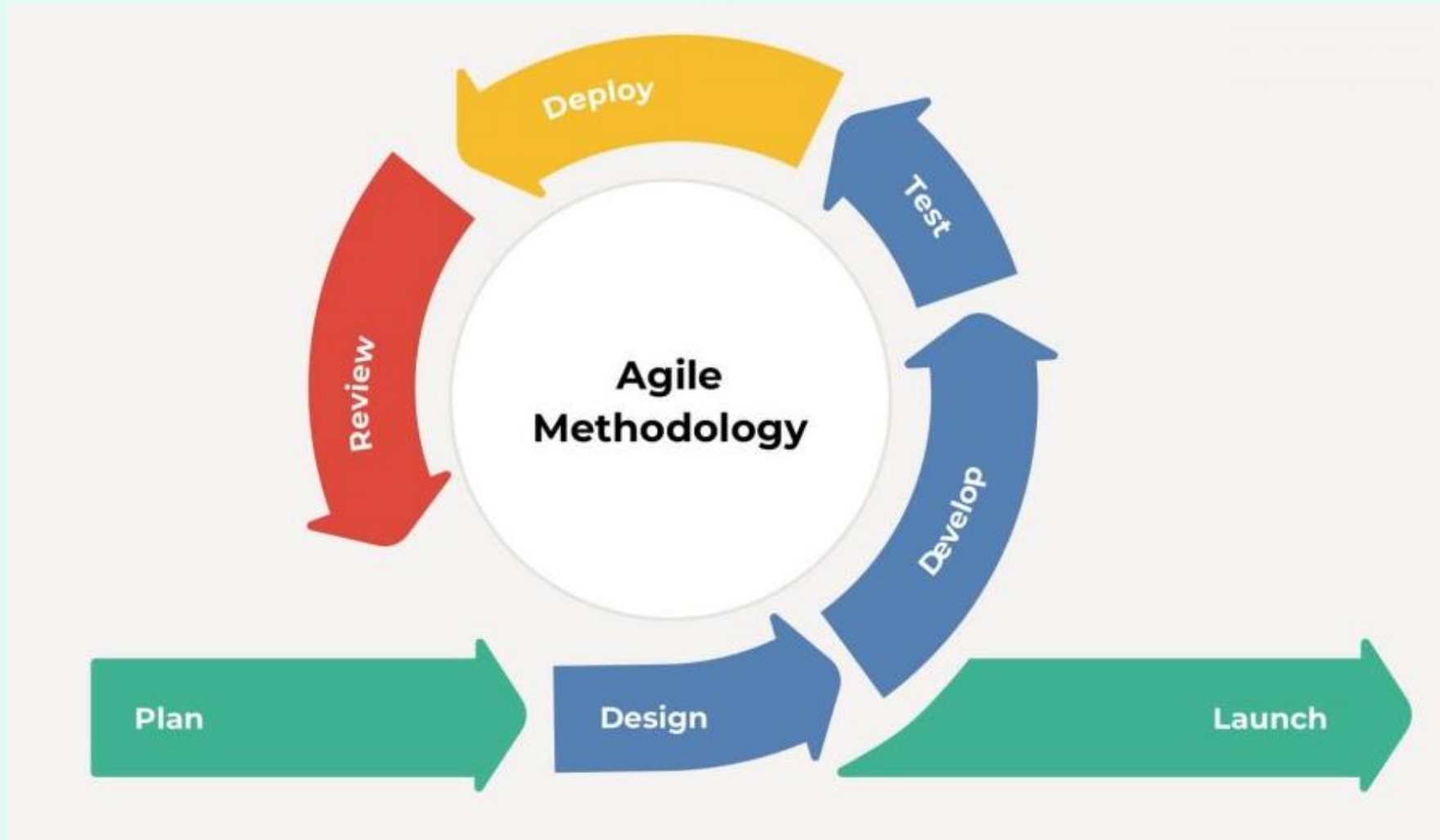
**СОСТАВЛЕНИЕ
КОНТРАКТОВ**

**ГОТОВНОСТЬ
К ИЗМЕНЕНИЯМ**

важнее

**СЛЕДОВАНИЕ
ПЛАНУ**

Ітеративна розробка





Які фреймворки Agile ви знаєте?

