

Лабораторна робота №5

Використання сучасних гнучких методологій розроблення інформаційних систем

I. Підготовка до лабораторної роботи

Ознайомитись з теоретичною та практичною частиною наведеною нижче для виконання даної лабораторної роботи

II. Теоретичні відомості

Scrum — одна з найпопулярніших гнучких методологій розробки програмного забезпечення з сімейства Agile. Легка й доступна у використанні, але складна в засвоєнні, якщо вірити офіційному опису. На практиці вся складність зводиться до того, щоб навчити розробників та інших фахівців дотримуватися цієї методології в роботі. Але все по порядку.

По-перше, методологія — це набір правил і практик, завдяки яким краще організувати роботу над проектом. Причому краще для всіх: самої команди, компанії-розробника, менеджерів і, звичайно ж, для вас як для замовника.

По-друге, Scrum — це не якась програма та не методичка, хоча ПЗ для управління проектами на основі скрам та відповідної літератури більш ніж достатньо. Це принцип, концепція-каркас та рекомендації, як менеджеру підвищити керованість, передбачуваність та ефективність роботи.

На офіційній сторінці The Scrum Guide можна почитати докладно, хто, як і навіщо придумав Скрам, а головне, що творці вкладають у це поняття.

Є багато методів проектного управління, і вашому проекту, хоч би яким він був, потрібно вибрати один з них. І як тільки ви вирішите, що використовуватимете методологію Scrum, ваш проектний менеджер адаптує всі ці принципи, правила та практики під конкретний проект, і почнеться робота.

Отже, 4 особливості Скрам як методології:

- робота над проектом розбивається на невеликі підзадачі;
- команда з 5-7 осіб виконує кожен з них у фіксований термін (1-4 тижні);
- протягом роботи над одним підзавданням проводиться 5 типів нарад;
- отриманий результат роботи над кожним підзавданням має цінність для замовника.

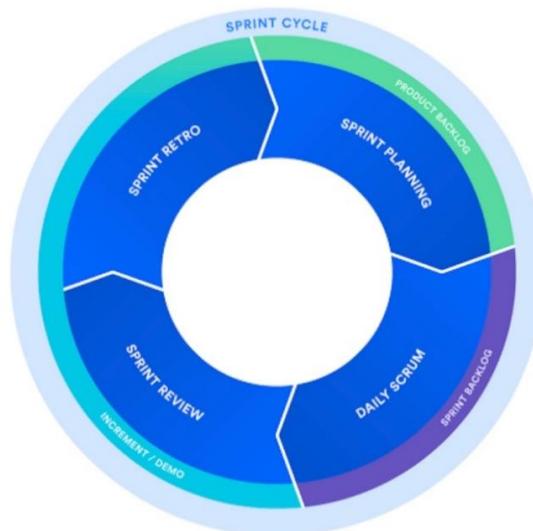


Рисунок 4.1 — Скрам

Тепер докладніше про кожен з особливостей:
Sprint

Спринт — головна фішка Скрам. Саме так називається кожне невелике підзавдання, з яких складається проект. Всі спринти повинні бути однаковими за тривалістю, та ви не повірите, але найчастіше довжина одного — два тижні, рідше за місяць. А скільки саме, залежить від особливостей вашого проекту. Зазвичай, чим складніше та незвичайніше завдання, тим спринт коротший, щоб швидше зрозуміти, скільки реально часу потрібно для досягнення більш масштабної мети, та не витратити час розробці на те, що може не знадобитися.

Загалом спринт — це про конкретні задачі. Бракувало якоїсь функції? Додали. Щось не працювало? Полагодили. Завдяки йому зручно організовувати роботу та ще зручніше стежити за прогресом проекту загалом.

Артефакти

Красивим словом «Артефакти» в Scrum називають речі, що створюються під час розробки:

1. Беклог продукту. Product Backlog — зона відповідальності власника продукту. Це список завдань або, як його називає Вікіпедія, "журнал побажань до проекту". Беклог — це не щось, що затвердили раз і назавжди, а гнучкий перелік функцій, покращень, виправлень тощо. У ньому вказуються актуальні задачі для команди та зазначаються ті, що вже виконані.

