

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

Навчально-науковий інститут телекомунікацій

Кафедра системного аналізу

Пояснювальна записка

до бакалаврської роботи

на ступінь вищої освіти бакалавр

на тему: «Розробка і впровадження Agile методологій в ІТ компанії»

Виконав: студент 4 курсу,

групи САД-41 спеціальності

124 Системного аналізу

(шифр і назва спеціальності)

Нещадим М.А.

(прізвище та ініціали)

Резнік С.Ю. _____

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

Київ – 2023

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ**Навчально-науковий інститут телекомунікацій**

Кафедра Системного аналізу_____

Ступінь вищої освіти - «Бакалавр»_____

Спеціальність - 124 Системний аналіз_____

ЗАТВЕРДЖУЮ

Завідувач кафедри

Системного аналізу

_____Гордієнко Т.Б

“ _____ ” _____ 2023 року

ЗАВДАННЯ**НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ**Нещадим Микиті Андрійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка і впровадження Agile методологій в ІТ компанії»

Керівник роботи _____Резнік С.Ю._____ ,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “ 15 “_лютого_2023 року №__.

2. Строк подання студентом роботи _____

3. Вхідні дані до роботи:

Trello, Jira, Asana, Monday, MS Project, kanban, agile, scrum, extreme programming, lean, PDCA, оцінка вартості сучасних рішень управління проектами,
основи проектного менеджменту, ключові підходи та методології керування ІТ проектами

Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити).

1. Провести аналіз сучасних існуючих методологій управління ІТ проектами
2. Розкрити сутність роботи РМ
3. Визначити головні складові в управлінні проектом.
4. Визначити структуру роботи проектного менеджера з метою подальшої автоматизації робочого процесу та порівняти task-менеджмент системи

Перелік ілюстративного матеріалу

1. Презентація до захисту

Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Отримання завдання на дипломну роботу та побудова плану-графіку виконання робіт	01.03.23- 09.03.23р.	
2	Аналіз літератури та джерел за темою дипломного проекту	19.03.23- 25.04.23р.	

3	Проведення консультації з науковим керівником щодо плану розділів.	05.04.23-10.04.23р.	
4	Розробка розділу 1	15.04.23-21.04.23р.	
5	Розробка розділу 2	27.04.23-01.05.23р.	
6	Розробка розділу 3	08.05.23-13.05.23р.	
7	Захист роботи	02.06-21.06	
8	Випуск	30.06	

Студент

(підпис)

М.А. Нецадим

(прізвище та ініціали)

Керівник роботи

(підпис)

Резнік С.Ю

(прізвище та ініціали)

ЗМІСТ

ЗМІСТ.....	5
ВСТУП.....	7
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ	8
РОЗДІЛ 1. ПРОДЖЕКТ МЕНЕДЖЕМЕНТ.....	9
1.1 Відмінність ІТ проектів від звичайних та класифікація факторів складності	9
1.2 Аспекти та функціональні методи керування ІТ проектами.....	10
1.2.1 Функціональні методи керування	11
1.2.2 Аспекти керування ІТ проектами	11
1.3 Життєвий цикл ІТ проекту (SDLC)	16
1.4 Обов'язки та нюанси роботи РМ.	20
Висновки до розділу 1.....	23
РОЗДІЛ 2. ПЛАНУВАННЯ ПРОЦЕСІВ, ЗАСОБИ ТА ТЕХНОЛОГІЇ КЕРУВАННЯ ІТ ПРОЕКТАМИ.....	25
2.1 Agile методології та їх фреймворки	25
2.1.1 Waterfall.....	27
2.1.2 Scrum.....	30
2.1.3 Kanban	34
2.1.4 Lean	36
2.1.5 Extreme Programming	39
Висновки до розділу 2.....	42
РОЗДІЛ 3. ПОРІВНЯЛЬНА ХАРАКТЕРИСТИКА СУЧАСНИХ МЕНЕДЖМЕНТ СИСТЕМ.....	46
3.1 Проектний підхід до керування процесом	46
3.2 Загальний огляд рішень присутніх на ІТ ринку	48
3.3 Jira.....	49

3.4 Trello.....	54
3.5 Monday	58
3.6 Оцінка та аналіз результату порівняльної характеристики продуктів.....	62
Висновки до розділу 3.....	65
ВИСНОВКИ	67
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	70

ВСТУП

Актуальність теми: Моя дипломна робота була зосереджена на аналізі інструментів управління робочими процесами які використовують провідні ІТ компанії для РМ та наданні порад щодо розробки та керування власним проектом.

В сучасних умовах існує велика кількість схожих реалізацій, які надають середовища управління ІТ-проектами, що відповідають вимогам конкретного користувача або підприємства.

У цій дипломній роботі досліджено та порівняно методи нагляду та організації діяльності, що використовуються в цих системах. Тому вважаю, що робота актуальна.

Об'єкт дослідження: гнучкі методології розробки ПЗ та їх візуалізації.

Предмет дослідження: Agile методології і таск-трекінгові системи.

Мета дослідження: В рамках цієї роботи стояла мета дослідити, які стратегії контролю та організації діяльності найефективніше застосовуються в рамках фреймворків в ІТ компаніях.

Були розглянуті як традиційні, так і сучасні методи, а також їх переваги та недоліки, частково базуючись на власному досвіді. Аналітику систем адміністрування було проведено з точки зору економічної доцільності та корисності для різних клієнтів.

Структура роботи. Робота складається з трьох розділів, вісімнадцяти підрозділів, висновків та списку використаних джерел. Загальний обсяг роботи - 77 сторінок.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

ПЗ – Програмне забезпечення

ТЗ – Технічне завдання

PMBOK – Основний довідник РМ

РМ – Проектний менеджер

БА – Бізнес аналітик

DM – Делівері менеджер

ARCH.L – Архітектор проекту

DEV – Розробник програмного забезпечення

PRG.M – Програмний менеджер

SDLC – Життєвий цикл програмного забезпечення

KPI – Ключові показники ефективності

AGILE – Філософія гнучкої розробки програмного забезпечення

SCRUM – фреймворк Agile для гнучкої розробки

KANBAN – ще один фреймворк Agile управління розробкою

QRM – Швидко реагуюче виробництво

SQL – Декларативна мова програмування для взаємодії користувача з базами даних

СТЕЙКХОЛДЕРИ – усі зацікавлені сторони бізнесу

РОЗДІЛ 1. ПРОДЖЕКТ МЕНЕДЖЕМЕНТ.

1.1 Відмінність ІТ проектів від звичайних та класифікація факторів складності

Основною відмінністю ІТ-проектів від проектів, що реалізуються в інших сферах людської діяльності, наприклад, у будівництві чи виробництві є те, що проектне керування в ІТ має справу з невленими результатами в інформаційному просторі. Крім того, ІТ-проекти мають ряд властивих лише їм факторів, що впливають на успіх реалізації проекту.

Враховуючи, що інформаційні технології стали невід'ємною частиною конкурентної переваги, більшість організацій поспішають впроваджувати інформаційні системи без ретельного планування та визначення цілей процесу інформатизації, що може мати дуже негативний вплив на результати проекту. Варто зазначити, що ІТ-проекти, як правило, є більш складними та ризикованими, ніж ті, що реалізуються в інших сферах. Складність ІТ-проекту визначається такими факторами, як:

- Кларифікованість вимог
- Часові обмеження
- Командний ресурс (стек технологій, кількість людей)
- Бюджет та амбіції клієнта
- Ситуація на ринку ІТ та загальносвітові економічні показники
- Критерії інформаційної безпеки
- Зміни у функціональному обсязі проекту
- Зміни в організаційному обсязі проекту
- Налагодженість процесів взаємодії з клієнтом
- Налагодженість процесів в команді

Перелічені основні фактори впливають на складність ІТ-проектів щодо

впровадження інформаційних систем та присутні у всіх проектах цього виду, проте остаточний набір факторів, їх вага та критичність залежать від специфіки кожного конкретного проекту.

1.2 Аспекти та функціональні методи керування ІТ проектами

Для того, щоб досягти стабільних, сприятливих результатів, проект має бути простим у керуванні. Створення таких проектів вимагає певного контролю: - Важливо, щоб у проекті брали участь ті, хто отримає найбільшу вигоду від нього. - Шанси на успіх проекту повинні бути оцінені економічно. Зазвичай, успіхом користуються бізнес моделі, які побудовані по моделі WIN-WIN, що означає для обох сторін взаємодії безпрецедентну вигоду і максимальну зацікавленість.

Отже, виходячи з цього, можна зробити висновок, що процес управління проектами це координація та передбачення якості роботи людських і матеріальних ресурсів у комбінації протягом усього її життєвого циклу проекту.

Завершенням можна визначити досягнення заздалегідь визначеної мети або визнання впливовими органами управління проектом того, що проект відповідає вимогам.

Контроль проекту передбачає структурований підхід до досягнення цілей і може здійснюватися командою проекту. Використовувані методи є невід'ємною частиною процесів управління. Для глибшого розуміння ефективного управління проектами може бути корисною комплексна модель координації, що використовує системно-орієнтований підхід. Цю модель можна розбити на три основні частини: управління цілями проекту на стратегічному рівні, управління діяльністю на операційному рівні та управління існуючою системою за допомогою інструментальних методів. Процес управління можна додатково розділити на три етапи: етап початку та будівництва, етап управління та координації, етап кризи та завершення проекту.

Успіхи розвинутих країн продемонстрували, що впровадження системи управління проектами є ефективним підходом до подолання економічних

труднощів і вирішення наукових, промислових і соціальних проблем. Ті, хто бере участь в управлінні проектами, згодом змогли об'єднати національні та міжнародні організації, а також усі інструменти, необхідні для їхньої роботи, у тому числі доступні на ринку персонального програмного забезпечення.

1.2.1 Функціональні методи керування

Для того, щоб функції управління були ефективними, необхідно, щоб усі учасники процесу були обізнані про важливі події, які необхідно виконати. Якщо є непорозуміння або некоректне спілкування, вкрай важливо розглянути та виправити проблему. Це гарантує, що всі працюють на одному боці та можуть працювати разом для досягнення бажаних цілей. Без чіткої комунікації та спільних знань, функції управління можуть бути не в змозі досягти запланованих результатів. Тому для менеджерів важливо віддавати пріоритет ефективній комунікації та співпраці в своїх командах.

До функціональних методів керування проектом відносяться:

- Створення проджект-плану
- Контроль процесів
- Адміністрування процесів та працівників
- Ризик-менеджмент
- Менеджмент можливих змін
- Аналітика і прорахування успіху функціоналу\проекту
- Оцінка результатів
- Обробка результатів
- Створення звітів
- Тестування результатів

1.2.2 Аспекти керування ІТ проектами

Успіх аспектів управління проектами значною мірою залежить від поєднання різних предметних областей і компонентів, специфічних для проекту. Ці предметні

області та керовані елементи зазвичай включають терміни, необхідні ресурси, орієнтовні витрати, надання послуг, управління ризиками та коригувальні дії.

Аспекти відіграють вирішальну роль у більшості проектів і іноді можуть вимагати використання спеціалізованих підсистем. Важливо відзначити, що існує суттєва різниця між аспектами та функціональними методами управління проектами. Перші зосереджені навколо предметної області, а другі зосереджені на процесах. Управління підсистемою відповідає за виконання всіх необхідних функцій, викладених у технічному завданні проекту.

Аспекти зазвичай розділені на окремі компоненти, якими можна ефективніше керувати.

Аспекти керування змістом

Керівництво предметної області несе відповідальність за визначення та включення критичних дій у проект. Ця предметна область визначається встановленням цілей, визначенням властивостей і структури продукту чи послуги та визначенням необхідної роботи для досягнення бажаного результату.

Процеси, пов'язані з управлінням предметною областю, включають розробку концепції, визначення обсягу проекту, розподіл робочих завдань, створення протоколів звітності, впровадження системи контролю та завершення проекту.

Зрештою, ефективне управління предметною областю має важливе значення для забезпечення успіху проекту.

Аспекти керування якістю

Управління якістю є ключовим аспектом забезпечення того, щоб проект відповідав потребам і очікуванням, заради яких він був ініційований. Це передбачає використання різноманітних методів та інструментів, таких як стандарти, правила, вимоги та системний контроль, щоб забезпечити відповідність проекту необхідним стандартам якості.

Процеси, пов'язані з управлінням якістю, включають планування якості, яке передбачає визначення відповідних стандартів успіху для проекту та визначення того, як вони будуть реалізовані.

Продуктивність проекту регулярно оцінюється, щоб переконатися, що він відповідає прийнятим стандартам якості. Нарешті, управління цим аспектом включає моніторинг і контроль за проектом, щоб гарантувати, що всі проміжні результати відповідають стандартам якості, і визначення шляхів усунення будь-якої незадовільної продуктивності.

Загалом, управління якістю має важливе значення для забезпечення відповідності проекту необхідним стандартам якості для задоволення потреб і очікувань зацікавлених сторін.

Аспекти керування часом

Управління часом є критично важливим аспектом забезпечення виконання проекту в межах установленого графіка. Це включає в собі різні процеси, такі як: визначення серії завдань, необхідних для досягнення бажаного результату, визначення зв'язку між цими завданнями, оцінка часу, який знадобиться для виконання кожного завдання, і розробка графіка проекту, який враховує як необхідний час, так і необхідні ресурси.

Крім того, ефективне управління часом передбачає постійне керування розкладом для внесення будь-яких необхідних змін.

Також, включає кілька ключових сфер діяльності, включаючи планування, необхідного для завершення проекту, оцінку тривалості виконання кожного завдання, створення календаря для вчасного виконання завдань і моніторинг прогресу в проекті для забезпечення що графік дотримується.

Ретельно керуючи часом і ресурсами протягом усього життєвого циклу проекту, можна досягти оптимальних результатів і досягти цілей проекту в установлені дедлайни.

