Лабораторна робота 7.
SCRUM

Мета роботи – набути практичних навичок з використання методології SCRUM ; отримати навичку створення Product Backlog та Sprint Backlog нового проекту.

Теоретична частина

***SCRUM** (рис. 1) – гнучкий фреймворк розробки ПЗ (заснований на ітеративній моделі), який передбачає розподіл проекту на ітерації (спринти) та проведення щоденних нарад (scrum), результатом яких є визначення функції системи, реалізованих за попередній день, складності та план наступного дня.*

0. Розробка Product Backlog.

Вимоги до програмного забезпечення розробляє *Product Owner* (представник інтересів кінцевого користувача).

***Product Backlog** – пріоритезований список наявних на даний момент вимог до програмного забезпечення. Product Backlog постійно переглядається та доповнюється – до нього включаються нові вимоги, видаляються непотрібні, переглядаються пріоритети.*

1. Планування спринту (*4-8 годин*).

Перед кожною ітерацією (*звичайно тривалістю 1 місяць*) вибирається список функцій системи (**Sprint Backlog**), які планується реалізувати протягом наступного спринту (*на кожну функцію не більше 12 годин (або однієї доби)*).

2. Щоденна нарада (***scrum** (Скрам), 15 хвилин максимум*).

Протягом наради кожен член команди розробників (*Developers*), що складається з 3-8 осіб, відповідає на 3 питання:

– що я зробив з моменту минулої зустрічі, щоб досягти Цілі Спринту?

– що я зроблю сьогодні для того, щоб досягти Цілі Спринту?

– чи бачу я перешкоди, які могли б ускладнити досягнення Цілі Спринту?

3. Скрам над скрамом (*ті ж 15 хвилин максимум*).

Проводиться після скраму у разі *розподіленої розробки продукту*. Об'єднана нарада кількох груп розробників, сфокусована на загальних галузях та взаємній інтеграції. Запитання ті ж, що й на скрамі (із заміною «я» на «наша команда»), плюс ще одне питання:

– чи потрібно іншій команді зробити щось із завдань вашої команди?

4. Огляд підсумків спринту (*до 4 годин*).

Проводиться наприкінці спринту. Команда демонструє приріст функціональності продукту (*Інкремент*) всім зацікавленим особам.

5. Ретроспективна нарада (*1-3 години*).

Проводиться наприкінці спринту. Члени команди висловлюють свою думку про минулий спринт і відповідають на два основні питання:

– що було зроблено добре в минулому спринті?

– що треба покращити в наступному?

Рисунок 1 – Графічне зображення SCRUM

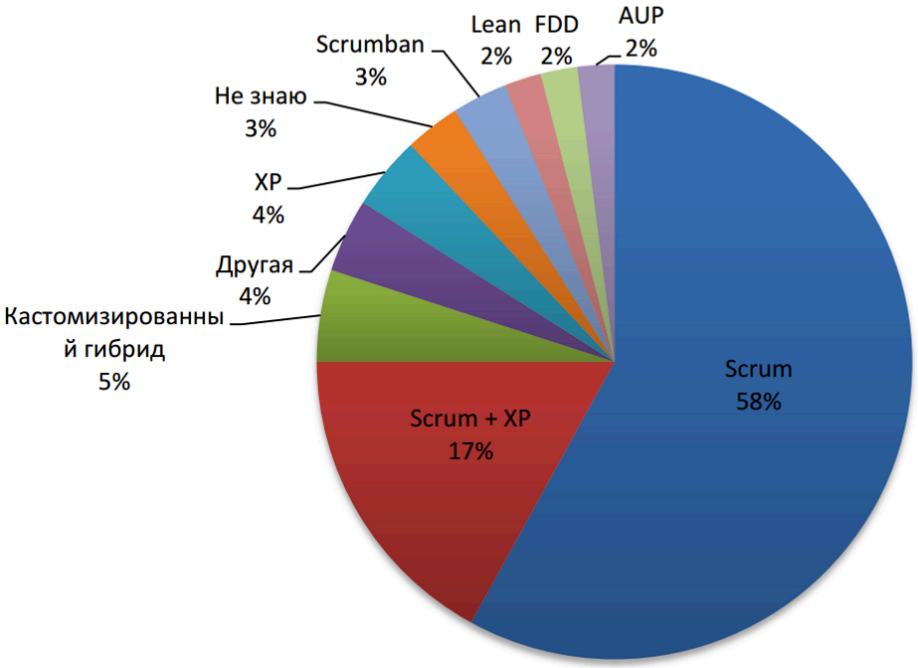


Рисунок 2 – Популярність гнучких методологій

Практична частина

1. Зберіться всією командою.

2. Ознайомтеся з основними елементами SCRUM (див. теорію та лекцію).

Презентація по Scrum https://docs.google.com/presentation/d/197bI5X8OJp0R6SVrSL0BGv8y91iQD4yW/

Лекція 9. Scrum https://drive.google.com/file/d/1y403et2bX0ibsSfNIMS6Xo8eEimRf2IF/

Лекція 10. Scrum. Part 2 https://drive.google.com/file/d/1i_OtTbQLO4tyN66TTEKzk-kNzCwQEekH/

Лекція 11. Scrum. Part 3 https://drive.google.com/file/d/1U1Gi37oVtkRGaN6xws3ZFPv_Dc-tNq35/

Посібник зі Скраму https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Ukrainian.pdf

3. Розподіліть основні ролі (відповідно до останнього керівництва Скраму варто говорити «зони відповідальності», але тут і далі буде використаний хоч і застарілий, але більш звичний термін «ролі») в команді (придумайте роль також відсутнім членам команди).