2. Беклог спринту. Ще один беклог, але менший і конкретніший. Це список завдань на конкретний спринт, який формується на мітингу щодо його планування. Він теж може змінюватися, якщо команда зіткнулася зі складнощами, і потрібно зробити щось ще, крім того, що запланували. Але його мета, вона мета спринту, залишається незмінною.

3. Інкременти. Це саме той, отриманий результат роботи над кожним підзавданням, що має цінність для замовника. Інкрементом він називається тому, що його вже можна так чи інакше додати до решти проекту та подивитися, як він працює. Це не обов'язково має бути ціла нова функція, цілком можливо й удосконалення тієї, що вже є, або взагалі виправлення помилки. Але обов'язково те, що команда мала зробити протягом спринту. Загалом це очікуваний (найчастіше) результат, який показують власнику продукту, щоб він бачив, як йде робота над його проектом.

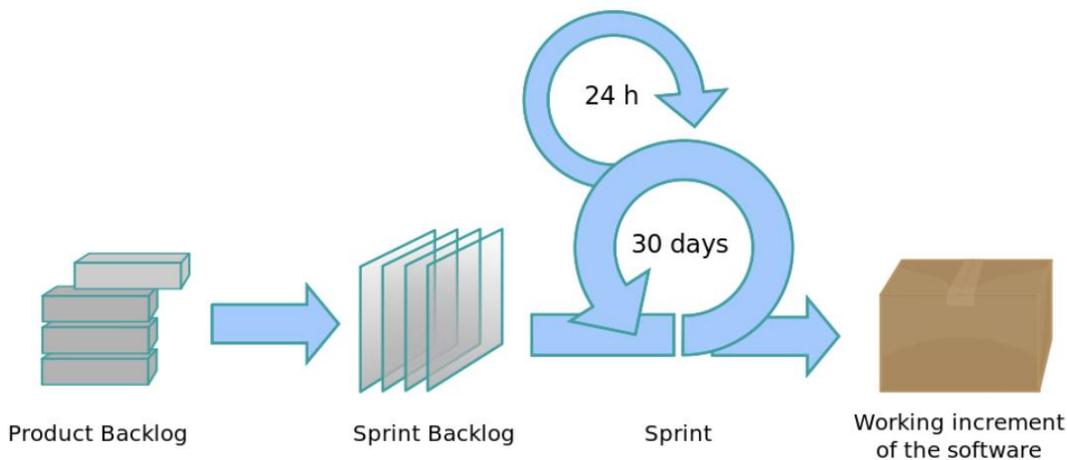


Рисунок 4.2 — Артефакти Скрам

Наради

Scrum — штука циклічна, і цей цикл складається з повторення різних нарад, вони ж мітинги:

– Project/Product backlog. Власник продукту приходить на першу таку нараду з найголовнішим артефактом проекту — підготовленим списком завдань, які потрібно вирішити для запуску. Беклог — це сучасна версія технічного завдання, в якій можна й потрібно міняти будь-що, якщо щось змінилося на ринку або в уподобаннях користувачів. У ньому зібрані та відсортовані за пріоритетом усі робочі завдання (Story, Bug, Task та інше) та в нього ж вноситься інформація про хід виконання робіт: зробили завдання — поставили галочку. На цьому мітингу всі просто ознайомлюються з тим, над чим вони мають працювати в цілому, а ось на наступному починають обговорювати.

– Sprint Planning. Планування самого спринту — обговорення найпріоритетнішого завдання командою та скрам-майстром. На цьому етапі вибираються історії з беклог, які потрібно зробити для

виконання мети спринту. Наприкінці цієї наради всі учасники команди повинні чітко розуміти, що їм слід робити.

– Daily Standup Meeting. Щодня всі учасники команди збираються в один і той же вибраний час, щоб розповісти, як у них справи. Із задачами за проектом. Кожен у двох реченнях описує, що він зробив учора та що робитиме сьогодні, а якщо зіткнувся з труднощами, то ще й про них — як вирішив чи як збирається вирішувати. Назва стендап буквально означає, що, якщо справа відбувається в офісі, то розробникам рекомендується спілкуватися стоячи. Щоб нікому не хотілося балакати довше, ніж треба. В умовах віддаленої роботи щоденні мітинги теж не затягуються — докладні обговорення проблем або ідей, що виникли, виносяться на окремі дзвони між зацікавленими учасниками, а решта відзвітували й пішли займатися своїми справами. Тому що Скрам — це про ефективність!