Аспекти керування вартістю

Управління витратами охоплює низку процедур, які гарантують успішне завершення проекту в рамках заздалегідь визначеного бюджету.

Для досягнення цієї мети використовуються різні методи та інструменти управління, наприклад ті, що оцінюють ефективність інвестицій і мережевих графіків. Основні сфери діяльності в рамках управління витратами включають оцінку та прогнозування витрат, створення бюджетів та кошторисів, контроль витрат, проведення функціонального та вартісного аналізу.

Аспекти керування ризиками

Підсистема управління ризиками є ключовим аспектом управління проектом, оскільки вона має справу з непередбаченими подіями та їхнім потенційним впливом на проект. Цей процес передбачає визначення потенційних ризиків і створення плану реагування на них, враховуючи ймовірність успіху та мінімізуючи потенційні проблеми, які можуть виникнути під час закриття проекту.

Ризики характеризуються ймовірністю їх виникнення та збитками, які вони можуть завдати. Для оцінки ризиків зазвичай використовуються формальні методи, такі як розрахунки та прогнози.

Процес управління ризиками передбачає визначення плану усунення, визначення ймовірності та характеристик ризиків, оцінку впливу ризиків, збір даних щодо ймовірності та наслідків, створення процедур реагування, а також моніторинг і контроль ризиків, що залишилися, з одночасним отриманням інформації про нові.

Аспекти керування комунікаціями

Для досягнення ефективної комунікації необхідно розділити вимоги учасників проекту та кураторів і визначити, кому яка інформація потрібна, коли, в якій формі та як. Розповсюдженням інформації також необхідно ретельно керувати, щоб усі учасники отримували необхідний обсяг інформації. Звітність також є

важливим аспектом комунікаційного менеджменту, оскільки передбачає отримання та розповсюдження інформації про виконання роботи.

Нарешті, фаза завершення передбачає підготовку та видачу інформації для сприяння завершальній фазі проекту.

Ефективне управління комунікаціями вимагає пильної уваги до всіх цих сфер діяльності та використання відповідних інструментів і методів для полегшення спілкування між колегами та учасниками проекту. Цей аспект має важливе значення для забезпечення успішної підготовки, уніфікації, розподілу, збереження та остаточної обробки інформації про проект. Це вимагає встановлення процесів, які сприятимуть обміну ідеями та матеріалами між колегами.

Процес передбачає як формальні, так і неформальні методи спілкування, включаючи використання інформаційних технологій і міжособистісні контакти.

Аспекти керування працівниками

Управління людськими ресурсами передбачає використання людей у проекті найефективнішим способом. Це включає різні процеси, такі як визначення проектних ролей і обов'язків, залучення необхідного персоналу та розвиток індивідуальної та групової компетентності для покращення реалізації проекту.

Методи та інструменти управління, які використовуються переважно неформальні, стосуються службових відносин, оцінки та оплати праці, трудового законодавства, зовнішніх учасників проекту та команди проекту.

1.3 Життєвий цикл ІТ проекту (SDLC)

Кожна організація, незалежно від її розміру чи галузі, прагне до успішної реалізації проекту. Хоча кожен проект потребує унікального підходу, усі вони мають схожу структуру. По суті, кожен проект має початок, середину та кінець, незалежно від обраної методології управління. Цей проектний цикл має вирішальне значення для успіху будь-якого проекту. Життєвий цикл проекту означає серію етапів, через які повинен пройти кожен проект, від початку до завершення, незалежно від специфіки проекту. Глибоке розуміння цих етапів дає можливість менеджерам і лідерам ефективно контролювати проекти. Життєвий цикл служить для забезпечення простої структури, яка спрямовує та керує проектами. Життєвий цикл проекту має кілька переваг, включаючи покращення зв'язку між командою та клієнтами, забезпечення досягнення цілей за допомогою наявних ресурсів, а також управління й мінімізацію ризиків.

Фази життєвого циклу проекту

За даними Project Management Body of Knowledge, або PMBOK, життєвий цикл складається з п'яти фаз:

- Ініціація
- Планування
- Виконання
- Контроль
- Завершення

Як правило, етапи циклу йдуть послідовно, один за одним. Але буває інакше. Якщо під час реалізації з'являються зміни, можна повернутися на стадію планування, щоб скоригувати роботу команди проекту у майбутньому. Розглянемо кожен етап життєвого циклу окремо.

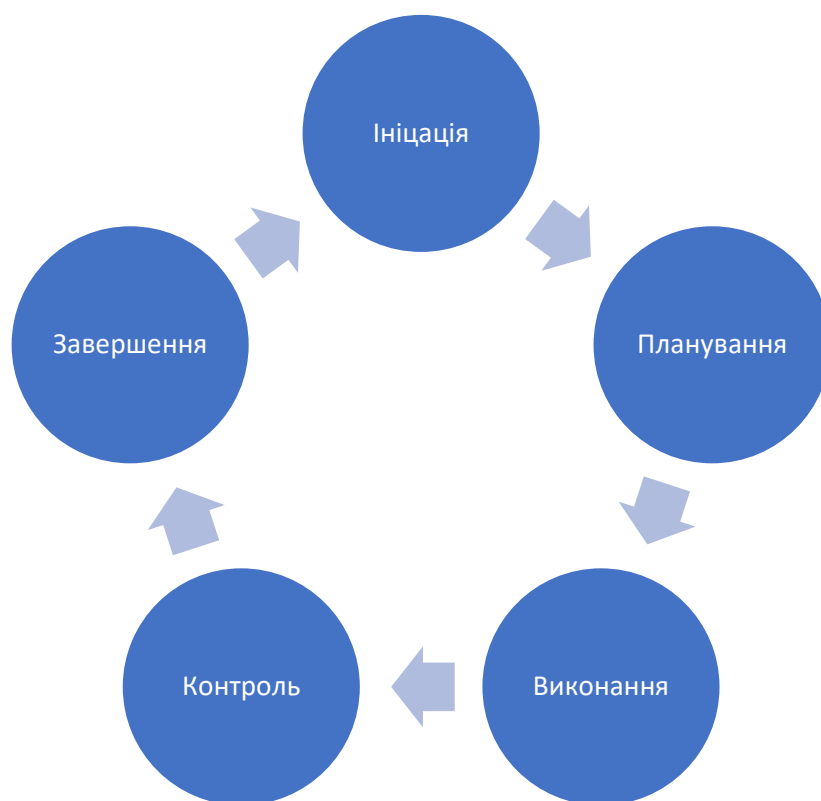


Рисунок 1.1 Схематичне відображення фаз життєвого циклу проекту

Ініціація

Ініціація означає початок роботи над проектом, що передбачає необхідну підготовку до планування та реалізації.

Перший крок — визначити завдання, яке стоїть перед нами, і оцінити, чи сприятиме воно вирішенню проблеми. У разі ствердної відповіді команда може приступати до розробки концепції та бізнес-аналізу, а також до пошуку потенційних партнерів.

Основною метою цього етапу є встановлення загальних цілей, які відповідають бажаному результату для всіх залучених сторін. Після досягнення домовленостей важливо задокументувати ключові ідеї та домовленості в статуті проекту. Цей офіційний документ є лаконічним, але суттєво важливим для планування, і на нього посилаються протягом усього життєвого циклу проекту, слугуючи засобом вирішення будь-яких розбіжностей, які виникають під час робочого процесу.

Планування

Після підписання всіх необхідних документів та затвердження умов стейкхолдерами починається етап планування. На цьому етапі менеджер розбиває робочий процес на більш дрібні завдання, формує команду, розподіляє ролі та розробляє покрокову послідовність завдань і термінів виконання. Важливо переконатися, що кожне завдання має достатньо ресурсів для успіху.

Далі створюється календарний графік із позначенням ключових ролей і сфер відповідальності, визначеними можливими ризиками, визначеними способами їх запобігання та плануванням бюджету. Проводиться організаційна нарада, на якій керівник презентує проект та його цілі, обговорює важливі етапи плану, відповідає на запитання та представляє інструменти, які будуть використовуватися. Після зустрічі члени команди повинні мати чітке уявлення про проект, його етапи та реалізацію. Рекомендується постійний доступ до проекту, щоб члени команди могли бути в курсі прогресу та змін протягом усього робочого процесу.

Виконання

Після того, як проект отримав зелене світло, можна зібрати команду та підготуватися до етапу виконання. На цьому етапі керівник зобов'язаний контролювати координацію та співпрацю між усіма відділами, забезпечуючи бездоганну співпрацю та виконання поставлених завдань. Це передбачає ретельний моніторинг прогресу та втручання, коли це необхідно, щоб підтримувати проект на шляху до успішного завершення.

Контроль

Зміни на цьому етапі є цілком нормальним явищем, і гнучкість може принести користь проекту. Бути готовим до змін – один з основних аспектів Agile підходу в ІТ сфері.

Виявлення проблем на ранній стадії дозволяє швидше їх вирішити. Ця фаза часто відбувається одночасно з фазою ініціації. Простого досягнення бажаного

результату та успішного завершення проекту недостатньо, щоб робочий процес був успішним.

Менеджер повинен постійно стежити за дотриманням командою початкового плану, щоб забезпечити максимальний успіх. На цьому етапі життєвого циклу проекту менеджер відповідає за контроль ресурсів і забезпечення виконання завдань вчасно. Вони також повинні координувати роботу членів команди та оперативно вносити корективи в план проекту, якщо виникають непередбачені обставини.

Завершення

Фінальна стадія проекту знаменує собою його офіційне завершення, але важливо не кидатися відразу на інші завдання. Щоб залишити позитивне враження від професіоналізму та роботи колективу, досвідчений керівник повинен зробити кілька кроків.

По-перше, вони повинні передати проект і всю супутню документацію клієнту або команді, яка візьме на себе роботу в майбутньому.

По-друге, менеджер повинен провести нараду «робота над помилками», де команда може поміркувати про успіхи та невдачі проекту та отримати цінний досвід.

По-третє, керівник повинен звітувати про успіх проекту перед клієнтами та менеджерами, які інвестували в результати команди. Нарешті, документація повинна бути розміщена в централізованому сховищі для легкого доступу в майбутньому.

Дотримуючись цих кроків, команда може забезпечити гладке та успішне завершення свого проекту.

Управління проектом часто може здаватися складним, але розбиття проекту на п'ять чітких фаз, відомих як життєвий цикл проекту, може допомогти забезпечити ефективне використання ресурсів і часу. Це значно підвищує ймовірність успіху для проектів різного розміру та рівня складності. Варто зазначити, що проект офіційно завершено, коли етапи пройдені.

1.4 Обов'язки та нюанси роботи РМ.

Основний обов'язок проектного менеджера полягає в тому, щоб переконатися, що бачення клієнта буде реалізовано в межах визначеного часу та виділених ресурсів.

Отримавши проект, керівник проекту повинен встановити процеси розробки, делегувати завдання команді, скоригувати процес розробки за потреби, сприяти позитивному робочому середовищу, встановити відкриті канали спілкування між командами та клієнтом, передбачати та запобігати перешкодам, а також контролювати проект. терміни, якість і послідовність.

Менеджер проекту — це експерт, який відповідає в основному за нагляд і управління загальним прогресом проекту. До їхніх повсякденних обов'язків входить встановлення пріоритетів завдань, планування виконання роботи, розподіл і делегування завдань, нагляд за взаємодією співробітників, встановлення каналів зв'язку та швидке вирішення будь-яких проблем, які можуть виникнути.

Обов'язки РМ'а:

- Аналіз плану проекту
- Аналіз та розробка стратегії для нівелювання можливих ризиків
- Коректна комунікація з клієнтом, збір вимог, вміння задавати правильні питання
- Онбоардити потенційних кандидатів на певні ролі в команді
- Відповідати за релевантний склад команди
- Розробка та ведення усієї можливої проектної документації
- Обговорення дедлайнів з клієнтом та створення плану комунікації
- Налаштування робочих процесів
- Декомпозивання проекту на дрібні логічні блоки і грамотний розподіл між виконавцями
- Визначення пріоритетності плану робіт
- Контроль роботи команди

- Відстеження морального стану та завантаженості кожного окремого працівника
- Вміння надихати та мотивувати команду досягти поставлених цілей
- Створення довірливих відносин між стейкхолдерами
- Керування відповідністю результатів роботи очікуванням клієнта
- Звітувати клієнту, щодо статусу виконання задач
- Знатися на всіх можливих методологіях управління проектами та вміти приймати рішення про вибір правильного методу відштовхуючись від ситуації

Робочий процес є важливим аспектом плану управління проектом, оскільки він визначає необхідні кроки для виконання завдань і способи переходу між ними. Проекти зазвичай обмежені за часом і служать певній меті, і успіх кожного завдання в рамках проекту залежить від проходження етапів у значущій послідовності.

Незалежно від розміру чи складності проекту, він матиме власну схему робочого процесу.

Щоб забезпечити плавний і успішний проект, ефективна система управління проектом повинна організовувати та документувати етапи робочого процесу. Добре продуманий робочий процес має забезпечувати чіткий шлях до кінцевого пункту призначення, що призведе до найефективнішого та ефективного результату.

Проектна політика також має вирішальне значення для керівників проектів. Ці політики використовуються для встановлення основних угод з командою та клієнтом, визначення меж співпраці та встановлення основних правил для досягнення цілей проекту. Вони відіграють важливу роль в оцінці придатності виконання завдань або планів дій як усередині, так і ззовні. До політики проекту часто звертаються в проблемних ситуаціях, щоб допомогти знайти рішення та вирішити проблеми.