**Extreme
Programming
(XP)**

Kanban

Scrum

**FDD
(Feature
Driven
Development)**

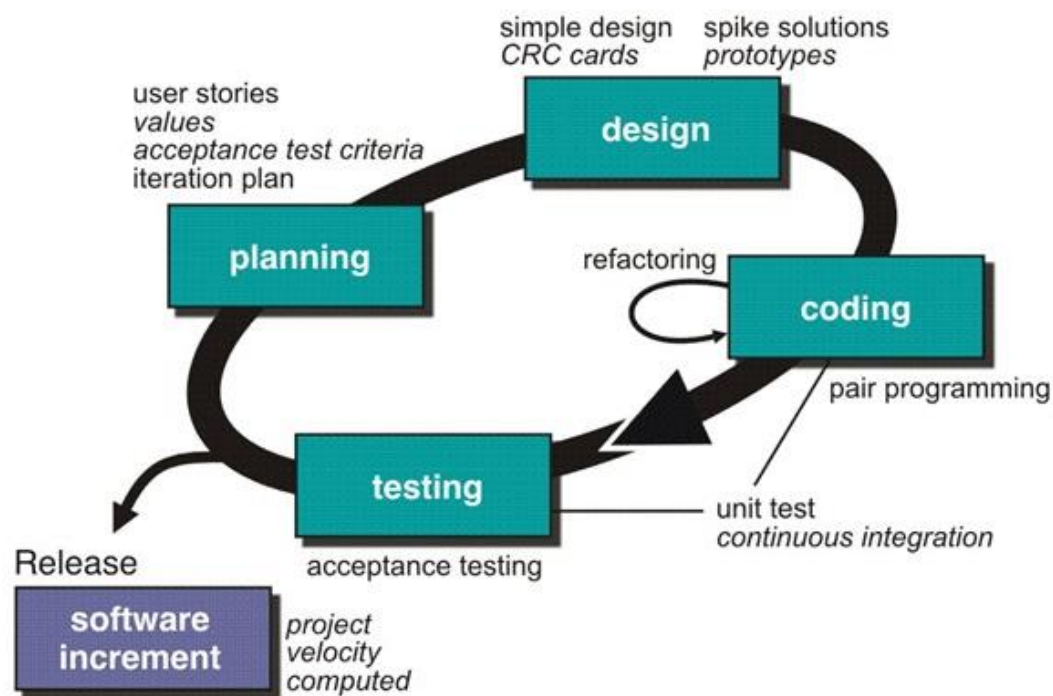
Crystal



Extreme Programming (XP)

спрощена методологія організації розробки програм для невеликих і середніх за розміром команд розробників в умовах незрозумілих або мінливих вимог.

Основними цілями XP є підвищення довіри замовника до програмного продукту за допомогою реальних доказів успішності розвитку розробки, а також різке скорочення термінів розробки продукту. Основними принципами є ітеративний зворотний зв'язок із замовником, інтенсивна розробка малими групами і простота рішень



Extreme Programming (XP)

Переваги

- Замовник отримує саме той продукт, який йому потрібен
- Команда швидко вносить зміни до коду та додає нову функціональність за рахунок простого дизайну коду, частого планування та релізів
- Код завжди працює за рахунок постійного тестування та безперервної інтеграції
- Висока якість коду адже він написаний за єдиним стандартом і постійно рефакториться
- швидкий темп розробки за рахунок парного програмування, відсутності переробок, присутності замовника у команді
- Витрати розробку нижче, адже команда орієнтована на код, а не на документацію і зустрічі

Недоліки

- Успіх проекту залежить від залучення замовника, якого не так просто досягти
- Важко передбачити витрати часу на проект оскільки на початку ніхто не знає повного списку вимог
- успіх XP сильно залежить від рівня програмістів, методологія працює тільки з senior спеціалістами
- менеджмент негативно ставиться до парного програмування
- регулярні зустрічі з програмістами дорого коштують замовникам
- вимагає надто сильних культурних змін, щоб не контролювати кожне завдання
- через брак структури та документації не підходить для великих проектів

Kanban

метод управління розробкою, який реалізує принцип «точно в строки» і сприяє рівномірному розподілу навантаження між працівниками. За цього підходу весь процес розробки прозорий для всіх членів команди. Мета – отримувати готовий якісний продукт вчасно. Для візуалізації процесів використовується Kanban-дошка.