– Sprint Review. Наприкінці спринту, коли все готово, інкремент показують власнику продукту, а також усім, кому це цікаво, якщо досвід може бути корисним колегам. Якщо все добре, то інкремент випускають на прод, а в беклог вносяться відповідні зміни. Дуже часто цей етап плавно перетікає до першого з наступного спринту.

– Sprint Retrospective. Зустріч команди для обговорення робочих і супутніх моментів. Скрам-майстер проводить аналітику спринту, всі діляться думкою про те, як він пройшов, а також про учасників команди — хто молодець, а хто не дуже, а потім обговорюють, як працювати краще.

Дійові особи

У класичному Scrum існує 3 базові ролі:

1. Product owner (PO). Це ви як власник продукту, а найчастіше хтось із ваших співробітників, кого ви зробите відповідальним за спілкування з командою розробки. Та людина, яка створюватиме беклог проекту та доповнюватиме його, слухатиме наприкінці спринту, що ж там ця команда розробки зробила, а що ні, і що робитиме далі. PO не обов'язково повинен розумітися на технологіях розробки, але мусить бути спеціалістом у своїй галузі. Його робота — точно знати, що має робити готовий проект і кожна його частина, а також вникати в те, як розробляється.

2. Scrum master (SM). Скрам-майстер — проєкт-менедже. Його робота, з одного боку, допомагатиме продукту оунеру розібратися в нюансах роботи зі Скрам, а з іншого — організувати роботу команди. Він відповідає і за пошук кадрів для команди, і за те, щоб у них були матеріально-технічні ресурси, і загалом за те, щоб усі товаришували та ефективно працювали. Планування та проведення всіх мітингів у спринті — теж робота SM.

3. Development team (DT). Команда розробників — +/- 5 спеціалістів, які займатимуться роботою над проектом. The Scrum Guide вимагає від них не тільки навичок для виконання завдань, але ще й бути спроможними самостійно організувати робочі процеси, а також нести колективну відповідальність за досягнення мети спринту. Команд цих може бути будь-яка потрібна вашому проекту кількість, але вони повинні складатися з фахівців у певних технологіях і бути невеликими, щоб уникнути проблем із комунікацією. А комунікувати потрібно буде багато, інакше як навчитися самоорганізуватися, працювати разом, брати відповідальність, аналізувати успіхи та невдачі та робити інші речі, що постулюються методологією Скрам? Загалом, для роботи в командах Scrum мало бути хорошим технічним фахівцем, потрібні ще й soft skills вищі за середні.

