

## **ПРАКТИЧНА РОБОТА №7**

### **ОЦІНКА РОЗМІРУ (ЕКОНОМІЧНИХ ПОКАЗНИКІВ) В AGILE МЕТОДАХ (НА ПРИКЛАДІ SCRUM)**

Мета: ознайомитися та навчитися використовувати методи для розрахунку економічних показників розробки програмного забезпечення при застосуванні Agile методів.

#### **КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ**

Методологію Agile характеризує маніфест гнучких принципів. Принципи маніфесту AGILE:

1. Найвищим пріоритетом є задоволення клієнта шляхом ранньої та постійної доставки програмного забезпечення.
2. Зміни вимог, навіть на пізній стадії розробки. Agile процеси обробляють зміни, щоб забезпечити конкурентну перевагу замовника.
3. Поставляти робоче програмне забезпечення часто, від декількох тижнів до кількох місяців, з перевагою щодо скорочення часу.
4. Замовник та розробники повинні щодня працювати разом протягом усього проєкту.
5. Будуйте проєкти навколо мотивованих людей. Забезпечте їм необхідне середовище та підтримку та довіряйте їм виконувати роботу.
6. Найефективнішим методом передачі інформації до команди розробників та в рамках команди розробників є особиста розмова.
7. Робоче програмне забезпечення є основним показником прогресу.
8. Agile процеси сприяють сталому розвитку проєкту. Спонсори, розробники та користувачі повинні підтримувати постійний темп.
9. Постійна увага до технічної досконалості та гарного дизайну підвищує гнучкість.
10. Простота - мистецтво максимізувати обсяг роботи, яку не потрібно виконувати - має важливе значення.
11. Найкращі архітектури, вимоги та проєкти притаманні командам, що самоорганізовані.

12. Через рівні проміжки часу команда розмірковує над тим, як стати більш ефективною, а потім відповідно налаштовує та коригує свою поведінку.

Agile методологія містить декілька методів. До таких методів належать, наприклад, наступні: Dynamic System Development Method, Adaptive Software Development, Crystal Clear, Scrum, XP, Feature-Driven Development, Agile Unified Process. Серед найбільш розповсюджених є Scrum. На рис. 29 представлено Scrum Process.