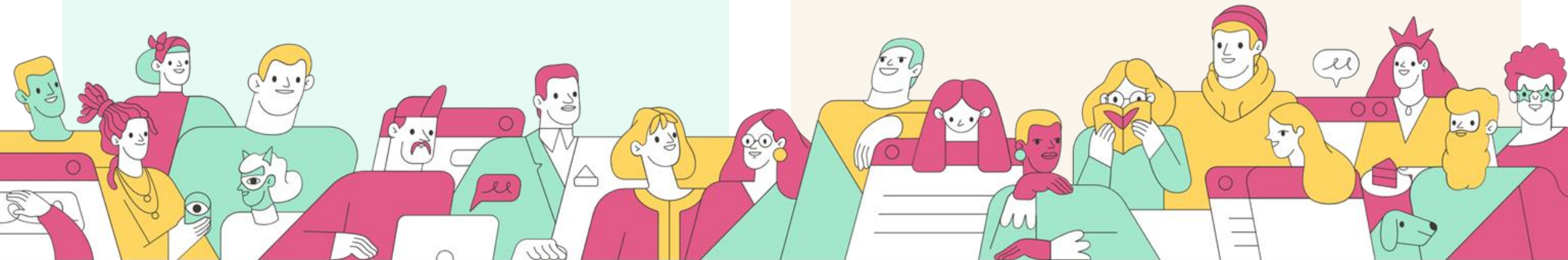
Kanban

Переваги

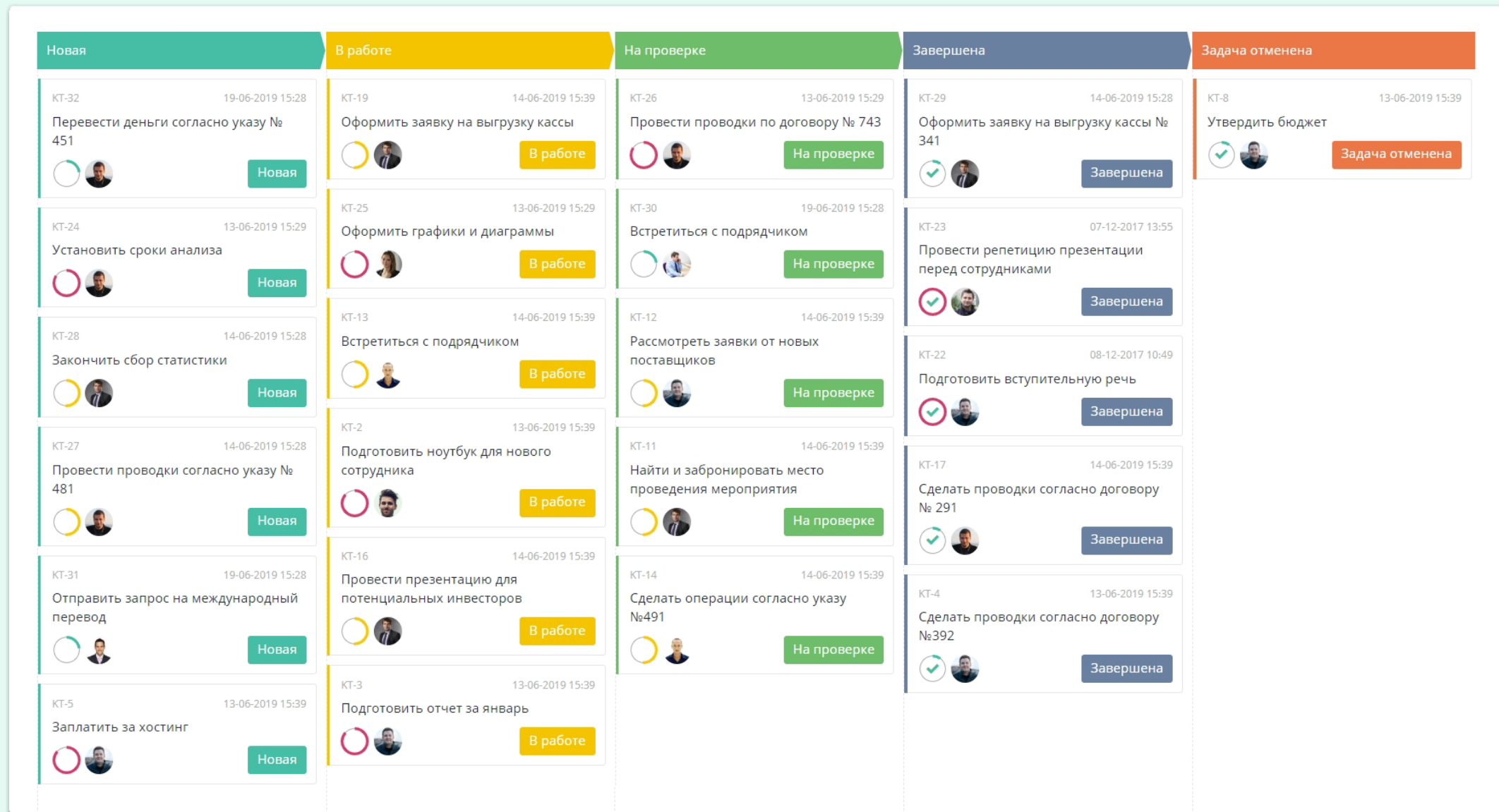
- Проста у використанні
- Сприяє безперервним та стійким поліпшенням
- Невисока вартість та можливість швидкого впровадження
- Розвиває адаптованість та співпрацю в команді