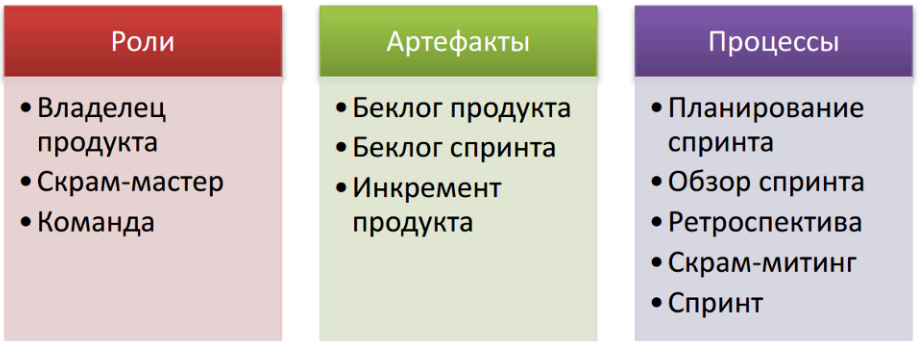


Рисунок 4 – Основні елементи SCRUM

У Scrum прийнято виділяти три основні ролі: Product owner, скрам-майстер та розробники.

Product owner (Продукт оунер, Власник продукту, Менеджер продукту) – це людина, відповідальна пріоритезацію вимог і часто за їх створення.

Як такий Product owner у вас відсутній, фактично він повинен представляти бажання майбутніх користувачів ПЗ, а як її реалізовувати і які підзавдання повинні бути вирішені визначає команда.

Тобто Product owner можете не призначати його обов'язки виконуватиме команда в цілому.

Скрам-майстер (Scrum Master) *– проводить наради, стежить за дотриманням усіх принципів SCRUM, дозволяє протиріччя і захищає команду від факторів, що відволікають. Ця роль передбачає нічого іншого, крім коректного ведення SCRUM-процесу.*

Скрам-майстер – член команди, який додатково відповідає за процеси, координацію роботи команди та підтримку соціальної атмосфери у команді.

Це власне капітан вашої команди.

Розробники – 7±2 чоловік, які реалізують вимоги власника товару.

Ваше завдання визначити ролі всередині команди:

– програміст;

– дизайнер;

– тестувальник;

– сценарист;

– геймдизайнер ...

Гейм-дизайнер – спеціаліст, який розробляє правила, стиль та дизайн комп'ютерних ігор. Роль гейм-дизайнера аналогічна режисера кіно.

Один член команди може поєднувати кілька ролей, але це небажано.

4. Складіть Product Backlog.

Шаблон для 7 л.р., Product Backlog та Sprint Backlog

Product_Backlog_Template

Беклог продукту складається з бізнес-вимог, які зазвичай оформлюються з погляду користувача (геймера, гравця).

Давайте поглянемо докладніше, що є окремим елементом Product Backlog (окрема історія користувача).

Унікальний числовий ідентифікатор історії – зазвичай збігається з ідентифікатором історії користувача із трекера, яким користується команда.

Цей ідентифікатор дозволяє точно сказати, про яку історію користувача зараз йдеться.

Назва історії користувача – короткий (приблизно до 10 слів) опис функціоналу з погляду користувача, сформульований як трійки «Роль», «Дія», «Мета». Наприклад: "Користувач вводить логін і пароль для того, щоб авторизуватися на сайті".

Важливість – унікальний числовий пріоритет історії користувача, що вона вище, тим раніше цю історію необхідно зробити.

Оцінка – числова відносна оцінка історії користувача за спеціальною шкалою у story point.

Зазначені поля зручно розміщувати на стікері, що прикріплюється на дошку. Наприклад, історію користувача для авторизації на сайті з оцінкою в 10 сторипоінтів, важливістю 200 та номером у трекері 1453, можна представити на стікері так (рис. 5):

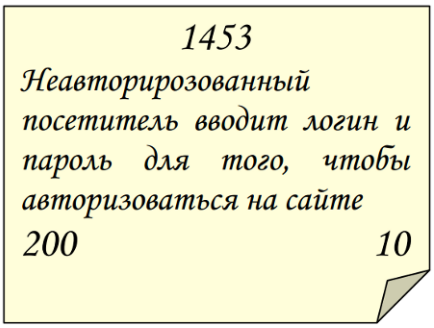


Рисунок 5 – Приклад історії користувача

Отже, ваше завдання скласти список основних елементів Product Backlog. Ввести для кожної з них ідентифікатор (достатньо їх пронумерувати) і короткий опис. Оцінки поки не розставляйте.

Цей документ слід оформити в електронному варіанті – наприклад, у вигляді Google-таблиці із загальним доступом.

Власник продукту повинен ставити перед собою та командою чіткі та зрозумілі цілі, для цього існує кілька критеріїв, які збираються до англійської абревіатури SMART ("розумний"):

Літера	Англійський термін	Український термін
S	Specific	Точні та конкретні
M	Measurable	Вимірювані
A	Achievable	Досяжні
R	Relevant	Релевантні
T	Time bound/framed	Цілі зі терміном

Приклади (рис. 6).