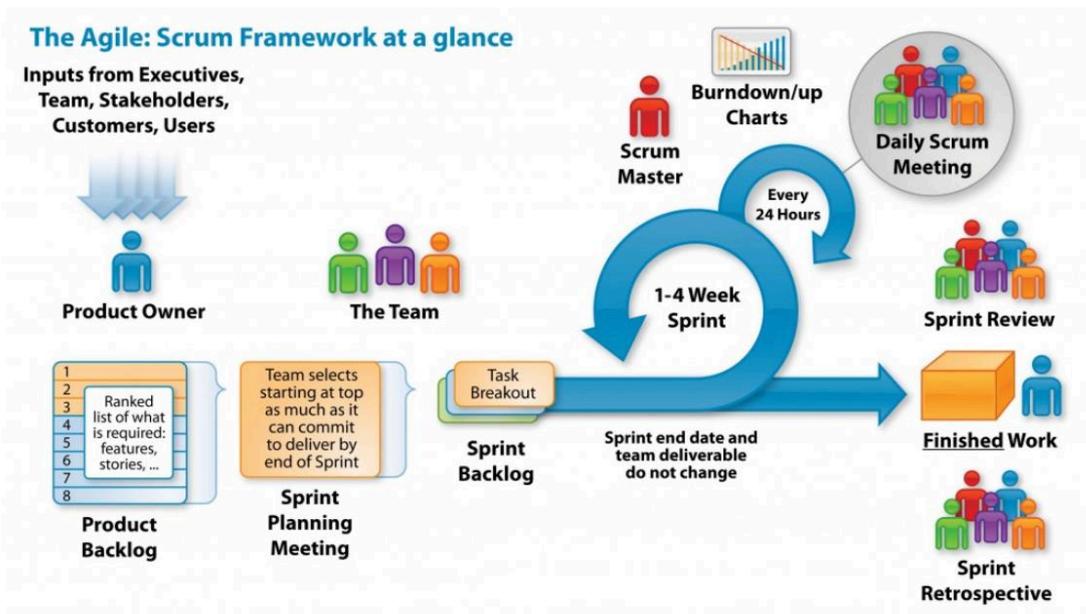


Рисунок 4.3 — Дійові особи Скрам

2 переваги та 2 недоліки Scrum

1. Гнучкість. Scrum — просто ідеальна система управління проектами, які ростуть і масштабуються, а це буквально будь-яка мобільна або веб-додаток, і навіть сайти. Сьогодні ви додали нову функцію, подивилися, як вона працює, і вже у наступному спринті можете почати її вдосконалювати, міняти чи прибрати! І це актуально не тільки для MVP, для яких кожна функція нова, але й для проектів, яким вже кілька років, і вони постійно тестують гіпотези, щоб стати кращими.

2. Видимі результати. Підсумок кожного спринту — щось реальне. Нова функція або виправлення помилки не так важливо, як можливість бачити, що робота йде, і йде успішно. Саме за це, окрім можливості міняти беклог, коли їм хочеться, і люблять Scrum власники продуктів. Учасникам команди це теж дуже важливо, оскільки умовно «закриває гештальт», дає змогу відчути задоволення від виконаної роботи.

3. Мотивація. Хотіти дотримуватися принципів Agile і робити це насправді — дві великі різниці. Знаєте, скільки людей здатні на самоорганізацію? 15%, у кращому випадку! А скільки готові погодитись на колективну відповідальність? А скільком подобаються «безкорисні» наради? Робота з усім цим і є та сама «складність в освоєнні». Тільки від скрам-майстра залежить, чи працюватиме Scrum для команди. Зробити так, щоб дорослі та розумні дяді та тьоті якщо не дружили, то поважали один одного, та розуміли важливість спільної роботи, може бути дуже складно. І хоча, з одного боку, вас як замовника не повинно хвилювати, як ваш підрядник свої справи вирішує. З іншого боку, рекомендую все-таки звернути на це увагу, якщо вам не хочеться одного дня залишитися без найважливішої для проекту команди. А, ну і будьте готові, що й вам доведеться готуватися до цих нарад раз на два тижні.

4. Невідповідність мети та інструменту. Scrum безумовно хороший для багатьох завдань, навіть не пов'язаних із розробкою. Але, при цьому, всі методології сімейства Agile об'єднує не просто терпимість, а любов до змін. Якщо вашому проекту заявлена гнучкість ні до чого, та ви впевнені, що точно знаєте, як має виглядати проект від початку й до кінця, краще вибрати класичне проектне управління, що буде значно ефективнішим.

Отже, Scrum — гнучка й неймовірно популярна методологія управління проектами. У ній великий проект розбивається на безліч маленьких підзадач-спринтів, кожна з яких виконується досвідченою та злагодженою командою в середньому за 2 тижні. Результати спринту — завжди щось цінне для проекту, що можна оцінити й протестувати в роботі. Для кожного спринту вибираються задачі зі списку-беклогу, який може вільно змінюватися відповідно до нової інформації про споживачів, ситуації на ринку та інших даних аналітики.

Головні принципи Scrum — ясність комунікації, прозорість і прагнення постійного вдосконалення.

III. Завдання на лабораторну роботу

1. Ознайомтеся з основними принципами методології Scrum.
2. Розробіть систему поділу ролей у проекту за принципами методології Scrum.
3. Визначте основні фази проекту за методологією Scrum.
4. Сформууйте журнал продукту, журнал спринту і графіка спринту.

IV. Контрольні питання

1. Що таке Scrum і як він працює?
2. Які ролі і відповідальності існують у Scrum?
3. Які засоби управління проектами використовуються в Scrum?
4. Які події відбуваються під час ітераційного процесу Scrum?
5. Які артефакти Scrum і яку роль вони відіграють у процесі розробки програмного забезпечення?
6. Як Scrum відрізняється від інших методологій розробки програмного забезпечення?