Недоліки

- Може не підходити в дуже швидко мінливому середовищі
- Відсутність чітких термінів може знизити її корисність
- Не зосереджена на стратегії чи цілях



Kanban Board





SCRUM

Одна з найбільш популярних методологій розробки ПЗ. Каркас розробки, з використанням якого люди можуть вирішувати проблеми, що з'являються, при цьому продуктивно виробляючи продукти найвищої значущості.

Scrum орієнтований на клієнта, він адаптивний і дає клієнтові можливість вносити зміни в вимоги в будь-який момент часу. Наголос робиться на самоорганізованій і багатофункціональній команді, що здатна вирішити необхідні завдання з мінімальною координацією.

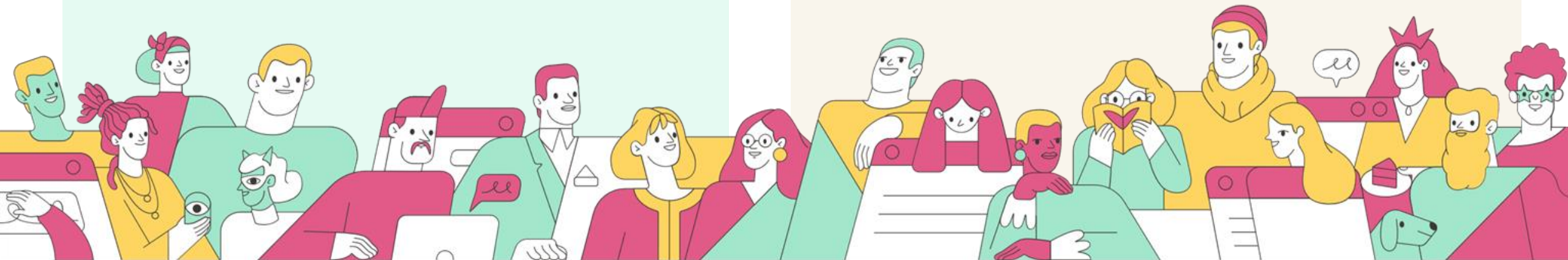
SCRUM

Переваги

- Швидкий та ефективний спосіб досягнення результатів проекту
- Великі завдання розбиті більш керовані частини
- Відгуки клієнтів швидко враховуються
- Зміни можуть бути реалізовані швидше