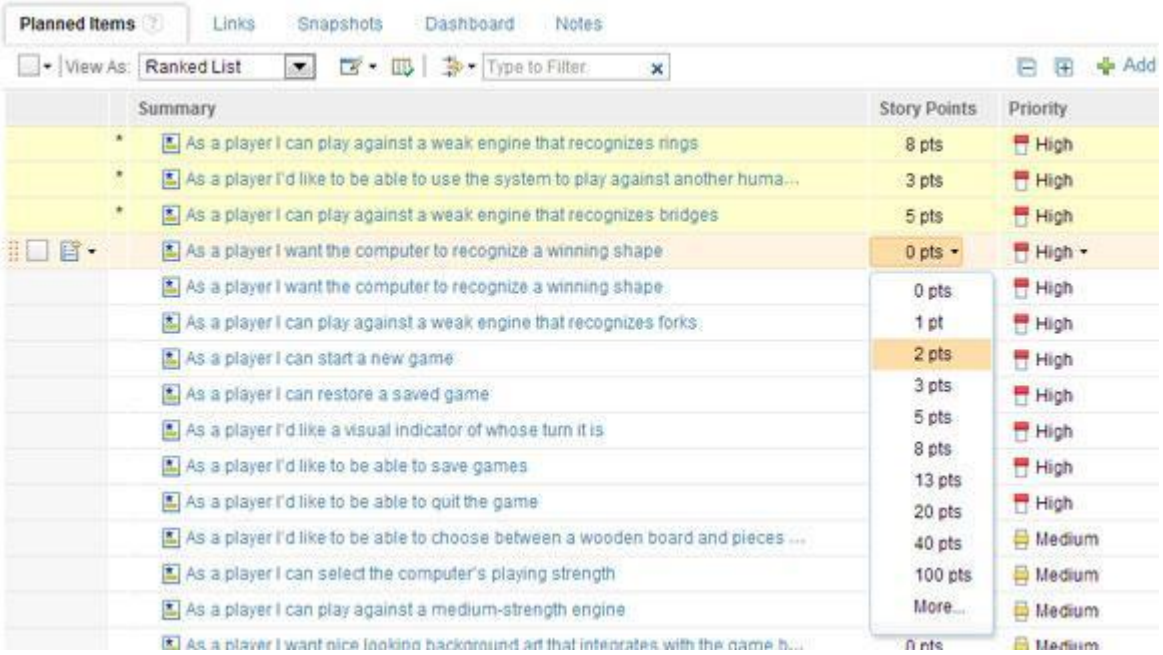
PRODUCT BACKLOG (PRIORITISED)

	FEATURE	REMARKS
Player	Can walk around as new avatar that wasn't in the Restaurant	
Player	Can pick up an object	
Player	Can put down an object	
Player	Can chat and be heard by other player	
Player	Can quit	
Player	Can use action menu from the Restaurant to do things to objects	
Player	Can look around (camera)	
Player	Can explore the test area	
Player	Can tell which object is selected for usage	
Player	Can find each other and start a game	
Player	Can hear sounds made by objects	
Team	Can place objects in the game world using text	
Player	Can explore the Crash Site	
Player	Can use appropriate props	
Player	Can spawn as man	
GM	Auto-spawns as alien	
Player	When Player walks around, avatar moves accordingly	
Player	When Player stands still, avatar moves accordingly	
Player	Can do a default attack action (extension of arm)	
Player	When player attacks, avatar moves accordingly	
Player	Can shoot a gun	
Player	Can see objectives	
Player	Can select objectives at the beginning	
Player	Can tell which character said what in chat	
GM	Can decide win/lose	Using hotkey
GM	Can decide win/lose	Using GUI
Player	Can read character's background story at start	
Player	Can control music volume	
Player	Can control SFX volume	
Player	Can run game on Windows	
Player	Can run game on Mac	
Player	Can enter and leave menu mode	
GM	Can spawn other NPCs	Using hotkey
Player	Can spawn alien	Using hotkey
Player	Is reminded by game when time has passed	Using hotkey

GM	Can control multiple NPCs	Using hotkey
GM	Can spawn Agent	Using hotkey
GM	Can spawn Scientist	Using hotkey
GM	Can spawn other NPCs	Using GUI
GM	Can spawn Alien	Using GUI
Player	is reminded by game when time has passed	Using GUI
GM	Can control multiple NPCs	Using GUI
GM	Can spawn Agent	Using GUI
GM	Can spawn Scientist	Using GUI
Team	Can place objects in game world with a textless editor	
Team	Can edit object properties in text	
Team	Can edit object properties with a textless editor	
Team	Can change sound properties of an object in text	
Team	Can change sound properties of an object with a textless editor	
GM	Can change background music	Using hotkey
GM	Can remove in-game assets	Using hotkey
GM	Can change background music	Using GUI
GM	Can remove in-game assets	Using GUI
Player	Can use emotes	
Player	Can explore Area 51	
Player	Can explore Government Building	
GM	Can go into God Mode	Using hotkey
GM	Can move characters in God Mode	Using hotkey
GM	Can alter lighting	Using hotkey
GM	Can spawn woman	Using hotkey
GM	Can spawn man	Using hotkey
GM	Can go into God Mode	Using GUI
GM	Can move characters in God Mode	Using GUI
GM	Can alter lighting	Using GUI
GM	Can spawn woman	Using GUI
GM	Can spawn man	Using GUI
Player	Hears sound when he wins or loses	
Player	Can check off done objectives	
GM	Can spawn NPCs where he desires	Using hotkey
GM	Can always see where the Players is	Using hotkey
GM	Can spawn NPCs where he desires	Using GUI
GM	Can always see where the Players is	Using GUI
Player	Can use a radial menu to do actions to objects	

GM	Can see mini-map	Using hotkey
GM	Can see mini-map	Using GUI
Player	Can sit down	
Player	Can eat things	
Player	Can get a closer view of objects ("inspect")	
Player	Can read character's background story during play	
Player	When Player uses object, avatar behaves accordingly	
Player	Can die	
Player	Can tell which object is selected for use (more polished)	
Player	Can see chat box with previous statements	
Player	Can expand chatbox to see history	
Player	Can distinguish chat and emotes/description (using icons?)	
Player	Can stand outside 50's house	
Player	Can see pretty speech bubble	
Player	Can indicate to GM when they're done with the game	
Player	Can spawn as woman	
Player	Can spawn as alien	
Player	Can fill a survey at the beginning/end	
Player	Must play tutorial	
Allie	Can play back previous playthrough for bug-testing	