З цих причин вони є суттєво важливим компонентом ефективного управління проектами. Для професіоналів, які контролюють проекти, таких як менеджери проектів, бізнес-аналітики, делівері менеджери та операційні менеджери, система управління є важливим інструментом для ефективної організації роботи. Системи управління проектами розроблені для вирішення фундаментальних питань, які

мають вирішальне значення для успішного виконання проекту. Ці запитання включають: які цінності ми поділяємо як команда, і як ми можемо застосувати ці цінності до проекту нашого клієнта? Як ми можемо спланувати та скоординувати нашу командну роботу для досягнення цілей проекту? Наскільки ми здатні адаптуватися у своїй проектній діяльності та наскільки добре ми можемо впоратися зі змінами?

Відповівши на ці запитання, проектні групи можуть створити міцну основу для ефективного управління проектами.

Модель делівері, заснована на типі життєвого циклу розробки програмного забезпечення (SDLC) і SDLC KPI, забезпечує гнучку структуру для планування та управління доставкою функціональності.

Ця модель спрощує процес планування як для компанії, так і для клієнта, а також дозволяє їм швидше зрозуміти та керувати наданням функцій. Структура команди та особливості кожного етапу доставки і моніторинг процесу впливатимуть на загальну ефективність та продуктивність команди. Успішна команда проекту завжди керується набором керівних принципів.

Ці принципи, засновані на принципах якісної роботи, можуть інформувати повсякденні звички та поведінку команди. Також може бути корисним розуміння головних методів мотивації і пускових механізмів команди. Люди є найважливішим активом будь-якої компанії, і команда виступає не лише як постачальник проектних послуг, але й як ключовий компонент у створенні плану та культури майбутніх робочих взаємодій.

Ефективна співпраця з клієнтами залежить від єдності цілей і перспектив команди. Щоб визначити специфіку дизайну проекту, важливо розуміти поведінку об'єктів, які будуть використовуватися в роботі. Команді потрібна модель поточного стану, щоб бути в курсі прогресу та майбутнього розвитку проекту, що є важливим для контролю змін стратегії. Ця модель також дозволяє створювати звіти на основі поточних даних для клієнта.

Основними показниками для вимірювання ефективності роботи команди є:

- якість роботи команди
- точність розподілу часу
- зростання проекту та його учасників
- рівень задоволеності співпрацею клієнта
- дотримання вимог і стандартів якості.

Робота над проектами може бути складним завданням як для команди проекту, так і для керівника, оскільки кожен проект унікальний і вимагає індивідуального підходу. Немає універсальної системи, яка б могла вирішити всі проблеми, пов'язані з проектом. Найбільша проблема полягає в етапі впровадження, який можна розбити на кілька частин.

По-перше, необхідно навчити команду проекту, оскільки вони мають різний досвід і мають різний рівень досвіду та знань.

По-друге, вкрай важливо встановити спільне бачення серед команди та клієнта щодо завдань, які необхідно виконати в рамках.

Крім того, структура проекту повинна бути достатньо гнучкою, щоб враховувати будь-які зміни, які можуть виникнути в ході проекту. Це вимагає пильної уваги до деталей і повного залучення команди впровадження, що може бути складним завданням. До ключових факторів, що впливають на успіх проекту, належать витрачений час і рівень участі команди проекту.

Висновки до розділу 1

Система керування завданнями необхідна для підтримки будь-якого проекту, і важливо створити модель із відкритим вихідним кодом, яку можна вдосконалювати шляхом практичного тестування.

Для удосконалення моделі рекомендується визначити пріоритетність комплексного управлінського бачення, проаналізувати наслідки змін, використовувати універсальні засоби розробки та розглянути альтернативні погляди впровадження.

Вибираючи універсальний фреймворк, важливо враховувати різні домени, досвід реалізації та потенційні шляхи розвитку фреймворку. Ці теми можуть допомогти у виборі та створенні методів застосування, які відповідають запропонованій практичній моделі.

РОЗДІЛ 2. ПЛАНУВАННЯ ПРОЦЕСІВ, ЗАСОБИ ТА ТЕХНОЛОГІЇ КЕРУВАННЯ ІТ ПРОЕКТАМИ

2.1 Agile методології та їх фреймворки

Знайомство з архітектурою програм є вирішальним кроком в управлінні проектами. Різні методології пропонують різні стратегії, які можуть сприяти успіху проекту. Добре підібрана методологія враховує унікальні характеристики проекту та забезпечує ефективні принципи управління, командної роботи, контролю, оцінки тощо.

Щоб вибрати правильний підхід до управління проектами, важливо добре розуміти різні доступні методології. Також важливо розглянути переваги та недоліки кожної методики, перш ніж приймати рішення. Багато керівників проектів вважають методології Kanban, Scrum та Waterfall основою розробки проекту, тоді як інші розглядають їх як відправну точку для розвитку. Зрештою, успіх проекту залежить від вибору найбільш підходящої методології для конкретного проекту. Деякі методології надають пріоритет швидкості впровадження проекту, тоді як інші зосереджуються на комплексному управлінні чи спільних зусиллях.

У сучасному світі технології стали невід’ємною частиною нашого життя, а оцифрування набирає обертів. Як наслідок, програмне забезпечення стає все більш важливим для підприємств, щоб покращити свої можливості обслуговування та швидко реагувати на потреби клієнтів. Компанії, які займаються розробкою програмного забезпечення, стикаються з величезним тиском щодо впровадження нових продуктів і модифікації існуючих рішень, щоб не відставати від конкуренції. Ключ до успіху на цій конкурентній арені полягає у відкритості до змін, оновленні діяльності та управлінні розвитком за допомогою формальних і неформальних процедур і механізмів контролю. Продуктивність окремих людей і команд, час виходу на ринок, зрілість процесу та якість коду є критично важливими параметрами, які організації повинні збалансувати, щоб ефективно поставляти програмне забезпечення.

Однак з плином часу знайти правильний баланс стає все важче. Намагання швидко реагувати на потреби клієнтів зростає, що штовхає організації до прийняття гнучких методів як основного підходу до розробки програмного забезпечення. Гнучкі методи пропонують нові можливості в забезпеченні якості програмного забезпечення, оскільки вони дозволяють отримувати постійний зворотний зв'язок і змінювати вимоги протягом усього циклу розробки програмного забезпечення.



Рисунок 2.1 Цикл гнучкого підходу до розробки ІТ проектів

Цей підхід підтримує тісну співпрацю між клієнтами та розробниками та дозволяє ранні та часті випуски програмного забезпечення необхідних системних функцій. Таким чином, впровадження гнучких технологій розробки програмного забезпечення має важливе значення для того, щоб компанії залишалися конкурентоспроможними в сучасному світі, що швидко змінюється. Це не тільки задовольняє потреби клієнтів, але й дозволяє змінювати вимоги аж до моменту розгортання продукту. Ключ до успіху полягає в управлінні розробкою за

допомогою формальних і неформальних процедур і механізмів контролю, а також в збалансованості критичних параметрів, таких як продуктивність, час виходу на ринок, зрілість процесу та якість коду.

Маніфест Agile – основа, на якій побудовані методи гнучких технологій розробки ПЗ. Він був опублікований групою розробників ПЗ та консультантів у 2001 році.

Відповідно до маніфесту Agile: *Ми постійно відкриваємо для себе більш досконалі методи розробки ПЗ, займаючись розробкою безпосередньо і допомагаючи в цьому іншим. Завдяки виконаній роботі ми змогли усвідомити, що :*

- Люди і взаємодія важливіше процесів та інструментів
- Працюючий продукт важливіше вичерпної документації
- Співпраця з замовником важливіше узгодження умов контракту
- Готовність до змін важливіше проходження попереднім планом

Тобто, не заперечуючи важливості того, що справа, ми все-таки більше цінуємо те, що зліва.

Розглянемо декілька підходів та фреймворків Agile:

- Waterfall (водоспадна модель)
- Scrum
- Kanban
- Lean
- XP (extreme programming)

2.1.1 Waterfall

Модель заснована на послідовному процесі планування, який передбачає поділ етапів роботи на менші завдання. На початку проекту збираються вимоги та визначаються підходи, рішення та етапи реалізації відповідно до методології. Модель водоспаду, яка є однією з найстаріших методологій управління проектами, була представлена Вінстоном У. Ройсом у 1970 році.

Вона була створена як відповідь на швидке зростання галузі розробки програмного забезпечення. Сьогодні вона вважається традиційною та широко

використовуваною моделлю, і часто є відправною точкою для тих, хто вивчає методології управління проектами. Хоча модель водоспаду була основою управління ІТ-проектами протягом багатьох десятиліть, вона також використовується в інших галузях, таких як виробництво та будівництво.

Стадії

Модель водоспаду виконує послідовний процес для завершення роботи. Перш ніж переходити до наступного, потрібно пройти кожен етап. У проектах розробки програмного забезпечення цей процес можна розділити на різні етапи, як показано на малюнку 2.2.

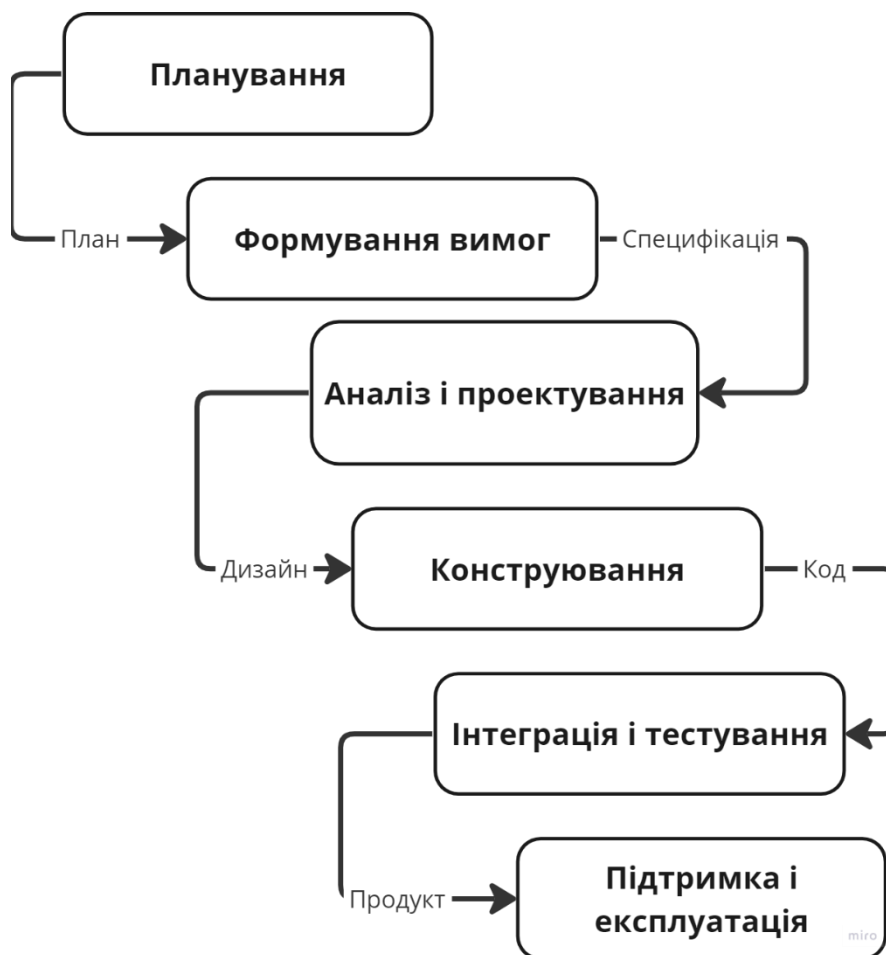


Рисунок 2.2. Етапи розробки ПЗ за допомогою каскадного підходу

Каскадний підхід передбачає створення контрольного списку цілей і забезпечення їх послідовного досягнення. Ця методологія особливо ефективна в галузях, де потрібні ретельні та чіткі вказівки.

Переваги та недоліки

Модель водоспаду — це простий для розуміння спосіб керування проектами, який можна використовувати знову і знову з деякими невеликими змінами. За допомогою цього методу ви повинні переконатися, що відповідаєте всім вимогам, перш ніж переходити до наступного етапу.

Ви також повинні записувати все, що ви робите, щоб, якщо хтось інший буде працювати над проектом, він міг легко зрозуміти, що вже зроблено. Це полегшує новим співробітникам без проблем розпочати роботу над проектом.

Також, цей метод допомагає вам відстежувати все, що відбувається на кожному етапі проекту.

Специфікою цієї методології відіграє важливу роль – етапи, коли вони встановлені, вони не можуть бути змінені. Ця негнучкість накладає серйозні обмеження і робить метод «впертим».

Якщо обсяг або масштаб проекту змінюється, план не може відповідним чином адаптуватися. Ця відсутність адаптивності та креативності змусила деяких критиків вважати модель застарілою. Крім того, метод не враховує важливість зворотного зв'язку від зацікавлених сторін, який може покращити результати проекту.

Незважаючи на деякі недоліки, основною перевагою методології є підхід до планування проекту, який можна застосувати до будь-якого складного процесу, такого як промислове виробництво, будівництво та розробка продукту. Модель водоспаду також полегшує повторення подібних проектів, наприклад будівництво будівлі в іншому місці. Ця методологія особливо корисна для проектів зі стислими термінами та тих, які мають низький рівень ризиків і помилок і також мають чітко прописане технічне завдання від клієнта, що буває доволі не часто.