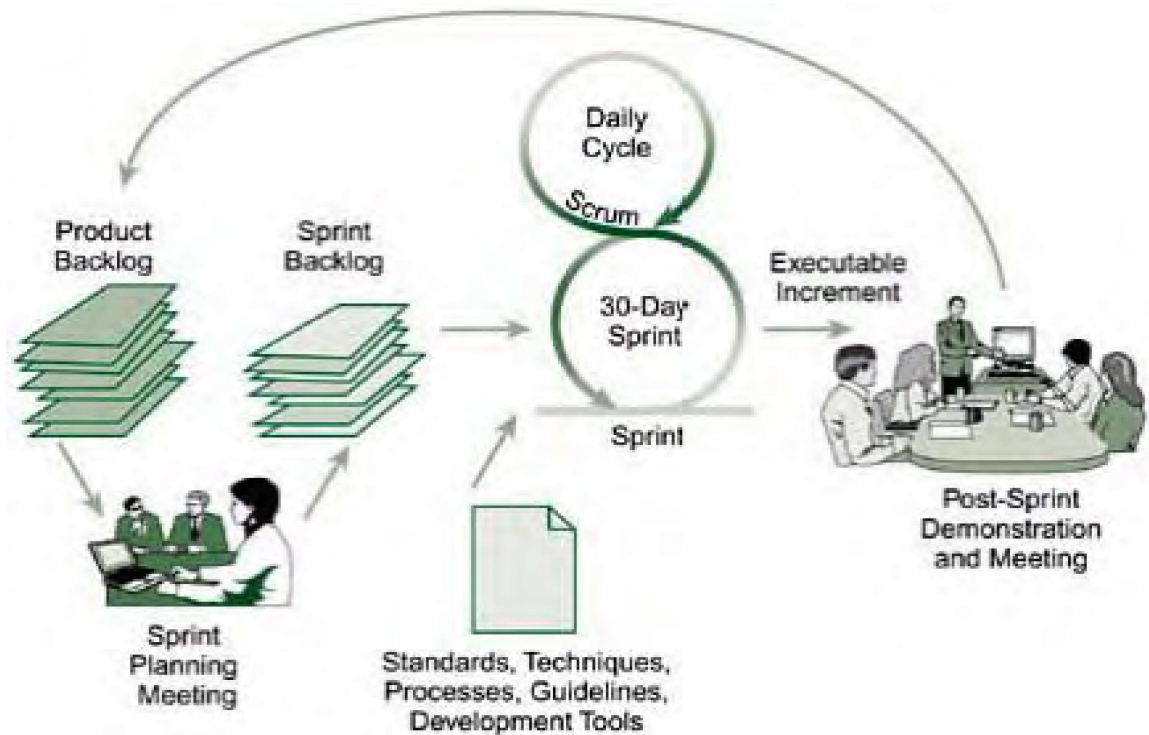


Рисунок 29 – Процеси відповідно до SCRUM

### Термінологія Scrum

Команда Scrum. Типова команда Scrum налічує від п'яти до дев'яти людей, але проекти Scrum можуть легко масштабуватися до сотень. Однак Scrum може бути легко використаний командами з однієї особи. Команда не включає жодної з традиційних ролей, таких як програміст, тестувальник або архітектор. Кожен, хто бере участь у проекті, працює разом, щоб виконати ту роботу, яку вони колективно взяли. Команди Scrum формують глибоку форму товарищескості та відчуття, що «ми всі в цьому разом».

Власник продукту (Product owner). Ключова зацікавлена сторона проекту і представляє користувачів, клієнтів та інших учасників процесу. Власником

продукту часто є хтось із керівництва продуктом або маркетингу, ключова зацікавлена сторона або ключовий користувач.

Scrum Master. Відповідає за те, щоб команда була максимально продуктивною. Майстер Scrum робить це, допомагаючи команді використовувати процес Scrum, усуваючи перешкоди для прогресу, захищаючи команду ззовні тощо.

Product Backlog. Список пріоритетних функцій, що містить усі бажані функції або зміни продукту. Примітка. Термін "відставання" може заплутати, оскільки він використовується для двох різних речей. Для уточнення, відставання товару - це список бажаних функцій продукту.

Sprint Backlog. Список завдань, які потрібно виконати в спринті. На початку кожного спринту команда вибирає певний обсяг роботи із Product Backlog та зобов'язується завершити цю роботу під час спринту. Частиною з'ясування того, які вони можуть взяти на себе зобов'язання, є створення Sprint backlog, тобто перелік завдань (та оцінка того, скільки часу триватиме кожне з них), необхідних для доставки вибраного набору елементів Product Backlog, які потрібно виконати у спринті.

В кінці кожного спринту команда робить потенційно можливий приріст функціональності продукту - тобто працююче високоякісне програмне забезпечення. Щодня під час спринту члени команди збираються, щоб обговорити свій прогрес та будь-які перешкоди для завершення роботи для цього спринту. Це відомо як Daily Scrum.

Планувальна нарада (Planning Meeting). На початку кожного спринту проводиться зустріч з планування спринту, під час якої власник продукту (owner) представляє команді пріоритетні елементи з продукту (Product Backlog). Команда Scrum обирає роботу, яку може виконати під час наступного спринту. Потім ця робота переноситься з Product Backlog до Sprint Backlog, що є переліком завдань, які команда зобов'язується виконати у спринті (рис. 30).