ID	Theme	As a/an	I want to...	so that...	Notes	Priority	Status
2	Game	moderator	create a new game by entering a name and an optional description	I can start inviting estimators	If games cannot be saved and returned to, the description is unnecessary	Required	done
2	Game	moderator	invite estimators by giving them a url where they can access the game	we can start the game	The url should be formatted so that it's easy to give it by phone.		done
5	Game	estimator	join a game by entering my name on the page I received the url for	I can participate			done
6	Game	moderator	start a round by entering an item in a single multi-line text field	we can estimate it			done
8	Game	estimator	see the item we're estimating	I know what I'm giving an estimate for			done
~~40~~	~~Game~~	~~participant~~	~~always have the cards in the same order across multiple draws~~	~~it's easy to compare estimates~~	-	Replaced with A08 because I didn't want the story to talk about "the same order" as that might be a UI implementation detail	~~todo~~
35	Non-functional	user	have the application respond quickly to my actions	I don't get bored			done
36	Non-functional	user	have nice error pages when something goes wrong	I can trust the system and it's developers			done
A11	Non-functional	Researcher	results to be stored in a non-identifiable way	I can study the data to see things like whether estimates converged around the first opinion given by "estimator A" for example	No names or story text should be stored but we should store each card of each hand, know who played it, and know the final accepted estimate		
A05	Game	moderator	edit an item in the list of items to be estimated	so that I can make it better reflect the team's understanding of the item			
22	Archive	moderator	export a transcript of a game as a CSV file	I can further process the stories and estimates	Exported file should be directly importable back into the system.		done



Feature	Estimation, man*it	Priority
Интеграция с IntHR		
Синхронизация сотрудников, орг структуры, позиций и т.д. с учетом multisite	2	Must
ФИН		
Расчет утилизации и проч. финансовых показателей	1	Must
Расчет затрат по реальным зарплатам	1	Must
Редактирование справочных таблиц	1	Must
Закрытый период		
Заказчики		
Внутренние ставки		
Закрытый период по проектам	1	Should
Поддержка разных типов проектов	1	Should
Административный проект подразделения		
Проект обучения		
Multisite		
Синхронизация данных между серверами, управление синхронизацией	4	Must
Отчеты уровня компании	2	Must
Внешний бюджет		
Доработка интерфейса (общий стиль, тулбары, usability, AJAX)	2	Should
Additional view	2	Should
By Month		
By Person		
Прочие доходы	1	Should
Автоматизация PSR	1	Must

Элемент Бэклога Продукта	Задача	Волонтёр	Первоначальная Оценка	Новая Оценка оставшейся работы на конец дня...					
				1	2	3	4	5	6
Как покупатель, Я хочу положить книгу в корзину	Изменить БД	Санджей	5	4	3	0	0	0	
	Создать страницу (UI)	Джинг	3	3	3	2	0	0	
	Создать страницу (JavaScript)	Трейси и Сэм	2	2	2	2	1	0	
	Написать автоматические приёмочные тесты	Сара	5	5	5	5	5	0	
	Обновить раздел помощи для покупателей	Санджей и Джинг	3	3	3	3	3	0	
	. . .								
Улучшить производительность обработки транзакций	Объединить DCP код и законченные тесты слоёв		5	5	5	5	5	5	
	Закончить машинный заказ для pRank		3	3	8	8	8	8	
	Изменить DCP и читателя для использования pRank в HTTP API		5	5	5	5	5	5	
	. . .	. . .	. . .						
		Всего	50	49	48	44	43	34	

Рисунок 6 – Приклади Product Backlog

5. Вибір еталонного завдання.

Тепер кожну задачу необхідно оцінити і для цих відносних оцінок необхідно вибрати еталонну історію користувачів.

– проста та зрозуміла для команди;

– типова для цього проекту;

– невеликого розміру.

Цю історію користувача команда приймає за 1 сторіпоінт. Тоді історія користувача, яка буде вдвічі менша за еталонну, матиме розмір ½ сторіпоінт, а історія користувача, яка в п'ять разів більша за еталонну, матиме розмір 5 сторіпоінтів. Таким чином, сторіпоінт – це відносна безрозмірна одиниця виміру.

Це цікаво: для зручності менеджери переводять сторіпонти у години. Приблизно **1 сторіпоінт = 4 робочі години**, а 2 сторіпоінта = 1 робочий день. Хоча у різних командах існують різні відповідності. І головне: у Скрамі такої відповідності немає.

6. Оцінка складності інших завдань.

Для оцінки використовується дискретна логарифмічна шкала:

0 0,5 1 2 3 5 8 13 20 40 100

Оцінку ви проводитимете за допомогою покеру-планування. Для цього ви використовуватимете надані вам карти або намалюйте свої власні на шматочках паперу.

Покер-планування (Planning poker) – консенсусна відносна оцінка історій користувачів командою. Цей вид оцінки не входить до класичного Scrum, але є патерном для оцінки історій користувачів.

Планування покеру проводиться наступним чином:

1. Кожному учаснику лунає колода карток з числовими вагами для оцінки вимог.
2. Починається обговорення та оцінка чергової історії користувача: вона зачитується, команда ставить запитання власнику продукту, з'ясовує деталі, якщо це необхідно.
3. Кожен член команди робить свою оцінку, кладучи картку сорочкою догори.
4. Після того, як усі члени команди зробили оцінку – всі карти перевертаються та оцінки звіряються.
5. Якщо оцінки всіх учасників однакові – консенсусна оцінка заноситься в беклог, інакше починається повторне обговорення та проводиться другий раунд.

На повторному обговоренні пропонується виступити тим, хто поставив найменшу та найбільшу оцінку.

7. Складіть Sprint Backlog.

Тепер ви повинні вибрати з Product Backlog список завдань, які ви збираєтеся вирішити за найближчий спринт. Крім того, вам слід призначити **відповідальних** на кожне з вибраних завдань.