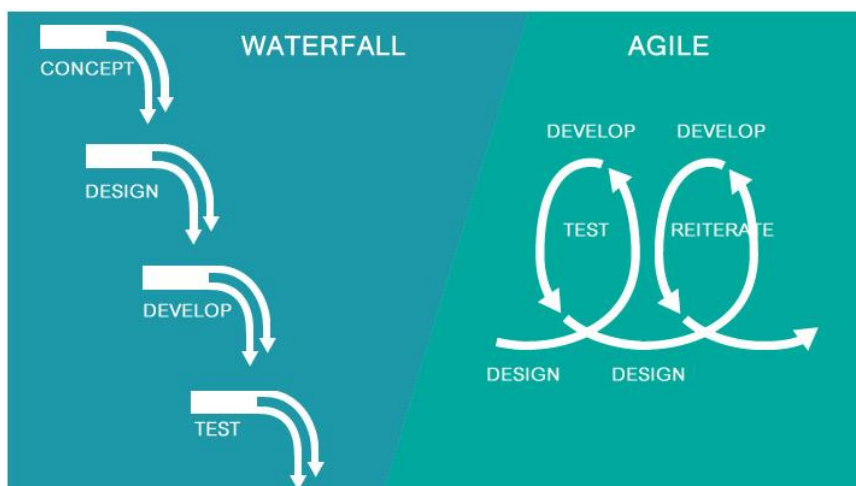


Рисунок 2.3. Схематична різниця каскадної моделі від гнучкого підходу до розробки ПЗ

2.1.2 Scrum

Scrum часто вважається найпопулярнішою та простою гнучкою методологією. Її основа — серія спринтів, яка може тривати від двох тижнів до місяця. Цей прийом допомагає віддавати пріоритет завданням і робить акцент на лідерство команди. Під час спринтів члени команди зустрічаються з 15-хвилинними інтервалами, щоб перевірити прогрес. Scrum — це поширена сучасна техніка, яка використовується в Agile-розробці програмного забезпечення, яка дотримується ітераційного та поступового підходу до управління проектами.

Використання визначених і повторюваних процесів є ефективним лише для вирішення виявлених і повторюваних проблем із визначеними та повторюваними проблемами в певних і повторюваних середовищах. Однак це нереально, оскільки більшість проектів стикаються з непередбачуваними проблемами. Щоб вирішити цю проблему, проекти поділяються на ітерації, також відомі як спринти. У цьому підході проекти розробляються поступово через певний процес, який складається із серії спринтів тривалістю 3-4 тижні.

Кожен новий спринт починається із зустрічі для визначення пріоритетів потреб користувачів. Під час спринту команда зосереджується на невеликій кількості вибраних функцій і впроваджує їх від стадії ідеї до закодованої та

перевіреної функціональності. Для забезпечення ефективної комунікації та співпраці команда Scrum проводить короткі щоденні зустрічі, на яких члени команди відповідають на три запитання: що вони робили вчора, що планують робити сьогодні та з якими перешкодами вони зіткнулися. Нарешті, наприкінці кожного спринту команда обмірковує свій прогрес і визначає області для покращення на ретроспективній зустрічі.

У проектах розробки програмного забезпечення на основі Scrum існують ключові ролі, такі як:

- Scrum – майстер: це цінний член команди, який гарантує, що ітерації Scrum ефективно використовуються для виконання процесів на найвищому рівні. На відміну від типового менеджера проекту, Scrum майстер не контролює щоденне управління командою та не призначає завдання окремим особам. Натомість їх роль більше зосереджена на сприянні успішній комунікації та співпраці між членами команди, забезпеченні усунення перешкод і дотримання командою принципів і практик Scrum. Scrum майстер, по суті, є слугою-лідером, який допомагає команді бути більш само організованою та наділеною повноваженнями, що зрештою призводить до кращих результатів та підвищення продуктивності.

- Product owner (власник продукту) – головна ланка представництва від клієнта. Знається на бізнес цілях продукту, розуміє що потрібно клієнтській базі і скеровує команду для створення якісного продукту.

- Scrum team (команда): За відсутності призначеної особи, яка приймає рішення для призначення завдань, група повинна покладатися на самоорганізацію. Команда в цілому приймає рішення щодо розподілу завдань і обов'язків на зборах. Очікується, що кожен член команди Scrum несе свій внесок у процес розробки від концепції до реалізації.

Щоб пояснити, як пов'язані ці ролі, подумайте про високопродуктивний спортивний автомобіль. Команда, як машина, здатна рухатися в будь-якому напрямку. Власник продукту, як водій, відповідає за те, щоб скеровувати команду в правильному напрямку. Scrum-майстер виконує роль головного механіка, який

стежить за тим, щоб команда була в хорошому стані. З точки зору артефактів, кожен проект Scrum включає три основних компонента:

- Product backlog – беклог це список задач і вимог, які потрібно враховувати при створенні фінального продукту. Це відповідальність власника продукту створити і керувати цим беклогом
- Sprint backlog – це похідна від продуктового беклогу, яка призначена для визначеного спринту.
- Project increment – це результат роботи команди після завершення спринту.

Scrum — це гнучка структура, яка надає пріоритет передбачуваності та управлінню ризиками за допомогою ітераційного та поступового підходу. Методологія складається з чотирьох заходів, які дозволяють тестувати та адаптувати під час спринту, головної події. Ефективність цих заходів базується на трьох фундаментальних стовпах Scrum, які віддають перевагу досвіду та експерименту.

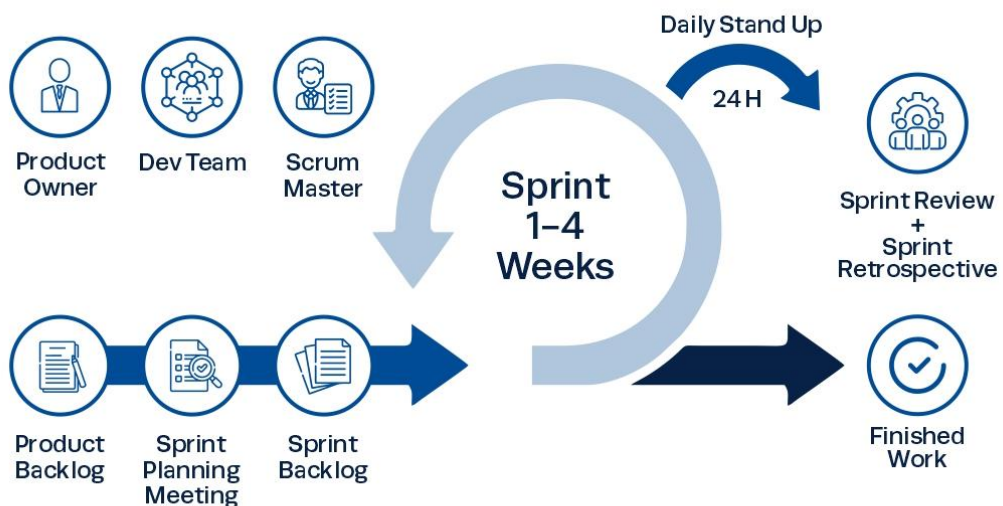


Рисунок 2.4. Схематичне відображення структури Scrum фреймворку

Прозорість має вирішальне значення в процесі роботи. Це важливо не лише для тих, хто бере участь у роботі, а й для тих, хто отримає кінцевий продукт. Приховані або незрозумілі артефакти можуть призвести до рішень, які можуть знизити загальну цінність роботи та підвищити ризик. Прозорість дозволяє перевіряти та допомагає уникнути оманливих тлумачень.

Необхідно проводити регулярні та комплексні перевірки для оцінки артефактів і прогресу, досягнутого на шляху до раніше встановлених цілей, щоб розпізнати та вирішити будь-які потенційні проблеми. Цей процес здійснюється через п'ять різних подій. Після перевірки можна винести відповідні пристосування. Важливо проводити перевірки перед внесенням будь-яких змін, оскільки метою цих заходів є ініціювання змін.

Адаптація є вирішальним кроком у будь-якому процесі, коли певні його аспекти виходять за допустимі межі або якщо кінцевий результат вважається незадовільним. Вкрай важливо якомога швидше ввести необхідні зміни в процес або матеріали, що використовуються, щоб пом'якшити будь-які подальші відхилення від очікуваного результату.

Цей ітеративний підхід гарантує, що проекти адаптуються до мінливих обставин і забезпечують високоякісні результати. Наприкінці кожної ітерації проводиться спринт-перевірка, щоб продемонструвати нові функціональні можливості, і після прийняття всіх розроблених функцій вони інтегруються в програмне забезпечення. Зустрічі, такі як встановлення пріоритетів завдань, щоденні зустрічі команди та огляди спринтів, забезпечують зворотний зв'язок, який може призвести до змін у функціональності продукту або додавання нових елементів до резерву продукту.

Наприкінці кожного спринту власник програмного забезпечення переглядає інкремент та оцінює, чи були досягнуті цілі, поставлені на початку спринту. Цей етап також є можливістю для навчання та вдосконалення з метою вдосконалення робочих практик і забезпечення готовності продукту до тестування. Зрештою, процес розроблено таким чином, щоб він був повторюваним і швидко реагував, що дозволяє безперервно вдосконалюватись і адаптуватися в міру розвитку проекту.

2.1.3 Kanban

Розробка програмного забезпечення не є виробництвом або виробничою діяльністю. Інженери-програмісти щоразу створюють різні речі, тоді як виробництво виробляє ті ж речі знову і знову.

Формальними початками залучення канбану до розробки програмного забезпечення є пов'язаний з Девідом Дж. Андерсоном (Anderson, 2010), коли його запросила компанія Microsoft допомогти в одній зі своїх невеликих команд для досягнення кращої візуалізації робочого потоку і обмеження робочого процесу. Результат виявився доволі успішно.

Починається процес успішного усиновлення від прийняття основних засад:

- Почніть з того, що у вас є зараз - це ваш поточний процес.
- Погодьтеся проводити еволюційний підхід до змін і вдосконалення
- Поважайте поточні ролі та обов'язки команди/організації

Виходячи з цих принципів, наступною річчю для реалізації є п'ять основних канбан принципів, які зазвичай спостерігалися в організаціях, які мав успіх у використанні методу Канбана.

1. Візуалізувати роботу і робочий процес, який він слідує
2. Обмеження Work-In-Progress (WIP) за допомогою віртуальної системи Kanban
3. Керування потоком
4. Зробити політику управління явним
5. Використовуйте моделі та науковий метод та вдосконалюйте спільно

Термін «віртуальна плата Канбана» був використаний Андерсоном, щоб зробити Work-In-Progress видимим, виявляючи обмеження та обмежуючи Work-In-Progress. один пункт. У традиційній розробці програмного забезпечення робота була «підштовхнута» через ліній розробки програмного забезпечення, в той час як

тут кожен має тільки одне завдання одночасно і роботу «тягнуть» після її закінчення.

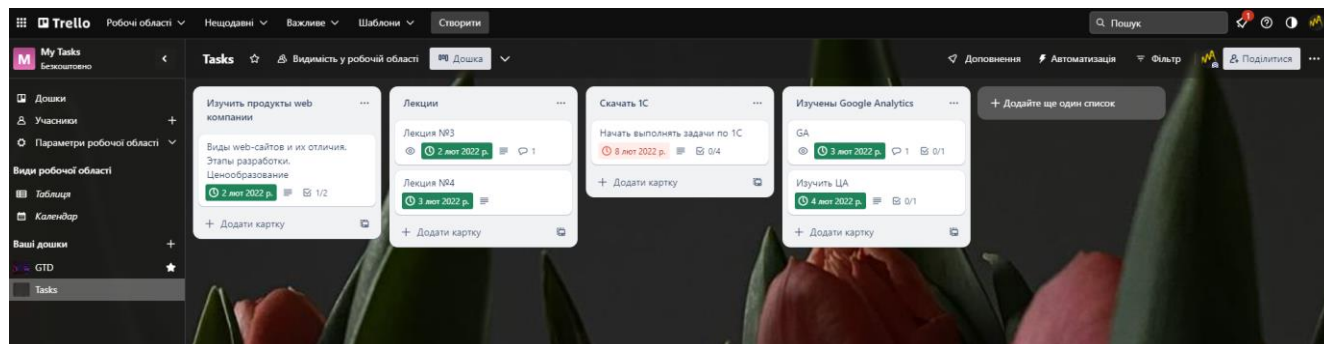


Рисунок 2.5. Приклад канбан дошки в застосунку Trello

Покращення командних комунікацій та потоків розвитку, скорочення часу на вихід на ринок, підвищення продуктивності і створення прозорості в організації як основні фактори мотивації для прийняття Канбана.

Проведено емпіричне дослідження поточного стану використання Канбана в програмному забезпеченні компанії, які широко використовують підходи Agile та Lean. Автори використовували опитування та тематичні інтерв'ю для аналізу мотивів використання канбану, вигоди, отримані від нього і зіткнулися з проблемами. Найпоширеніші досягнуті переваги використання Kanban було кращою видимістю роботи, поліпшенням прозорості та спілкування і кращий контроль потоку. Поряд з перевагами, кілька проблем зіткнулися між собою. Найпоширенішим був брак досвіду роботи з Канбаном. Це означає труднощі в управлінні обмеженнями WIP і пріоритетності завдань.

Kanban набув значної популярності як методологія розробки програмного забезпечення. Це пов'язано з багатьма перевагами, які він надає порівняно з традиційними важкими методологіями.

Однак, важливо зазначити, що не існує універсального підходу до розробки програмного забезпечення, і для кожного проекту потрібна власна унікальна методологія.

Хоча всі гнучкі методології мають свої плюси та мінуси, Kanban виділяється тим, що надає конкретні показники для моніторингу та навчання в рамках стохастичного плану. Це дозволяє точніше прогнозувати ефективність процесу розробки та покращувати командну співпрацю. Загалом Kanban пропонує ключові відмінності порівняно з іншими гнучкими методологіями, які вносять лише незначні зміни до поточного процесу.

Незважаючи на відомі переваги та позитивні результати опитувань, все ще були люди, які не погоджувалися з впровадженням Kanban у повсякденні процеси управління життям. Особливо це стосується сфери розробки програмного забезпечення, де Kanban все ще є відносно новою та інноваційною концепцією. Тому важливо не лише покращити наше розуміння того, як працює Kanban, але й кинути виклик традиційним методам, щоб сприяти швидшому прийняттю.