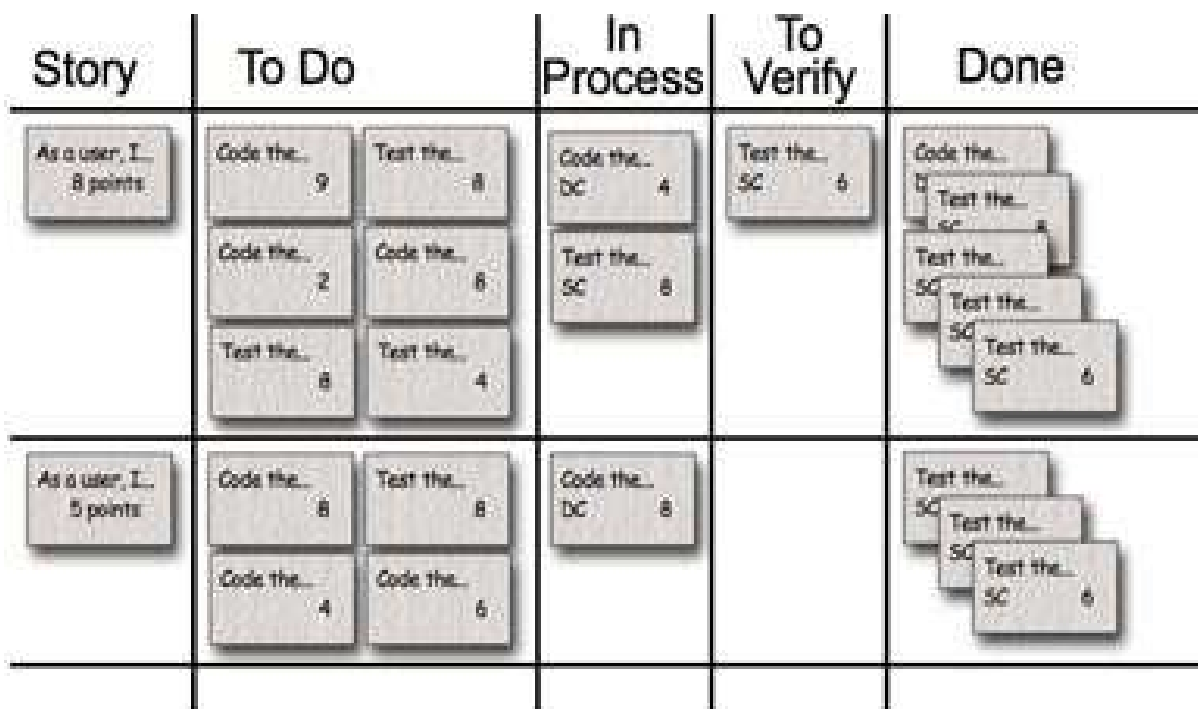


Рисунок 30 – Дошка SCRUM

Щоденна Scrum (Daily Scrum). Кожного дня під час спринту проводиться коротка зустріч, яка називається Daily Scrum. Ця зустріч допомагає встановити контекст роботи кожного дня та допомагає команді залишатися на шляху. Усі члени команди повинні відвідувати Daily Scrum.

Зустріч з огляду спринту (Sprint review meeting). Наприкінці кожного спринту команда демонструє завершену функціональність на оглядовому засіданні спринту, під час якого команда показує, що вони досягли під час спринту. Як правило, це приймає форму демонстрації нових можливостей, але в неформальній формі.

Ретроспектива спринту (Sprint retrospective). В кінці кожного спринту команда проводить ретроспективу спринту, це зустріч, під час якої команда (включаючи свого керівника Scrum та власника продукту) розмірковує про те, наскільки добре Scrum працює для них і які зміни вони можуть побажати щоб це працювало ще краще.