	Days Left in Sprint	15	13	10	8	
Who	Description	7/22/2002	7/24/2002	7/26/2002	7/31/2002	
	Total Estimated Hours:	554	458	362	270	0
-	**User's Guide**	-	-	-	-	-
SM	Start on Study Variable chapter first draft	16	16	16	16	
SM	Import chapter first draft	40	24	6	6	
SM	Export chapter first draft	24	24	24	6	
	Misc. Small Bugs					
JM	Fix connection leak	40				
JM	Delete queries	8	8			
JM	Delete analysis	8	8			
TG	Fix tear-off messaging bug	8	8			
JM	View pedigree for kindred column in a result set	2	2	2	2	
AM	Derived kindred validation	8				
	Environment					
TG	Install CVS	16	16			
TBD	Move code into CVS	40	40	40	40	
TBD	Move to JDK 1.4	8	8	8	8	
	Database					
KH	Killing Oracle sessions	8	8	8	8	
KH	Finish 2.206 database patch	8	2			
KH	Make a 2.207 database patch	8	8	8	8	
KH	Figure out why 461 indexes are created	4				

Рисунок 7 – Приклад Sprint Backlog

Якщо хочете оформити Sprint Backlog більш красиво, то можете скористатися шаблонами з Фігми (рис. 8):

https://www.figma.com/file/mVpqY6XHR7A2GneJ6876Mi/Sprint-Backlog?type=whiteboard&node-id=0%3A1 1

Або можна продовжити використовувати шаблон: Product_Backlog_Template

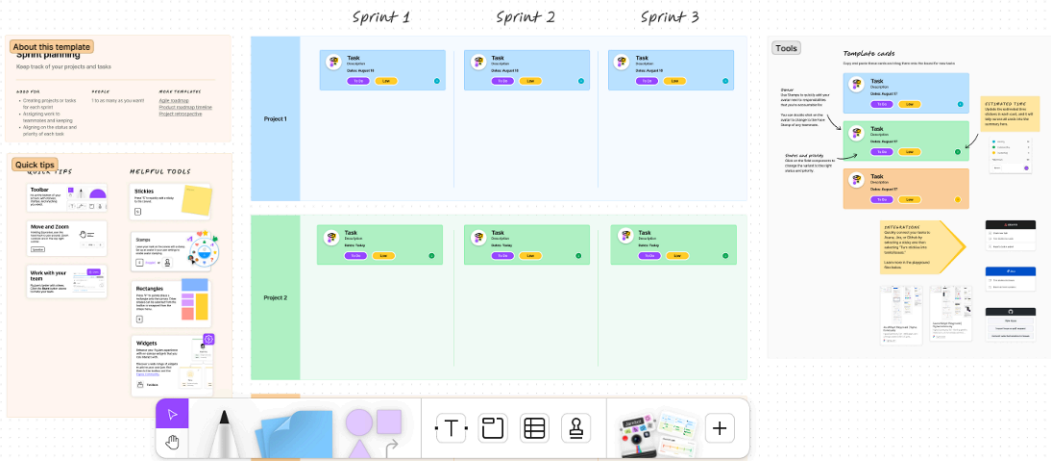


Рисунок 8 – Приклад Sprint Backlog у Figma

8. Створення проекту в Jira.

Перейдіть на сайт: https://www.atlassian.com/software/jira

Натискаємо на «Get Jira Free».

Реєструємось або входимо по існуючому акаунту.

Нас цікавить Jira Software (рис. 9). Праворуч ви можете побачити шаблони проєктів. Рекомендую обрати «Kanban» (за бажанням можете обрати і Scrum, однак Канбан буде зручніше).

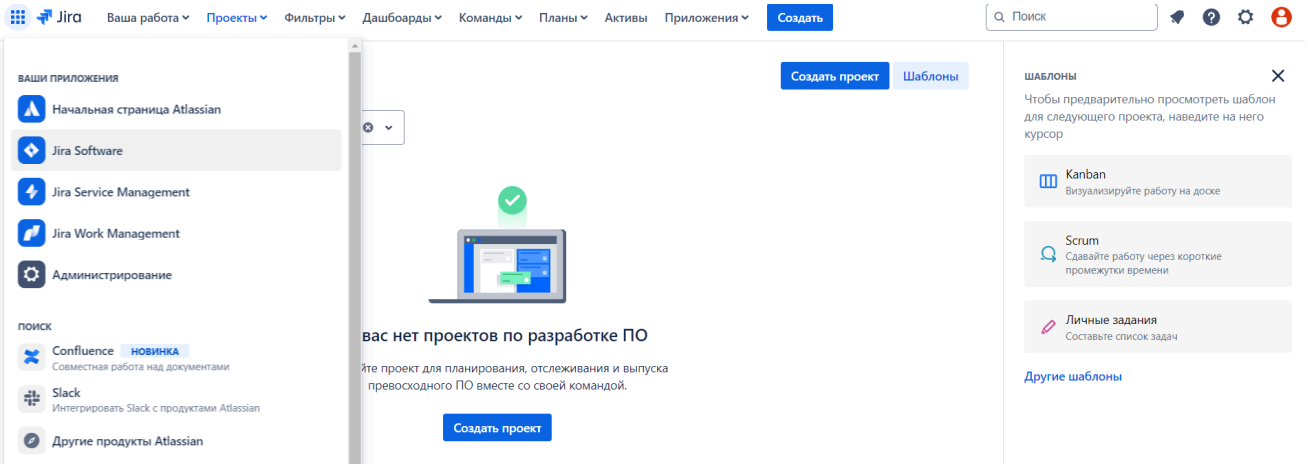


Рисунок 9 – Створення проєкту у Jira

Далі треба обрати тип проєкту (рис. 10):

1) «Керується командою»;
2) «Керується компанією».

Оскільки ми працюємо невеличкими командами над проєктом усі разом, то доцільніше обрати «Керується командою».