2.1.4 Lean

Методологія Lean - це набір бізнес-принципів, які визначають пріоритетність постійного вдосконалення, усунення відходів та поваги до людей.

Два загальні стовпи lean мислення - це постійне вдосконалення та повага до людей. На практиці методологія lean спрямована на створення якомога більшої цінності для клієнта, одночасно сприяючи ефективному робочому місцю, яке цінує повагу та командну роботу. Lean виробництво забезпечило оригінальну основу для застосування бережливих практик. Сьогодні методологія lean застосовується в багатьох областях, від охорони здоров'я до розробки програмного забезпечення.

П'ять основних елементів, які служать керівними принципами для створення економної системи, походять від принципів постійного вдосконалення та поваги до людей.

1. Оцінка: Основою методології економії є усунення відходів із робочого процесу. Іншими словами, видаліть усе, що не додає цінності робочому процесу. Будь-яка неефективність вважається марнотратством. Цінним вважається все, що є корисним для клієнта. Це включає все, що безпосередньо задовольняє потреби

клієнта, а також дії, спрямовані на додавання цінності для клієнта (наприклад, забезпечення якості, обслуговування клієнтів тощо).

2. *Відображення потоку створення цінності*: після того, як ви визначили цінність, яку створюють члени вашої команди, наступним кроком стане візуалізація потоку створення цінності для ваших клієнтів за допомогою інструменту економічного управління. Поширеним інструментом економічного менеджменту є дошка Kanban. Дошки Kanban візуалізують весь життєвий цикл продукту чи проекту.

3. *Ефективний робочий процес*. Впровадження економної методології передбачає створення плавного робочого процесу. Це допомагає зменшити різного роду відходи, щоб вчасно доставити товар клієнту. Налаштовуючи ефективний робочий процес, пам'ятайте про вузькі місця, які заважають процесу розробки продукту.

4. *Система тяги*: за допомогою системи тяги нова робота створюється лише тоді, коли є попит з боку клієнта. Замість того, щоб розподіляти нову роботу між членами команди, які активно працюють над завданнями, система витягування ставить завдання в чергу, доки члени команди не будуть готові до додаткової роботи. Це економічний спосіб зменшити перевиробництво та скоротити час циклу — час, потрібний членам команди для виконання завдання.

5. *Удосконалення*. Серед усіх концепцій ощадливого виробництва постійне вдосконалення має першорядне значення. Економні організації зосереджені на тому, щоб усе, що вони роблять, створювало більшу цінність для клієнта. Це означає вимірювання продуктивності, прагнення до досконалості та залучення команд на всіх рівнях до циклу вдосконалення. Існує низка методів управління проектами, які дозволяють постійно оцінювати процеси, щоб зменшити неефективність і надмірну обробку. Ці методи включають Lean Six Sigma, аналіз першопричини та цикл PDCA (план, виконання, перевірка, дія).

Додаткові випадки впровадження lean виробництва призвели до непередбачуваних результатів. Наприклад, компанія Nike не тільки мінімізувала

втрати, але й стала свідком покращення охорони праці та вдосконалення робочих процедур після прийняття методології економії.

Згідно з іншими дослідженнями, економічне управління може зменшити «серйозні» та «критичні» порушення трудових норм майже на 15% у певних регіонах глобального ланцюга поставок. Це вказує на те, що lean управління надає більший пріоритет безпеці та добробуту працівників, ніж деякі звичайні підходи до управління. Зрештою, цей принцип приносить переваги всій компанії.

Постійне вдосконалення може покращити ефективність комерційної авіації, яка стикається з частими простоями та складними процедурами. McKinsey припускає, що впровадження методів економічного виробництва може допомогти авіакомпаніям скоротити час польотів і покращити обслуговування на 30-50 відсотків.

В авіації, де оптимізація процесів може підвищити задоволеність клієнтів і оптимізувати роботу, економна практика має вирішальне значення.

Зробимо підсумок, що Lean та Agile є одночасно методологіями безперервного вдосконалення, але мають різні фокуси та підходи.

Lean фокусується на мінімізації відходів та максимізації цінності для клієнта, систематично усуваючи діяльність, що не додає цінності.

І навпаки, Agile - це гнучкий та ітеративний підхід до розробки програмного забезпечення, який надає пріоритет співпраці з клієнтами та адаптивному плануванню.

Lean і Agile можуть бути комплементарними і часто використовуються разом в організаціях для постійного вдосконалення та досягнення операційної досконалості.

2.1.5 Extreme Programming

Принципи методу екстремального програмування

Цінності важливі, але вони розпливчасті, в тому сенсі, що може бути неможливо вирішити, чи є щось цінним. Наприклад, щось просте з чиеїсь точки зору може бути складним з чужої точки зору.

Отже, в Extreme Programming основні принципи походять від цінностей, щоб практики розробки могли бути перевірені на основі цих принципів. Кожен принцип втілює цінності і є більш конкретним, тобто швидкий зворотний зв'язок – вами або, є це, або ви не.

Основними принципами екстремального програмування є:

- Швидкий зворотній зв'язок:

Концепція швидкого зворотного зв'язку передбачає процес отримання зворотного зв'язку, розуміння його значення та оперативного надання системі отриманих з нього знань. Мета полягає в тому, щоб сприяти швидкій та ефективній реакції на отримані відгуки.

Розробники створюють, впроваджують і оцінюють систему, а також швидко інтегрують отриману інформацію, не чекаючи тривалих періодів часу, таких як дні, тижні або місяці.

Протягом кількох днів або тижнів клієнти оцінюють корисність системи та пропонують відгуки, замість того, щоб чекати місяцями чи навіть роками, щоб висловити свою думку. Цей швидкий процес перегляду дозволяє швидко вдосконалювати та коригувати систему на основі вхідних даних клієнта. Здатність системи своєчасно збирати відгуки клієнтів і реагувати на них зрештою призводить до більшої задоволеності клієнтів і ефективнішого загального продукту.

- Припускається простота

Припускати простоту - це розглядати кожну проблему так, ніби її можна вирішити з простотою.

Традиційно вам кажуть планувати на майбутнє, проектувати для повторного використання. Результат такого підходу може перетворитися на «те, що сьогодні

вимагається замовником, не виконується, а те, що в кінцевому підсумку поставляється, може бути застарілим і важко змінити».

"Припускати простоту" означає "добре виконати сьогоднішню роботу і довіряти своїй здатності додати складності в майбутньому, коли це буде потрібно". В екстремальному програмуванні вам кажуть робити хорошу роботу (тести, рефакторинг та комунікації), зосереджуючись на тому, що важливо сьогодні.

- Додаткові зміни

У будь-якій ситуації великі зміни, зроблені відразу, просто не працюють. Будь-яка проблема вирішується серією найменших змін, які мають значення.

В екстремальному програмуванні інкрементні зміни застосовуються багатьма способами.

- Дизайн змінюється потроху за раз.
- План змінюється потроху за раз.
- Потроху змінюється команда.

Навіть прийняття екстремального програмування має відбуватися маленькими кроками.

- Охоплення змін

Найкраща стратегія - це та, яка зберігає найбільше варіантів, водночас вирішуючи вашу пріоритетну проблему.

- Якісна робота

Серед людей є загальним бажанням добре працювати та отримувати відчуття задоволення від якості роботи, яку вони виконують. Це почуття досягнення — це те, до чого люди прагнуть і чим пишаються. Робота в команді може ще більше підсилити прагнення до якісної роботи, оскільки люди можуть працювати разом, щоб виявити найкраще в один одному та досягти більшого успіху. Зрештою, прагнення до успіху та прагнення створювати якісну роботу є універсальною людською рисою, яка глибоко вкорінена в нашій природі.

Діє ефективно та результативно. Демонструє високий рівень продуктивності та дає задовільні результати. Оптимально працює та успішно виконує поставлені завдання.

Отримує задоволення від трудової діяльності. Знаходить задоволення у виконанні службових завдань. Знаходить самореалізацію у виконанні обов'язків і відповідальності, пов'язаних зі своєю професією. Отримує задоволення від роботи та виконання завдань. Відчуває радість і задоволення від продуктивності та внеску в успіх своєї роботи. Має позитивне ставлення до роботи та бачить її як джерело особистої реалізації та зростання. Знаходить радість у роботі над новими проектами та навчанні нових навичок. Витягує відчуття мети та досягнення від своєї роботи.

При парному програмуванні весь код пишеться двома розробниками за одним комп'ютером, з однією клавіатурою і однією мишею.

У кожній парі є дві ролі - перший розробник (той, що з клавіатурою і мишею) думає про те, як найкраще реалізувати цей метод саме тут. Інший розробник мислить більш стратегічно ставлячи собі питання:

Чи буде весь цей підхід працювати?

Які ще тестові кейси можуть не спрацювати?

Чи є спосіб спростити всю систему так, щоб поточна проблема просто зникла?

Спарювання є динамічним. Це означає, що дві ролі А і Б можуть мінятися місцями або об'єднуватися в пари з іншими членами команди. Найчастіше партнером виступає будь-хто з команди. Наприклад, якщо ви відповідаєте за завдання в незнайомій для вас сфері, ви можете попросити когось із недавнього досвіду працювати з вами в парі.

Перевагами парного програмування є дві голови краще, які точно краще ніж одна. Та фокусування яке можуть досягти двоє розробників при парному програмуванні.

Extreme Programming

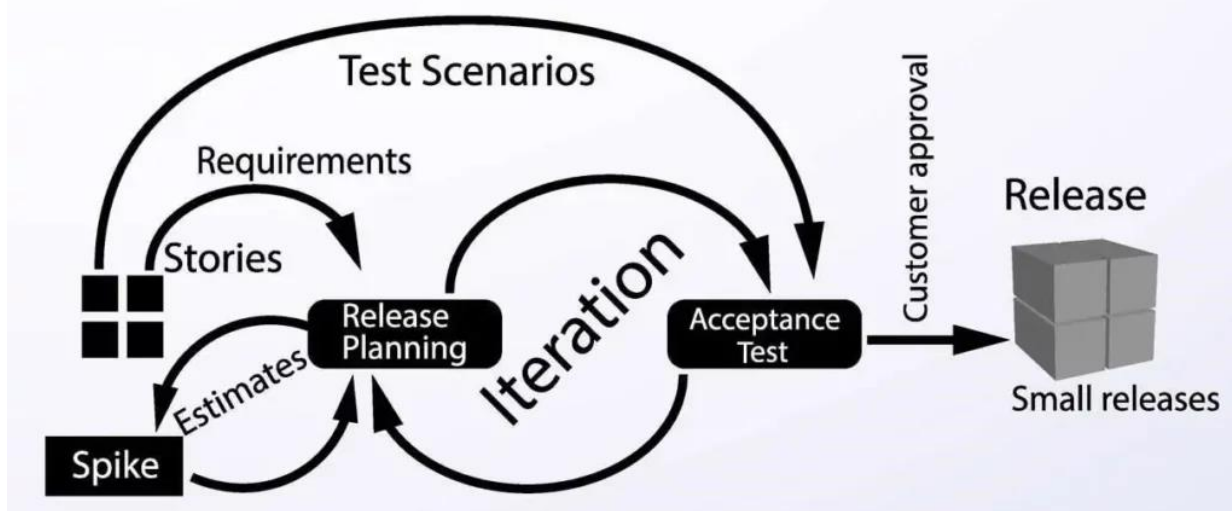


Рисунок 2.6. Схематичне відображення процесу розробки ПЗ за методологією XP

Висновки до розділу 2

Методологія водоспаду — це традиційний підхід до управління проектами, який сьогодні широко використовується на робочому місці. Він названий на честь підходу «зверху вниз», що означає, що кожен етап проекту завершується перед переходом до наступного.

Цей метод є найефективнішим у створенні бетонних кінцевих продуктів, таких як будівельні проекти. Успіх проекту водоспаду залежить від обсягу виконаної попередньої роботи, особливо дослідження, яке дозволяє точніше оцінити завершення проекту.

Однак, негнучкість методу водоспаду ускладнює реагування на невизначеність і зміни. Як наслідок, більш гнучкі методи, такі як Agile, стають все більш популярними. Це дозволяє командам легше адаптуватися та змінювати напрямок.

Отже, у висновку я хочу виділити основні переваги Agile методологій і недоліки.

Переваги Agile методологій

Переваги Agile-методології закладені в її 12 принципах, визначених Agile Alliance:

1. Найвищий пріоритет - задовольнити клієнта шляхом своєчасного та безперервного постачання цінного програмного забезпечення. Задоволення потреб клієнтів та якісні результати - в центрі уваги.
2. Зміну вимог вітається, навіть на пізніх етапах розробки. Гнучкі процеси використовують зміни для конкурентної переваги клієнта. Не боріться зі змінами, натомість навчіться використовувати їх переваги.
3. Надавати працююче програмне забезпечення, від кількох тижнів до кількох місяців, надаючи перевагу коротшим строкам. Постійні результати протягом усього проекту, а не лише в його кульмінаційний момент.
4. Бізнесмени та розробники повинні працювати разом щодня протягом усього проекту. Співпраця є ключовим фактором.
5. Створювати проекти навколо мотивованих людей. Створіть для них необхідне середовище та підтримку і довіряйте їм у виконанні роботи. Залучайте до команди талановитих і працьовитих людей і не заважайте їм.
6. Найефективніший метод передачі інформації команді розробників та всередині неї - це розмова віч-на-віч. Усуньте якомога більше можливостей для непорозумінь.
7. Працююче програмне забезпечення - це головне мірило прогресу. Воно не повинно бути ідеальним, воно повинно працювати.
8. Гнучкі процеси сприяють сталому розвитку. Спонсори, розробники та користувачі повинні мати можливість підтримувати постійний темп нескінченно довго. Повільно і впевнено виграє гонку.
9. Постійна увага до технічної досконалості та гарного дизайну підвищує спритність. Не забувайте звертати увагу на дрібниці.
10. Простота - мистецтво максимізації обсягу невиконаної роботи - має важливе значення. Обріжте зайве.