## Методика розв'язання типових задач (SCRUM Estimation Methods)

**1. Planning Poker.** Planning Poker є широко використовуваним методом. JIRA має для керування проектами Agile Poker. Планування Poker — це гнучка техніка оцінки, яка встановлює відносне визначення розміру в story points і гральних карток із позначеннями 0,1,2,3,5,8,13,21,34 і 55. Ці значення гральних карток представляють story points для певного item/story, за якими вимірюється команда на scrum meeting команди. Числова послідовність зазвичай послідовність Фібоначчі. Product owner або Scrum Master читає story користувача, пояснюючи всі властивості та вимоги і обговорюючи з командою усі технічні та нетехнічні вимоги для оцінок (рис. 31).

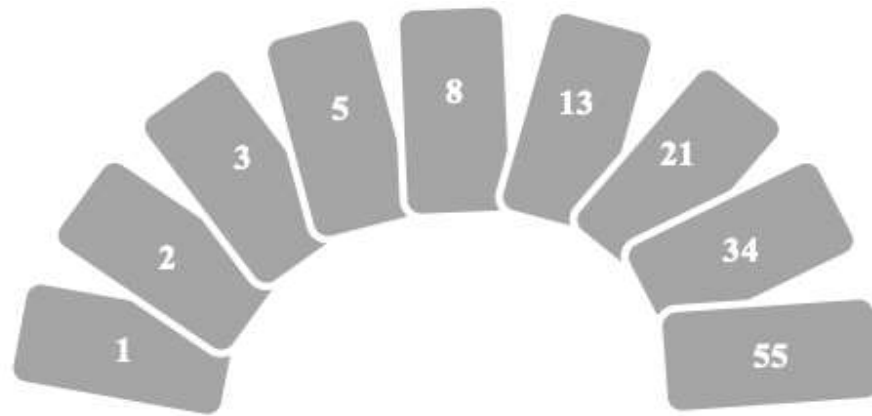


Рисунок 31 – Planning Poker

Після детального аналізу та обговорення item/story, всіх оцінювачів просять вибрати одну картку, щоб оцінити story користувача. Все оцінювачі вибирає еквівалентне значення з карт, що стає остаточною оцінкою. Якщо значення є різні, то модератори просять пояснення щодо оцінювачів, чому вони надали такі значення (враховуючи високі та низькі значення). Обговорюють вимогу до досягнення консенсусу в команді.

2. T-Shirt Size. В моделі для для приблизних оцінок використовуються розміри XS (Extra Small), S (Small), M (середній), L (великий), XL (дуже великий). Модель спрямовано зробити грубу оцінку проєкту дуже швидко. Встановити розмір (переважно середній) можливо після спільного обговорення в команді і узгоджуючи значення, присвоєне вимозі відповідно до середнього розміру (рис. 32).



Рисунок 32 – Модель приблизних оцінок

3. Bucket System. Bucket System (рис. 33), використовується для більшої кількості елементів, які потрібно оцінити а більша кількість команд. Є Bucket (множина переліків для того, що потрібно робить).



Рисунок 33 – Bucket System

Кожен перелік (Bucket) різного відносного розміру, наприклад, з розмірами 1, 2, 3, 4, 5, 8, 13, 20, 30, 50, 100, 200. Оцінювач підбирає task/story з backlog та помістити його в будь-яке відповідне поле Bucket, зазначене вище. При цьому оцінювач пояснює, чому це вміщується в цей Bucket. Дія повторюється доки backlog не буде завершено. Scrum Master перевіряє, що ніхто

не переміщує task/story, поки оцінки не будуть зроблені. Bucket System є одним із методів спільної оцінки техніки і швидко.