Тепер додайте назву до проєкта, яка відповідає назві вашого стартапу.

Нажаль, у Jira нормально не працює імпорт, тому, замість вказування назви проєкта, можна відразу перейти до імпорту (через 2 сторінки).

Буде створено пусту канбан-дошку для вашого проєкта, а праворуч можна буде побачити вікно «Quickstart», яке дозволить розібратися з тим, як працювати з системою.

Рекомендую пройти усі кроки.

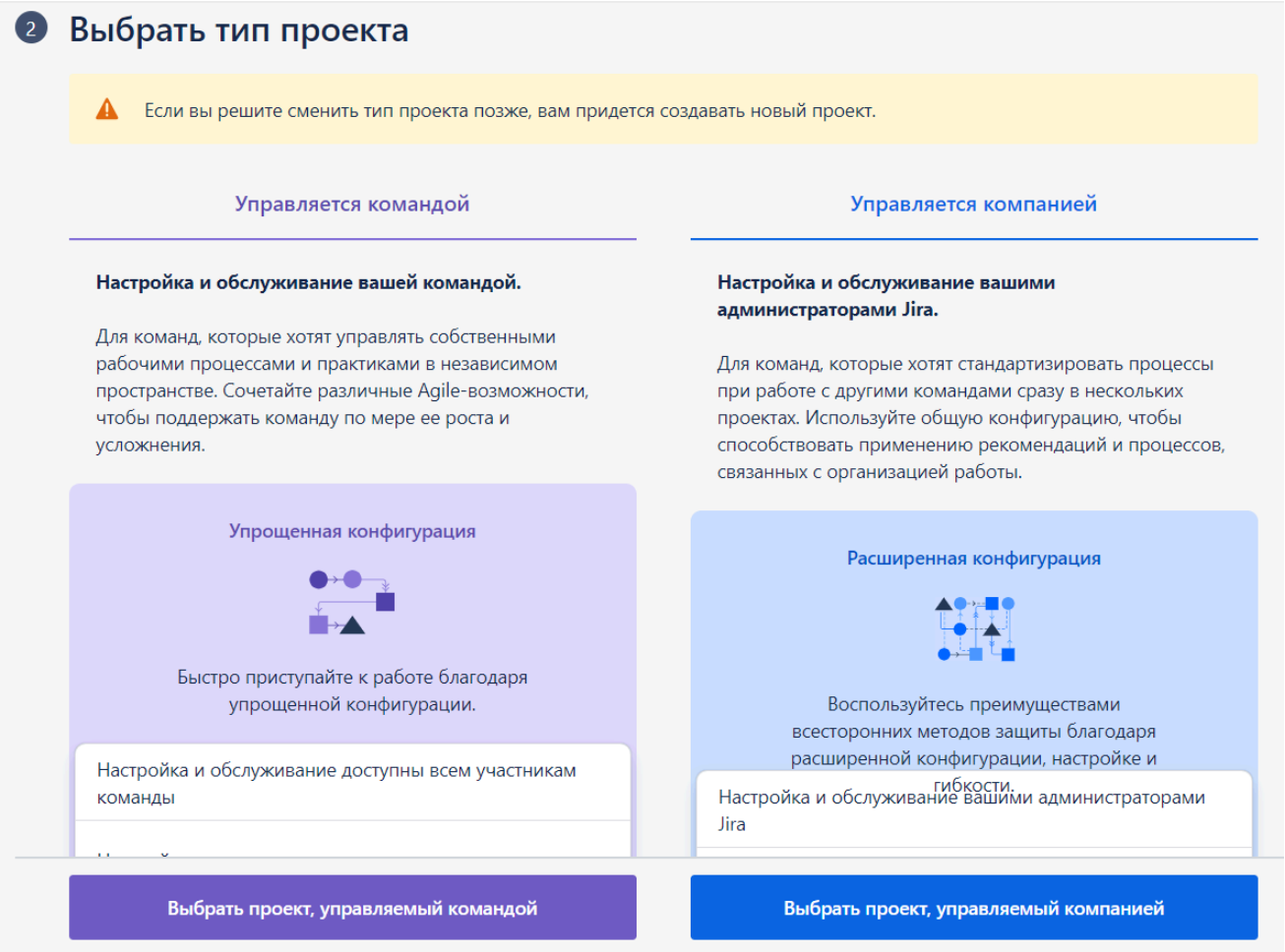


Рисунок 10 – Вибір типу проєкту у Jira

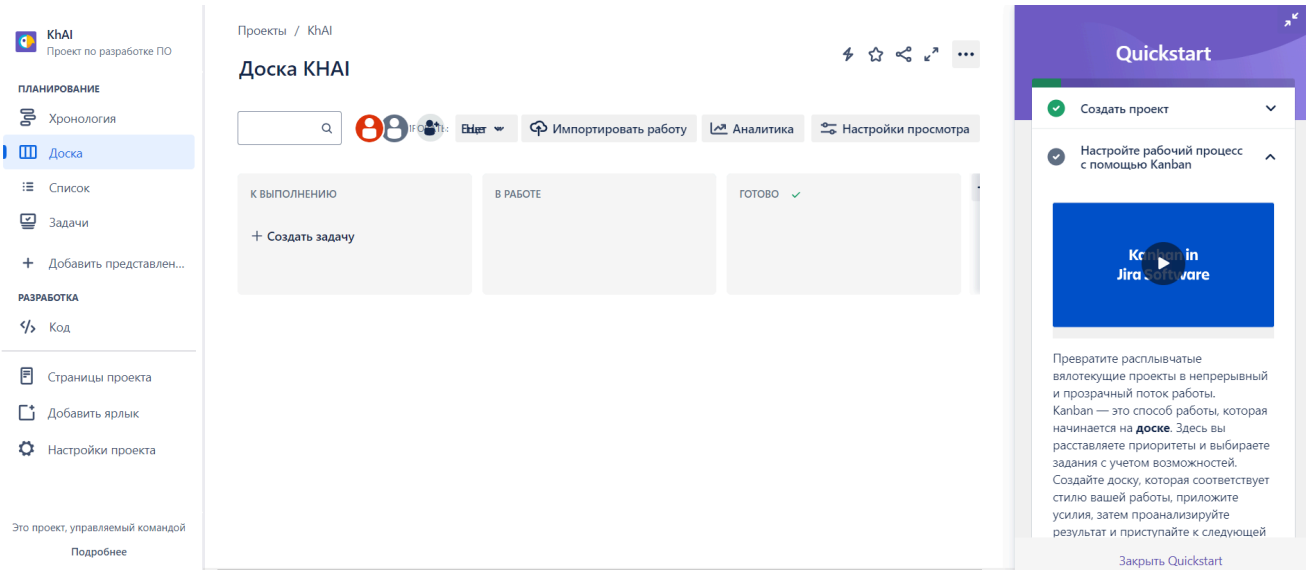


Рисунок 11 – Канбан-дошка у Jira

На третьому кроці «Quickstart» вам запропонують створити першу задачу (рис. 12).