11. Найкращі архітектури, вимоги та дизайн з'являються в командах, що само організуються. У зв'язку з Принципом 5, ви отримаєте найкращу роботу від вашої команди, якщо дозволите їм самим визначити свої ролі.

12. Через певні проміжки часу команда розмірковує над тим, як стати більш ефективною, а потім налаштовує і коригує свою поведінку відповідно до цього. З'ясовуйте та надавайте зворотній зв'язок, сприймайте його та коригуйте, де це необхідно.

Недоліки Agile методологій

Як і кожна методологія управління проектами, Agile не є ідеальною. Якщо Agile не використовується ефективно, її сильні сторони можуть стати падінням команди. Описую деякі з найпоширеніших недоліків з власного досвіду Agile-методологій:

1. Менша передбачуваність. Гнучкість, що лежить в основі методу Agile, також означає набагато нижчий ступінь передбачуваності. Може бути набагато складніше точно оцінити необхідний час або кількісно визначити ресурси і зусилля, необхідні для завершення проекту. Багато команд бояться цієї невизначеності, і цей страх може призвести до розчарування і прийняття неправильних рішень.

2. Більше часу та відданості. Спілкування та співпраця - це чудово, але постійна взаємодія забирає більше часу та енергії у всіх учасників.

3. Вищі вимоги до розробників і клієнтів. Для того, щоб Agile-методологія була ефективною, необхідна прихильність кожного учасника. Будь-хто, хто не бере участі в проекті, може негативно вплинути на його якість.

4. Відсутність необхідної документації. Оскільки завдання часто виконуються якраз вчасно для розробки за гнучким методом, документація, як правило, менш ретельна, що може призвести до непорозумінь і труднощів у подальшому.

5. Проекти легко збиваються зі шляху. Менш структурована природа Agile-методології означає, що проекти можуть легко збитися зі шляху або вийти за межі початкового обсягу проекту.

Agile стає все більш популярним методом керування проектами не лише у сфері розробки програмного забезпечення, а й в інших сферах. Його здатність адаптуватися до змін і гнучкість роблять його ідеальною методологією для сучасного управління проектами.

Хоча використання Agile має потенційні недоліки, за умови належного дотримання командою, яка дотримується своїх принципів, переваги можуть бути численними. Пандемія COVID-19 підкреслила важливість адаптації в умовах невизначеності, і ця невизначеність, ймовірно, триватиме й у майбутньому. Хоча Agile може не підходити для кожного проекту, його принципи можна адаптувати для різноманітних проектів і допомогти командам залишатися гнучкими та готовими до будь-яких викликів.

Щоб прийняти майбутнє за принципами Agile, бажано отримати сертифікат Agile. Це продемонструє потенційним роботодавцям, що ви готові присвятити себе досягненню успіху.

РОЗДІЛ 3. ПОРІВНЯЛЬНА ХАРАКТЕРИСТИКА СУЧАСНИХ МЕНЕДЖМЕНТ СИСТЕМ

3.1 Проектний підхід до керування процесом

Виділимо основні практики проектного керування для впровадження системи керування проектами:

1. Визначити обсяг задач
2. Ефективне управління вимагає участі в конструктивних дискусіях з відповідними сторонами для вирішення організаційних і ділових проблем раціональним і об'єктивним способом. Цей діалог має тривати протягом усього процесу імплементації, щоб забезпечити швидке усунення будь-яких перешкод, що виникають. Важливо усунути будь-які організаційні бар'єри, які можуть виникнути як із внутрішньої структури компанії, так і з боку будь-яких зовнішніх постачальників, залучених до проекту. Зрештою, усі сторони повинні бути повністю узгоджені зі спільною метою досягнення успішного результату проекту.
3. Визначити реальні часові рамки
4. Не рекомендується покладатися лише на наданий графік проекту. Декілька організацій схильні встановлювати нереалістичні цілі для початку своїх проектів, не звертаючи уваги на практичні обмеження та обставини, пов'язані з процесом.
5. Щоб досягти постійного прогресу та успіху в організації, вкрай важливо розпізнавати, синхронізувати та регулярно контролювати робочі процеси. Це передбачає розуміння взаємозв'язку завдань і графіків проекту на етапах планування, щоб забезпечити адекватний розподіл ресурсів. Важливо постійно контролювати ці взаємозалежності протягом усього проекту.
6. Прийнято вважати, що зелений не може бути червоним, але це не так. Насправді ретельний моніторинг і аналіз того, як реалізується проект, можуть виявити тривожні ознаки, навіть якщо всі показники проекту здаються позитивними. Якщо ці показники лише відображають прогрес на минулих етапах

проекту, але не вказують на готовність до майбутніх завдань і заходів, на них не можна покладатися для прогнозування майбутніх результатів.

7. Потрібно керувати очікуваннями всіма стейкхолдерами.

8. Контроль проекту вимагає пошуку балансу між ідеалістичними датами початку та зовнішніми факторами, а також урахування практичних обмежень, таких як наявні ресурси та фінансові обмеження. Дуже важливо встановити досяжні терміни та постійно повідомляти про них команді проекту, щоб переконатися, що вони залишаються зосередженими на великій картині та не загрузнуть у незначних деталях. По суті, керівники проектів повинні знайти баланс між оптимізмом і реалізмом, щоб забезпечити успіх.

9. Потрібно завжди дотримуватися здорового глузду і дивитись на задачі ширше.

Зовнішні експерти можуть надавати оцінки, які є вирішальними для максимізації цінності та успіху впровадження системи управління проектом, уникаючи дорогих невдач. Їхній досвід може надати дорожню карту впровадження та об'єктивну оцінку потенційних організаційних перешкод, допомагаючи всім залученим сторонам зберігати спокій.

Виконавчий або досвідчений експерт із впровадження управління проектами повинен проводити ці оцінки, оскільки вони мають необхідні знання для розпізнавання важко розпізнаних показників, втручання для управління ситуаціями та регулювання очікувань.

Впровадження методології управління проектом, гарантованого виходить за рамки традиційного управління проектами та пропонує рішення для підвищення шансів на успіх.

Ця методологія визначає та вирішує стратегічні, тактичні та нематеріальні проблеми, а також керує людськими факторами до того, як вони стануть некерованими. Довіра, отримана від керівництва проектом, гарантує, що всі залучені сторони знають, що проект на правильному шляху.

3.2 Загальний огляд рішень присутніх на IT ринку

Гнучкий підхід до розробки програмного забезпечення набирає обертів завдяки своїй здатності прискорити завершення проекту. Фактично, приблизно 71% компаній відзначили збільшення успіху проекту на 28%, використовуючи цей метод замість традиційних.

Цей висновок ґрунтується на дослідженні, яке продемонструвало, як гнучкі методології розробки програмного забезпечення можуть знизити витрати, підвищити якість, продуктивність і загальне задоволення від бізнесу. Ключовим фактором впровадження Agile є його здатність швидко та без зусиль адаптуватися до змін в управлінні проектами.

Agile Manifesto базується на ідеї прискорення процесу розробки та забезпечення виробництва високоякісного програмного забезпечення. Для досягнення цього в маніфесті викладено чотири цінності та дванадцять принципів.

Ці принципи включають надання пріоритету особам і взаємодії над процесами та інструментами, зосередження на робочому програмному забезпеченні замість обширної документації, співпрацю з клієнтами замість укладення контрактів і реагування на зміни, а не дотримання жорсткого плану. Різноманітні гнучкі методології, такі як Extreme Programming, Scrum і Kanban, застосовують ці цінності по-різному для забезпечення успішної розробки та доставки програмного забезпечення.

Гнучкі методології все частіше використовуються організаціями, що розробляють програмне забезпечення, щоб підвищити ефективність, швидкість і якість програмного забезпечення.

Ця тенденція спостерігається не тільки в невеликих командах розробників, а й у великих компаніях-розробниках програмного забезпечення. Серед різноманітних гнучких методологій, Scrum є найпоширенішим у понад 70% гнучких команд.

Управління проектом значно впливає на якість кінцевого програмного забезпечення. Отже, багато фахівців із програмного забезпечення визнали переваги

гнучких методологій у прискоренні завершення проекту, тоді як гнучкі інструменти управління проектами сприяють покращенню якості та збільшенню гнучкості проекту.

Опитування, проведене в 2023 році на основі 350 відповідей, від працівників і проджект-менеджерів різних компаній показало, що такі інструменти, як Excel, Microsoft Project, Monday, Atlassian/Jira, Trello, Asana, Gantt Pro, Base Camp, і LeanKit, зазвичай використовуються для гнучкого управління проектами. Оцінки задоволеності виявилися найвищими для Asana та Jira. Інше опитування, проведене в 2022 році, показало, що Trello & Jira найбільш часто використовуваним інструментом. За даними Google Trends, з 2004 по 2018 рік популярність Microsoft Project знизилася на користь Jira та Trello. Існуючі інструменти управління проектами не створювалися для віддаленого розширення клієнтів на місці.

3.3 Jira

Jira — це широко використовуваний інструмент управління проектами серед професіоналів програмного забезпечення та інституційних дизайнерів, які працюють у спільному середовищі. Однією з його найбільш помітних особливостей є можливість налаштовувати дошки Scrum і Kanban, що робить його адаптованим до індивідуальних потреб.

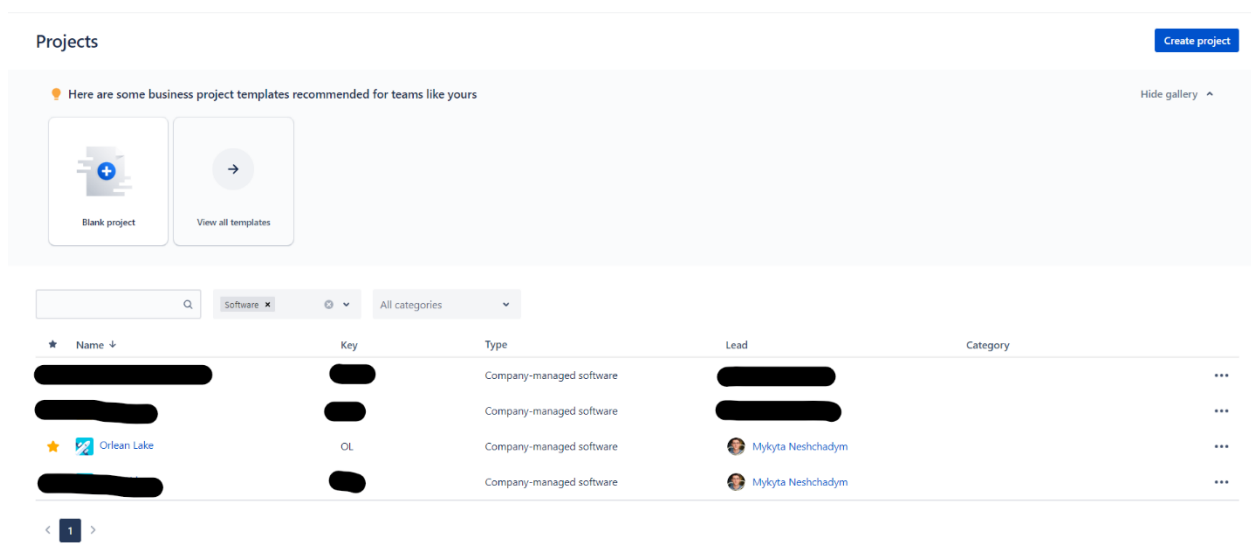


Рисунок 3.1 Початковий інтерфейс вкладки Projects в Jira

Jira також повністю інтегрована з популярними програмними середовищами, такими як Eclipse, і дозволяє використовувати спеціальні фільтри додатків за допомогою мови запитів Jira (JQL). Інші важливі функції включають інтеграцію налаштовуваних інструментів розробника, сукупні блок-схеми, звітність, випуски та відстеження помилок. Як зрілий і перевірений інструмент, якому довіряють великі компанії в усьому світі, Jira має процвітаючу спільноту користувачів.