4. Dot Voting. Метод Dot Voting (рис. 34) - це рейтинг між історією найвищого пріоритету та історіями найнижчого пріоритету з найбільш основних завдань/історій Product Backlog, які повинні розпочатися першими. Scrum board використовується для наклепки всіх історій користувачів з жовтими наліпками та їх описом. Голосування проводиться між зацікавленими сторонами на Scrum Meeting для кожної з вимог для визначення пріоритетних вимог. Кожен учасник надає свій голос за вимогу (позитивна/негативна). Власник продукту переставляє елементи Product Backlog відповідно до кількості голосів, отриманих історією. Після завершення пріоритизації Product Backlog поділяється на 3 групи, які називаються групами високого, середнього та низького пріоритету. Дії повторюються для кожної групи до досягнення узгодження. Цей метод простий і швидкий і ефективно оцінює невелику кількість історій (до 8-10).

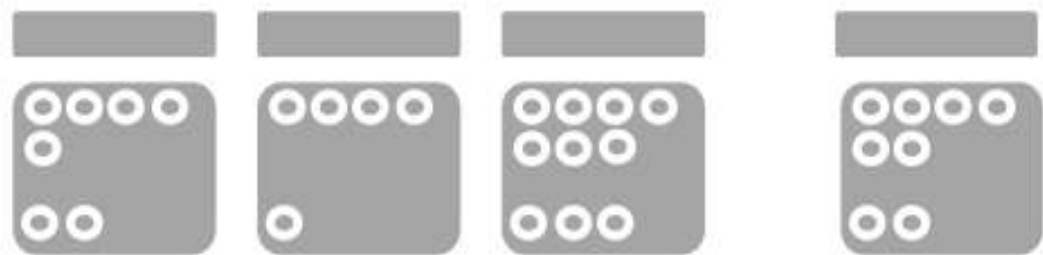


Рисунок 34 – Dot Voting

5. Large / Uncertain / Small. Існує лише три заздалегідь визначені розміри. Команда або оцінювачі оцінюють item/story у будь-яку категорію під назвою «Великий», «Невизначений» і Маленький. Цей метод оцінки використовується, коли є подібні item/story у backlog продукту. Цей спосіб схожий на Проста / Середня / Складна модель оцінок. Для кожної категорії, призначається кількість Story points, яка пов'язується з item/story.

6. Ordering Method. Цей метод оцінки підходить, коли є велика кількість елементів у Product Backlog і невелика кількість ресурсів. Випадковим чином визначаються та розміщуються порядкові шкали на діапазонах пріоритетів від низького до високого. Кожен член команди або оцінювач переміщує item з однієї шкали в іншу. Розміщенні та пріоретизовані items/stories, переміщуються вгору або донизу на одне місце, поки всі учасники команди не будуть задоволені з розміщенням у Backlog. Таким чином забезпечується пріоритетний порядок item у product backlog

7. Divide Until Maximum Size or Less. Команда Scrum визначає базове значення як максимальний розмір item для оцінки (наприклад, 8 годин зусиль) у проєкті. Кожне item у Product Backlog обговорюється щоб визначити, чи є кожен item більшим або меншим відносно оціненої базової вартості. Якщо item перевищує максимальний визначений розмір, він розбивається на кілька item, що відповідають оцінці розміру базового значення. Діяльність триває до тих пір, поки не буде розглянуто весь Product Backlog, а items знаходяться в визначеному базовому значенні.

## **ЗАВДАННЯ**

1. Створити команди та обрати один з методів оцінки. Підстави для обрання методу.
2. Створити (обрати) зміст Product Backlog.
3. Виконати оцінку розміру за обраним методом.
4. Обчислити економічні показники Product Backlog (навести опис застосування методу на конкретному Product Backlog).
5. Скласти звіт відповідно до встановлених вимог оформлення та захистити роботи..



## **КОНТРОЛЬНІ ЗАПИТАННЯ**

1. Які існують переваги та недоліки методології Agile?
2. Які методи оцінювання програмного забезпечення існують
3. Що таке підхід Scrum та які його принципи?
4. Які існують особливості оцінки розміру проєкту при використанні підходу Scrum?
5. На основі яких показників можна здійснювати порівняння методів оцінки розміру програмного забезпечення у контексті Scrum методу?