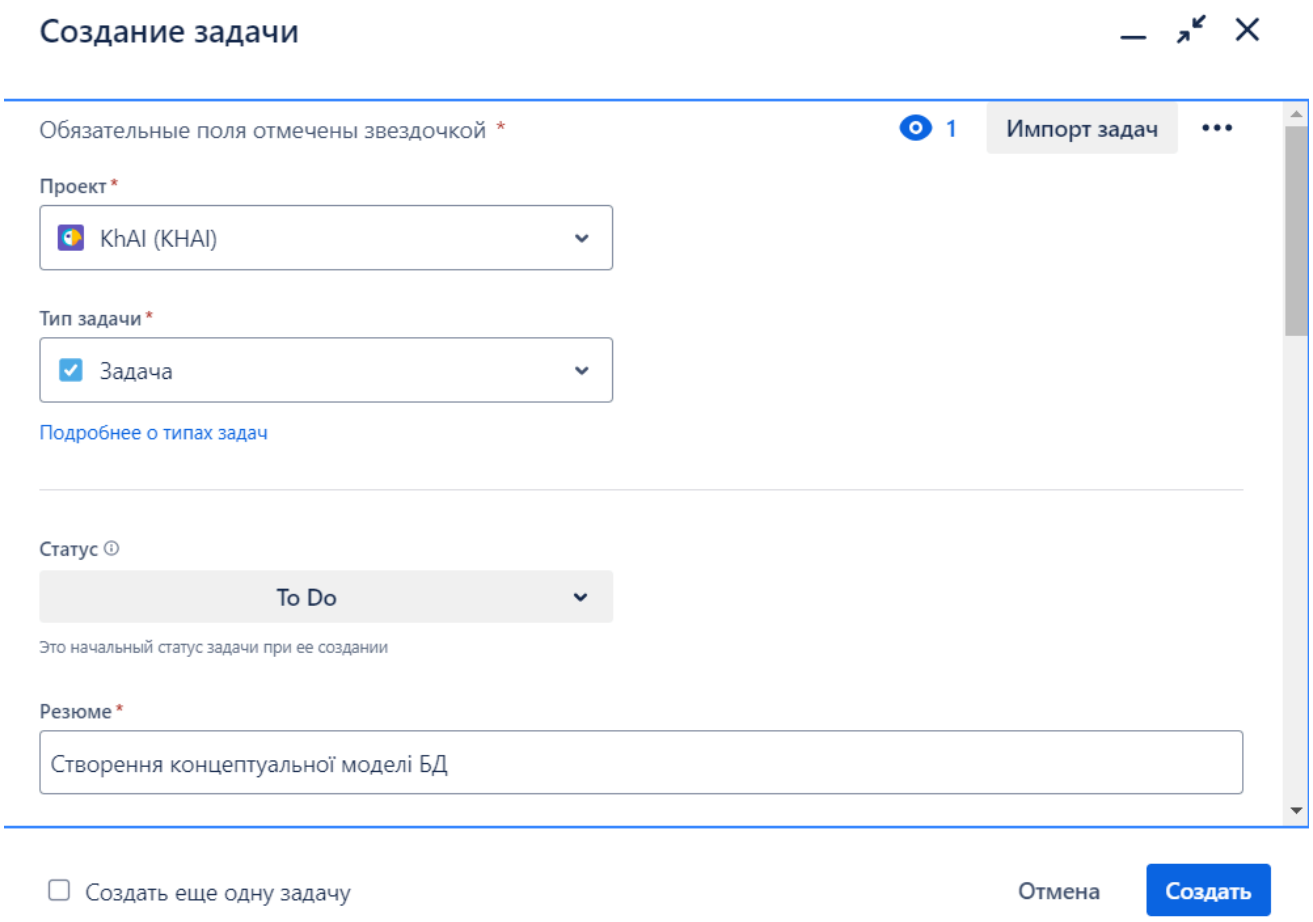


Рисунок 12 – Створення задачі у Jira

Додайте «Резюме» до задачі таке, як ви прописали в Спрінт Беклозі. Почніть з тієї задачі, яка не залежіть від інших й повинна бути виконана у першу чергу (бо інші задачі залежать від неї).

Залиште статус задачі «To Do» (якщо ви раптом вже виконали цю задачу, то – «Done», почали робити – тоді «In progress»).

Додайте розширений опис задачі у поле «Опис», сюди ж треба додати Acceptance Criteria.

Створіть усі задачі проєкту зі Спрінт Беклогу у цьому середовищі.

Щоб спростити заповнення скористайтеся **імпортом**.