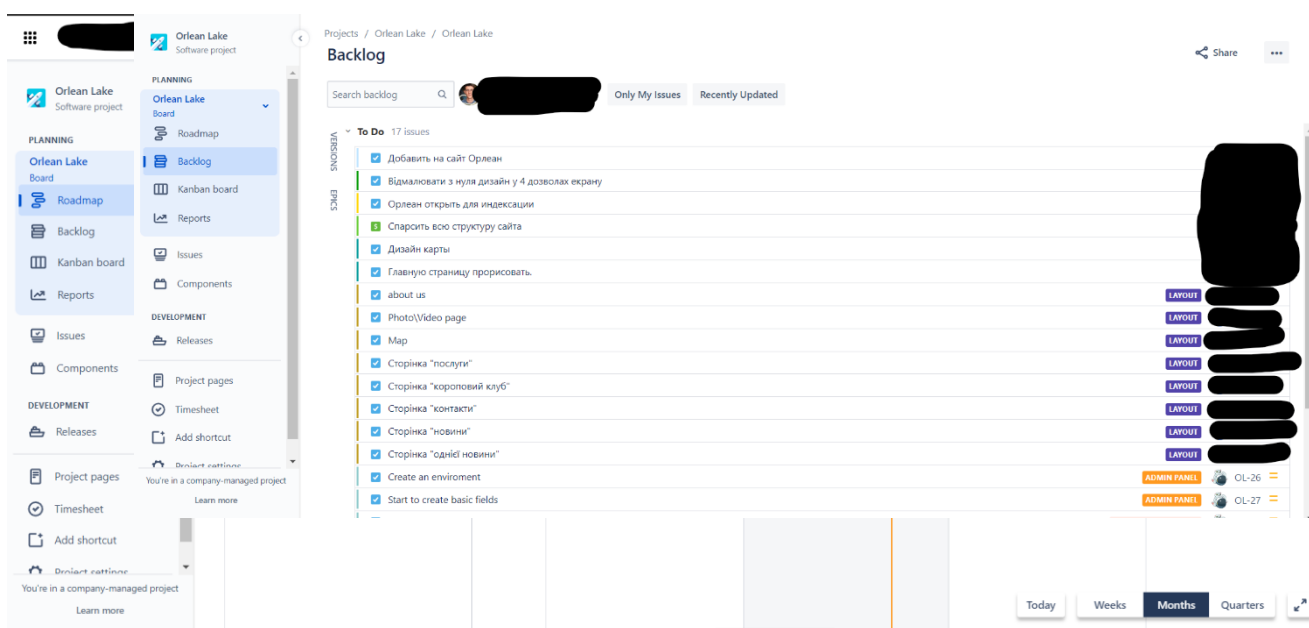


Рисунок 3.2 Загальний вигляд вкладки дорожньої карти на кожному проекті

Те, як працює ця служба, можна порівняти з диспетчером завдань у комп'ютерній системі, оскільки вона пильно стежить за поточними процесами (проектами) і керує розподілом ресурсів (співробітників). Керівник проекту Jira відповідає за розподіл завдань між співробітниками та ефективне планування робочого навантаження. Наприклад, якщо всі розробники вже зайняті чотирма поточними проектами, було б нерозумно починати новий проект, а замість цього було б розумніше дочекатися завершення хоча б одного проекту.

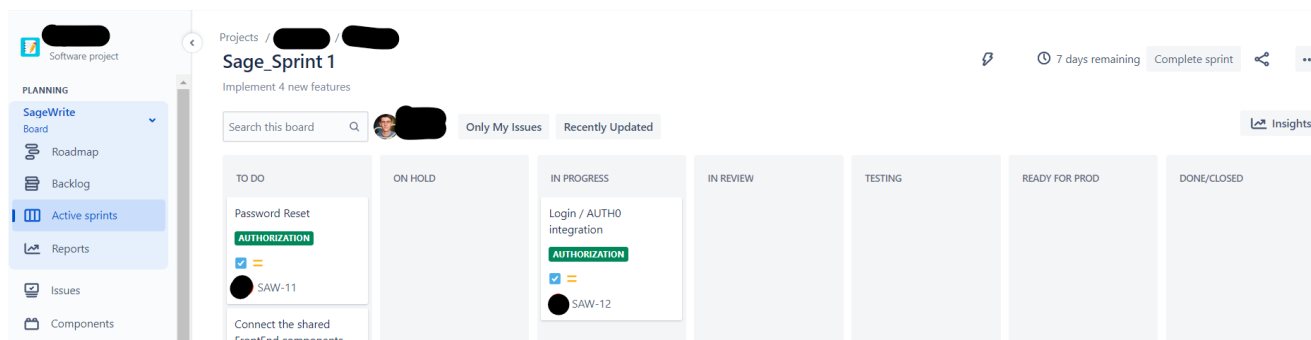


Рисунок 3.3 Беклог проекту в Jira

Create issue

Project *

Orlean Lake (OL)

Issue type *

Task

[Learn more](#)

Status ⓘ

Backlog/Open

This is the issue's initial status upon creation

Summary *

Components

None

Assignee

☐ Create another issue
 Cancel
Create

Рисунок 3.4 Канбан дошка в Jira

Щоб створити задачу в Jira, потрібно натиснути на «створити» в хедері системи. Або потрібно додати нову задачу до беклогу.

Підготовка передбачає створення вичерпного списку завдань, їх пріоритетність, оцінку цілей, термінів виконання та рівня складності.

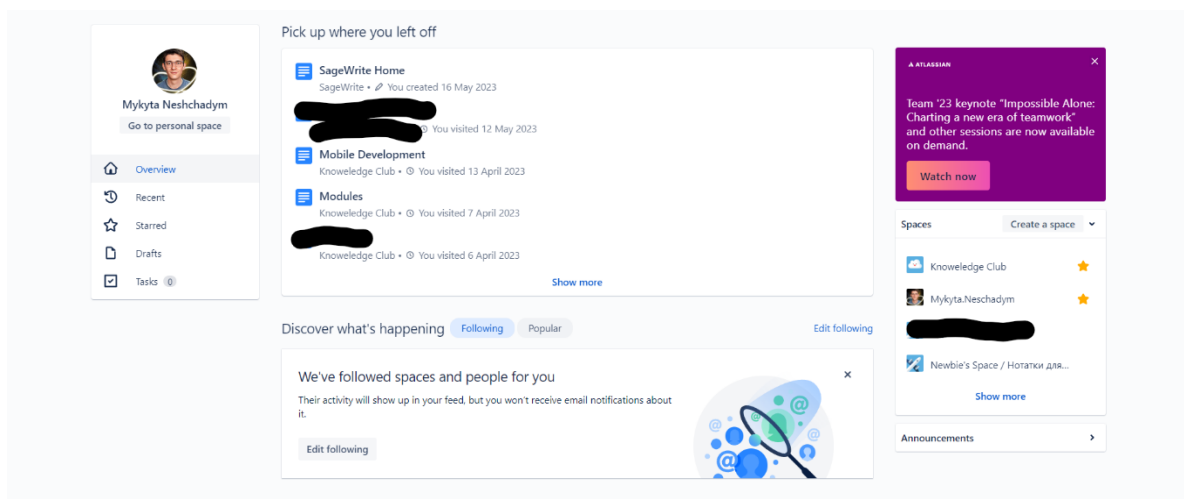


Рисунок 3.5 Картка створення задачі в Jira

Щоб забезпечити ефективне планування, розробники Jira пропонують приділяти мінімум дві години щотижня та використовувати різні методи та позначення для оцінки складності. Одним із таких підходів є використання розмірів одягу для вказівки розміру завдання, де XS означає невеликі та здійсненні завдання, а XXL позначає складні та складні завдання, які неможливо виконати за один спринт.

Зазвичай, використовують такі поняття як Epic і User Story.

Щоб ефективно відстежувати зміни в проекті, важливо вести облік вжитих дій. Цього можна досягти за допомогою журналів, які є файлами, що зберігаються у форматі .log і документують послідовність дій, виконаних у рамках проекту.

Ці журнали зазвичай зберігаються в Git, системі контролю версій. Зберігаючи проект після кожної зміни, команда може легко виявляти та виправляти помилки, а також аналізувати загальну продуктивність продукту.

Jira пропонує різні додаткові функції, які допомагають керувати проектами. Однією з його ключових особливостей є можливість зберігати код та інші ресурси безпосередньо в проекті, забезпечуючи централізоване місце для співпраці під час розробки проекту. Крім того, Jira можна інтегрувати з Confluence, комунікаційною платформою Atlassian і репозиторієм коду, а також Slack. Confluence — це

комплексний інструмент, який дозволяє членам команди спілкуватися, планувати завдання та організовувати зустрічі.

Рисунок 3.6 Головна сторінка профілю в Confluence

Jira — це широко використовуваний інструмент управління проектами, який пропонує безкоштовну версію для команд до 10 учасників. Однак, якщо команда більша, то існує стандартний план на 7 доларів США за користувача, яка надає доступ до 250 ГБ пам'яті та технічну підтримку в робочий час.

Для команд із більш просунутими потребами преміальна ціна в 14 доларів США за користувача пропонує необмежений доступ до репозиторію та цілодобову технічну підтримку.

Обидва ці тарифи дозволяють підключити до 20 000 користувачів і можуть оплачуватися щомісяця або щороку.

Підписка Enterprise є найдорожчим варіантом, і вона доступна лише за річну оплату. Рекомендується для тих, хто має особливі вимоги до безпеки та масштабованості.

3.4 Trello

Trello — це інструмент керування процесами на основі дошки канбан, який спочатку створив Fog Creek. Однак у 2017 році його придбала компанія Atlassian, і тепер і Trello, і Jira є частиною її портфолію. Trello відомий своїм зручним інтерфейсом і рекомендований для всіх типів команд, навіть тих, що мають обмежену кількість учасників.

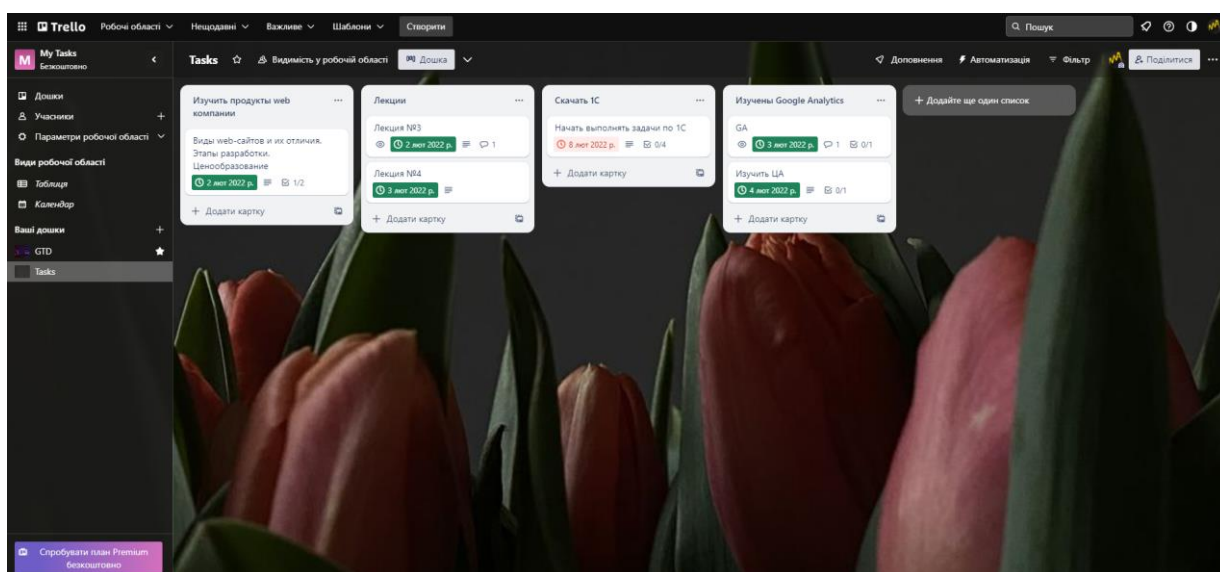


Рисунок 3.7 Інтерфейс дошки в Trello

У Jira організація роботи дотримується методу Канбан (але і Скрам можна налаштувати), де завдання класифікуються на заплановані, незавершені та виконані стовпці на дошках. Кожне завдання представлено карткою з різними атрибутами, такими як назва, опис, виконавець і мітки. Однак картки Trello відрізняються від завдань Jira тим, що вони можуть містити під завдання, списки або контрольні списки.

Картки служать централізованим сховищем для всієї важливої інформації, яка потрібна вашій команді. Ви можете призначати учасників, встановлювати кінцеві терміни, надавати відгуки та здійснювати інші дії. Таким чином ви зможете ефективніше керувати своїм проектом і оптимізувати робочий процес.

Додаючи членів до карток для їхніх конкретних завдань і проектів, можна підтримувати підзвітність і зникати потреба запитувати, хто відповідає за конкретне завдання. Ця функція гарантує, що всі члени команди чітко розуміють свої обов'язки, і допомагає уникнути плутанини чи непорозумінь.

Концепція термінів виконання тісно переплітається з актом виконання завдань. Встановлення їх є простим процесом, але за допомогою нагадувань їх важко забути. Виконання завдань до встановленого терміну - це приємний досвід.

Часи перегляду довгих листів електронної пошти, щоб знайти вкладення, минули. За допомогою нашої функції просто перетягніть необхідні файли на призначену картку, гарантуючи, що відповідні документи завжди пов'язані з відповідними завданнями. Це економить дорогоцінний час і усуває розчарування від пошуку вкладених файлів у захаращеній папці "Вхідні".

Контрольні списки — ефективний спосіб вирішення важких завдань, розділяючи їх на менші завдання, які можна впоратися. Відмічаючи кожен пункт у списку, можна відстежувати прогрес і відчувати досягнення, коли рядок стану досягає 100% завершення.

Також є можливість у Trello перетворити свою дошку на багатофункціональний інструмент, який легко підключається та працює в гармонії з різними програмами та службами, які використовує ваша команда. Це включає такі популярні платформи, як Slack, Outlook, Gmail, Salesforce, InVision, Jira та безліч інших доступних для вас варіантів. Таким чином ви можете підвищити загальну функціональність і ефективність своєї дошки, зробивши її ключовим компонентом робочого процесу вашої команди.

Вставивши Power-Ups у свої дошки Trello, ви можете створювати налаштовані робочі процеси, які перетворюють Trello на універсальний інструмент, який можна використовувати для різних цілей, наприклад для керування взаємовідносинами з клієнтами, підтримки продажу квитків, планування в соціальних мережах і багато іншого.