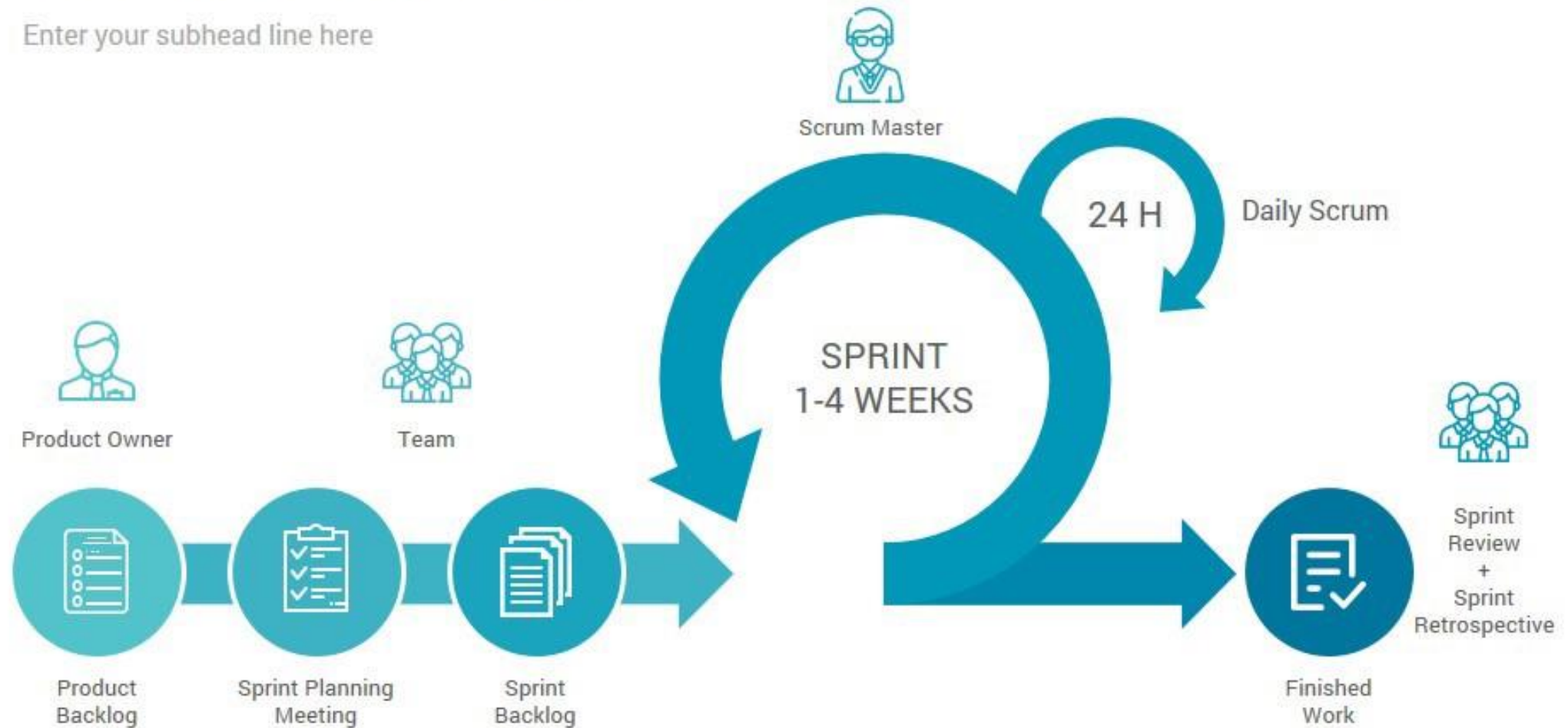
Недоліки

- Розмита мета може стати проблемою
- Необхідна досвідчена, мотивована команда
- Великі команди важко адаптуються до даної методології