1. Експортуйте вашу таблицю, отриману з Product_Backlog_Template, як CSV. У Google Таблицях: Файл → Завантажити → Comma-separated values (.csv).

Зауважте, що кожну сторінку треба завантажити окремо: Product Backlog та Sprint Backlog.

2. У Jira (маєш бути адміністратором проєкту):

Перейдіть в Jira Settings → System → External System Import (рис. 13).

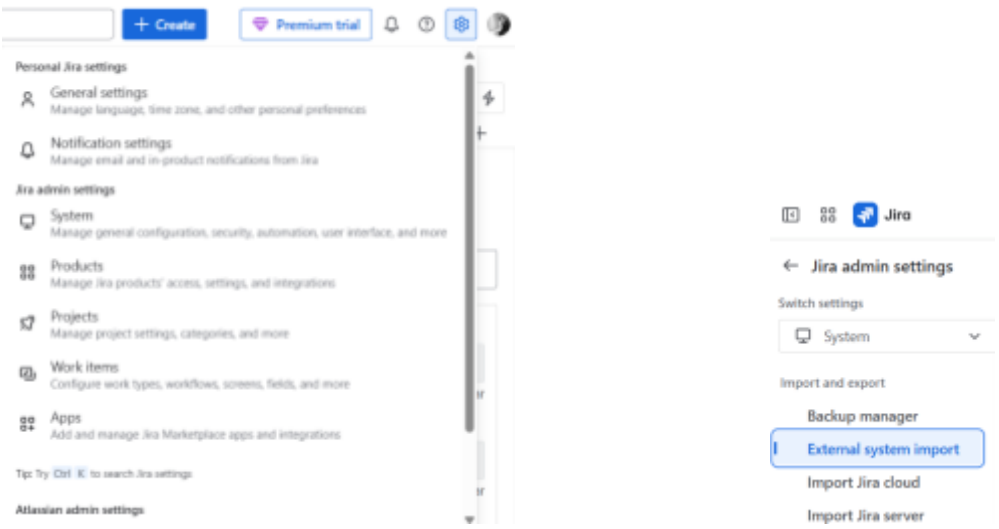


Рисунок 13 – Імпорт задач у Jira

Якщо ви бажаєте імпортувати все в новий проєкт (якого ще не існує), то знизу тицяйте на кнопку “CSV”, інакше обирайте “Switch to the old experience”.

3. Оберіть CSV
4. Завантаж свій CSV-файл
Вкажіть:
– проєкт, куди імпортувати задачі
– відповідність між колонками CSV і полями Jira (наприклад, Summary → Summary, Description → Description тощо, для тих полів, яких немає, як Acceptance Criteria, створіть власне Custome поле, а для поля Role оберіть – Labels).
5. Нажаль, чомусь за замовчанням відображення “Story point” відключено. Щоб додати оцінку складності до вашого командного проєкту:
– перейдіть до розділу Project Settings > Features (рис. 14).

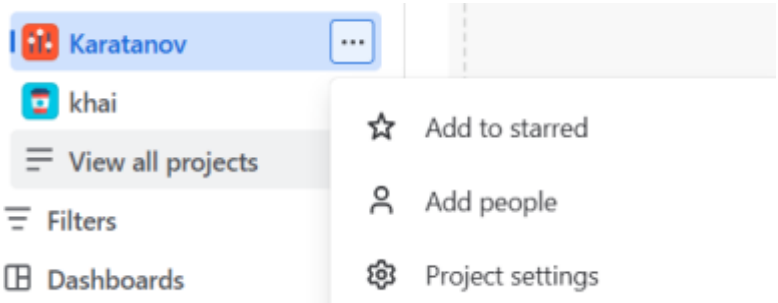


Рисунок 14 – Налаштування проєкту у Jira

– увімкніть функцію Estimation, а також інші фічі.
6. У цьому ж вікні (Project settings) натисніть Add Work Type і додайте інші типи робіт (такі як Bug та Story).
7. Також слід завітати у Notification settings (див. рис. 13, другий пункт меню) та відключити сповіщення за поштою.

Практичне завдання (резюме).

1. **Скласти Product Backlog.**
2. **Для кожного елемента Product Backlog обов'язково вказати пріоритет та відсортувати за ним.**
3. **Оцінити складність елемента Product Backlog.**

4. **Визначити Acceptance Criteria, для тих задач що будуть у найближчих спрінтах.**
5. **Вибрати елементи Product Backlog для Sprint Backlog.**
6. **Призначити відповідальних виконання завдання.**

Шаблон для 7 л.р., Product Backlog та Sprint Backlog

Product_Backlog_Template

7. **Перенести (імпортувати) Product Backlog та Sprint Backlog у Jira.**

Контрольні питання

1. Що таке Scrum?
2. Які ролі існують у Scrum та які обов'язки у кожної ролі?
3. Які події (заходи, зустрічі) входять до Scrum-процесу?
4. Що таке Product Backlog і як він використовується у Scrum?
5. Що таке Sprint Backlog і як він використовується у Scrum?
6. Що таке Scrum-зустріч, як довго вона триває, які питання можуть задаватися на цій зустрічі?
7. Що таке Sprint Review і як він використовується у Scrum?
8. Як проводиться оцінка завдань у Scrum?
9. Як команда розробки визначає, які завдання включати до Sprint Backlog?
10. Що таке Sprint Retrospective і як він використовується у Scrum?
11. Що таке Definition of Done і навіщо він потрібний у Scrum?
12. Як команда розробки повинна працювати з Product Owner у Scrum?
13. Які принципи Agile є основою Scrum?
14. Що таке Increment і як він використовується у Scrum?
15. Які метрики використовуються у Scrum для вимірювання продуктивності команди розробки?
16. Що таке Sprint Goal і навіщо він потрібний у Scrum?
17. Як Scrum може бути адаптований для роботи в проектах із розподіленими командами розробки?