Scrum Process

Enter your subhead line here





Назвіть основні ролі в Scrum

Основні ролі в Scrum

Product owner

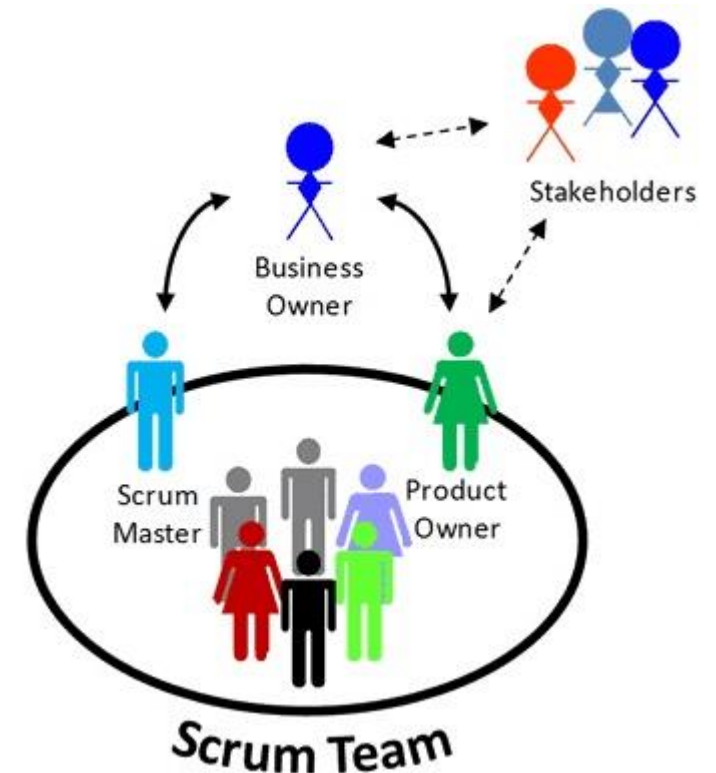
- людина, безпосередньо зацікавлена в якісному кінцевому продукті і яка розуміє, як саме цей продукт повинен виглядати/працювати
- працює на стороні замовника/клієнта (це може бути як інша компанія, так і інший відділ)
- працює з командою
- той, хто розставляє пріоритети для завдань

Scrum-майстер

- це по суті керівник проекту за методологією Scrum. Він доносить принципи методології як до своєї команди, так і до замовника, і відповідно стежить за тим, щоб всі вони їх дотримувалися

Scrum-команда

- команда, яка приймає всі принципи Scrum і готова з ними працювати



Product Owner

Владелец продукта



1 человек

Фулл-тайм или парт-тайм

Бизнес-ориентирован

Scrum Master

Скрам-мастер



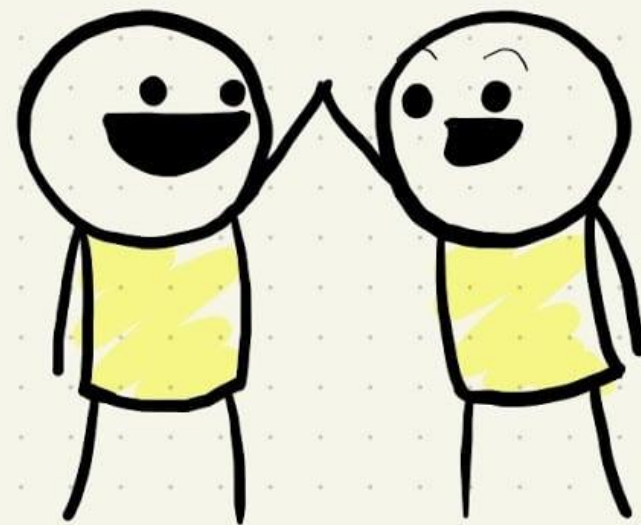
1 человек

Фулл-тайм или парт-тайм

Эксперт Scrum

Developers Team

Команда разработки



3-9 человек

Предпочтительно
фулл-тайм

Технические
специалисты

Артефакти



Беклог (backlog)

це впорядкований за пріоритетом перелік робіт, які планується виконати з урахуванням знань, що є на даний момент.



Product-беклог

це впорядкований та постійно оновлюваний список всього, що планується зробити для створення та покращення продукту.



Спринт-беклог

це список робіт, який визначила команда і погодила з власником продукту на найближчий звітний період (спринт). Завдання в спринт-беклог беруться з product-беклогу.



Користувацька історія (User Story)

це короткий та максимально зрозумілий опис функціоналу продукту або його особливостей, які отримає користувач як підсумкову цінність



Назвіть основні події в Scrum

Основні події в Scrum



DAILY STAND-UP MEETING



Time Box (15 min)



Same place



Same time



Facilitated by Scrum Master



Full team presence







Focus upon 3 questions

3 Main Questions

1. What I did yeasterday?
2. What I'll do today?
3. What's in my way?

Agile Retrospective PowerPoint Template

<p>What worked well?</p> <p>Insert your desired text here.</p> 	<p>What could have gone better?</p> <p>Insert your desired text here.</p> 
<p>What do we want to try next?</p> <p>Insert your desired text here.</p> 	<p>What questions do we have?</p> <p>Insert your desired text here.</p> 



Домашнє завдання