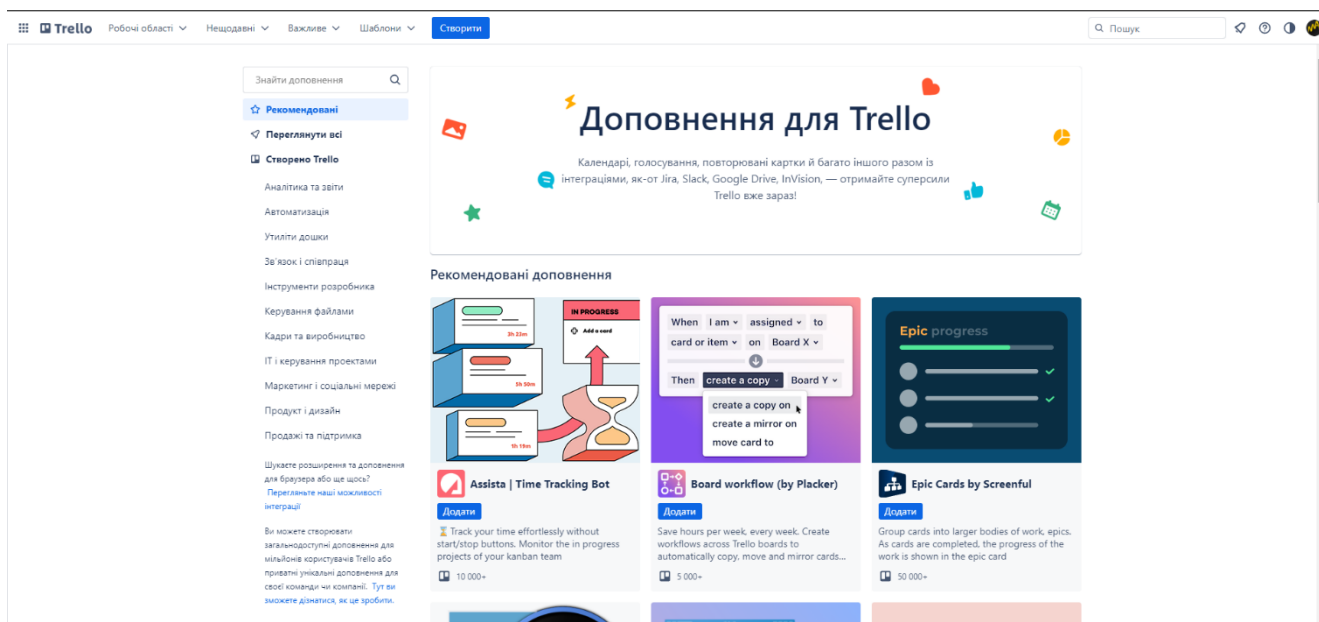


Рисунок 3.8 Меню з різноманітними додатками, які можна інстальювати в Trello

Trello — це універсальний інструмент, який можна використовувати для керування командами будь-якого розміру, від малих організацій до великих корпорацій. Він пропонує ряд функцій, які допоможуть вам структурувати робочий процес і задовольнити унікальні потреби вашої команди.

Користувальницькі поля дозволяють додати процес і формальність до ваших завдань, тоді як вбудована система автоматизації дозволяє встановлювати правила та команди для ефективного просування вашої роботи. Ви також можете легко керувати дозволами користувачів і контролювати вміст у режимі реального часу. Завдяки надійній і безпечній платформі Trello ви можете бути впевнені, що робота вашої команди ніколи не буде перервана.

Нижче, можна побачити прайс-лист від Trello та варіативність підписок на користування їх сервісом.

FREE	STANDARD	PREMIUM	ENTERPRISE
\$0 USD Free for your whole team For individuals or teams looking to organize any project. Get started	\$5 USD Per user/month if billed annually (\$6 billed monthly) For small teams that need to manage work and scale collaboration. Upgrade now	\$10 USD Per user/month if billed annually (\$12.50 billed monthly) For teams that need to track and visualize multiple projects in several ways, including boards, timelines, calendars, etc. Try for free	\$17.50 USD Per user/month - billed annually (\$210.00 annual price per user) For organizations that need to connect work across teams with more security and controls. Est. cost for 50 users  Contact sales
INCLUDED IN FREE: ✓ Unlimited cards ✓ Up to 10 boards per Workspace ✓ Unlimited Power-Ups per board ✓ Unlimited storage (10MB/file) ✓ 250 Workspace command runs per month ✓ Custom backgrounds & stickers	EVERYTHING IN FREE, PLUS: ✓ Unlimited boards ✓ Advanced checklists ✓ Custom Fields ✓ Unlimited storage (250MB/file) ✓ 1,000 Workspace command runs per month ✓ Single board guests ✓ Saved searches	EVERYTHING IN STANDARD, PLUS: ✓ Views: Calendar, Timeline, Table, Dashboard, and Map ✓ Workspace views: Table and Calendar ✓ Unlimited Workspace command runs ✓ Admin and security features ✓ Workspace-level templates ✓ Collections ✓ Observers	EVERYTHING IN PREMIUM, PLUS: ✓ Unlimited Workspaces ✓ Organization-wide permissions ✓ Organization-visible boards ✓ Public board management ✓ Multi-board guests ✓ Attachment permissions ✓ Power-Up administration ✓ Free SSO and user provisioning with Atlassian Access

Рисунок 3.9 Цінова політика Trello на підписку до їх сервісу

Система зручна з візуально привабливим і простим інтерфейсом, якою можуть з легкістю користуватися навіть люди старше 60 років.

Однак це може бути не найкращий варіант для масштабних проектів. Варто зазначити, що версія мобільного додатку Trello дотримується своєї основної концепції. Таким чином, неможливо переглянути будь-який список завдань, і завдання потрібно перетягувати між стовпцями.

Trello має низку переваг, серед яких логічна побудова системи постановки завдань, зручна система карток, базові операції, які вимагають не більше двох

кликів, інтеграція зі сторонніми сервісами та функціональна система кольорової фільтрації.

Однак, також має деякі недоліки, такі як обмежені можливості налаштування, відсутність аналітики, відсутність обліку часу на завдання в базовому функціоналі, інтерфейс, який може швидко захаращуватися, що ускладнює пошук потрібної дошки, і труднощі при переміщенні карток між колонками на телефоні. Крім того, завдання з різних проектів не видно на одному екрані, і він підходить лише для найпростіших лінійних завдань.

3.5 Monday

Monday Work Management — це рішення для управління проектами, яке можна налаштувати відповідно до конкретних потреб бізнесу. Його створено для спрощення операцій, дозволяючи командам відстежувати проекти, переглядати дані, ефективніше співпрацювати та користуватися іншими перевагами. Програмне забезпечення містить ряд функцій, таких як дошки канбан, гнучкі дорожні карти, планування спринтів, таблиці, діаграми Ганта тощо.

Діаграма Ганта — це інструмент планування проекту, який нагадує гістограму та використовується для зображення часової шкали проекту. Його можна порівняти з дошкою канбан для завдань, оскільки він спрощує процес моніторингу просування різних завдань у розкладі проекту. По суті, діаграми Ганта забезпечують візуальне представлення календаря проекту, що полегшує відстеження прогресу різних завдань.

Дошка канбан і діаграма Ганта служать інструментами для відстеження прогресу в проекті. Однак перша показує статус завдань протягом певного часу, тоді як друга показує відсоток виконання кожного окремого завдання. Залежно від проекту та вподобань користувача, одному може бути достатньо дошки канбан, а іншому може знадобитися більш детальна інформація, надана діаграмою Ганта.

Щоб створити діаграму Ганта для управління роботою в Monday, вам знадобиться платний акаунт, оскільки ця функція недоступна у безкоштовній версії. Платні акаунти коштують від \$8.00 за місце на місяць для базового акаунта до \$20

за місце на місяць для професійного акаунта. Однак, лише в планах Standard, Pro та Enterprise доступні діаграми Ганта.

Важливо також зазначити, що в кожному плані є мінімум три місця, тому план Standard коштуватиме вам щонайменше \$8.00 на місяць, який виставляється щорічно. Це означає, що якщо вам потрібні діаграми Ганта, ви заплатите мінімум \$288.0 на рік за три місця. Але якщо ви хочете спробувати, перш ніж купувати, ви можете протестувати план Standard протягом 15 днів.

У цьому аналізі управління роботою в Monday буде розглянуто його переваги, недоліки, унікальні функції та ціни. Крім того, буде надано кілька альтернативних варіантів і пояснимо наш підхід до проведення перевірок.

Завдяки різноманітним параметрам перегляду проекту, таким як шкала часу, діаграма Ганта та дошка канбан, monday.com полегшує візуальну співпрацю. Крім того, користувачі можуть легко персоналізувати свою інформаційну панель і шаблони відповідно до своїх потреб.

Програмне забезпечення також дозволяє імпортувати дані із зовнішніх джерел, таких як Microsoft Excel і Google Sheets. Крім того, пропонується безліч налаштованих засобів автоматизації для оптимізації процесів управління проектами. Щоб випробувати програмне забезпечення, можна або вибрати безкоштовний план, або скористатися 14-денним пробним періодом.

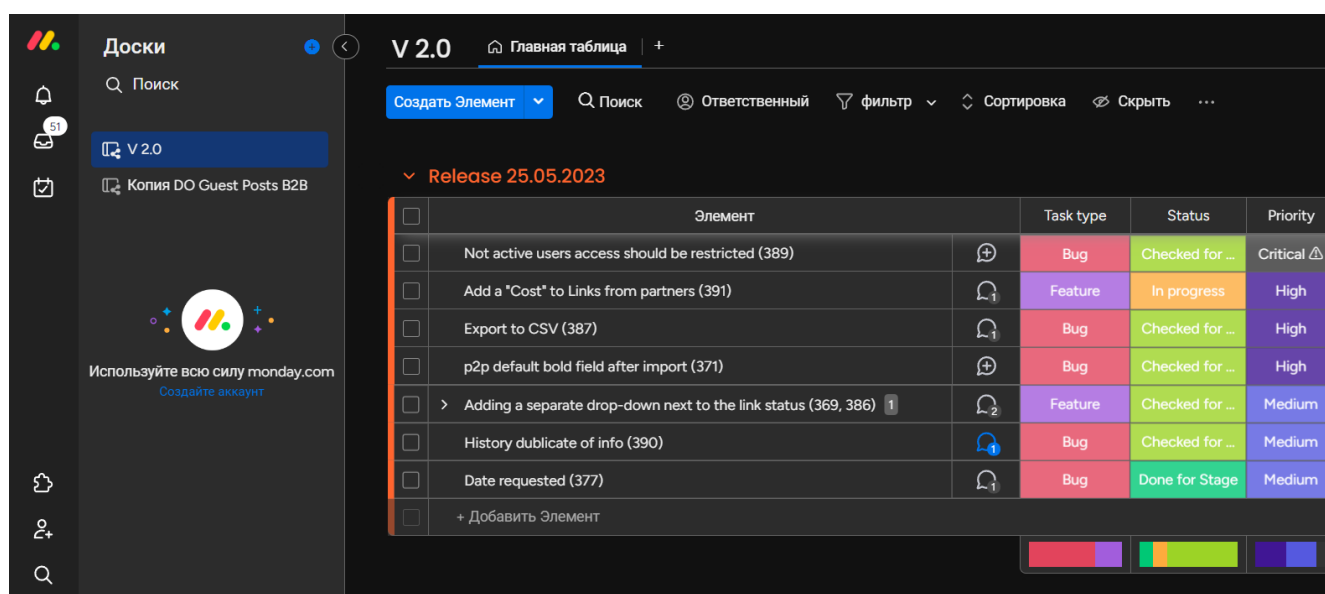


Рисунок 3.10 Інтерфейс робочого простору Monday

Тарифні плани monday.com

Безкоштовний: безкоштовно для двох місць.

Базовий: \$8 за користувача на місяць з річною оплатою, або \$10 за користувача щомісяця.

Стандартний: \$10 за користувача на місяць з річною оплатою або \$12 за користувача щомісяця.

Pro: \$16 за користувача на місяць з річною оплатою або \$20 за користувача з щомісячною оплатою.

Корпоративний: Індивідуальний. З корпоративним планом ви отримаєте індивідуальний вступний курс, преміум-підтримку, розширені функції звітності та багато іншого.

Monday.com пропонує своїм користувачам безліч переваг. Однією з найпомітніших є чистий та інтуїтивно зрозумілий інтерфейс користувача, який не обмежується широкими можливостями платформи. Крім того, платформа пропонує надійні параметри налаштування, що дозволяє користувачам пристосовувати свій досвід до своїх конкретних потреб.

Ще однією великою перевагою використання Monday.com є його потужні засоби автоматизації, які можуть оптимізувати робочі процеси та підвищити ефективність. Нарешті, Monday.com може похвалитися великим списком інтеграцій, що полегшує підключення до інших інструментів і платформ. Усі ці переваги роблять Monday.com конкурентним вибором для окремих осіб і компаній, які шукають потужне рішення для управління проектами.

Також, кілька недоліків використання monday.com, які слід враховувати.

По-перше, більш просунуті функції платформи можуть бути важкими для освоєння та потребують значної кількості часу та зусиль для освоєння.

Крім того, якщо ви зацікавлені у використанні можливостей відстеження часу, вам потрібно буде оновити обліковий запис до Pro плану. Нарешті, варто зазначити, що стандартні налаштування сповіщень на monday.com можуть призвести до надсилання надзвичайної кількості сповіщень, що може бути не бажаним для всіх користувачів.

Monday — має численні функції безпеки та конфіденційності даних. Користувачі можуть бути впевнені, що їхні дані захищені 256-бітним шифруванням і надсилаються через TLS 1.2. Крім того, хмарні дані безпечно розміщуються в численних центрах обробки даних AWS у Сполучених Штатах, щоб забезпечити максимальний захист.

Крім того, для підвищення рівня безпеки кожні п'ять хвилин створюється резервна копія важливої інформації. Крім того, кваліфікований технічний персонал monday.com гарантує, що веб-сайт залишається в робочому стані та функціонує оптимально, незважаючи на поточні заходи з обслуговування, які можуть відбуватися у фоновому режимі.

З довідником про роботу в Monday ви отримаєте універсальність, а це означає, що він підходить для різних типів користувачів. Будь-яка компанія, яка шукає налаштовуване рішення для управління проектами, яке пропонує широкий спектр функцій, може звернути увагу на Monday.

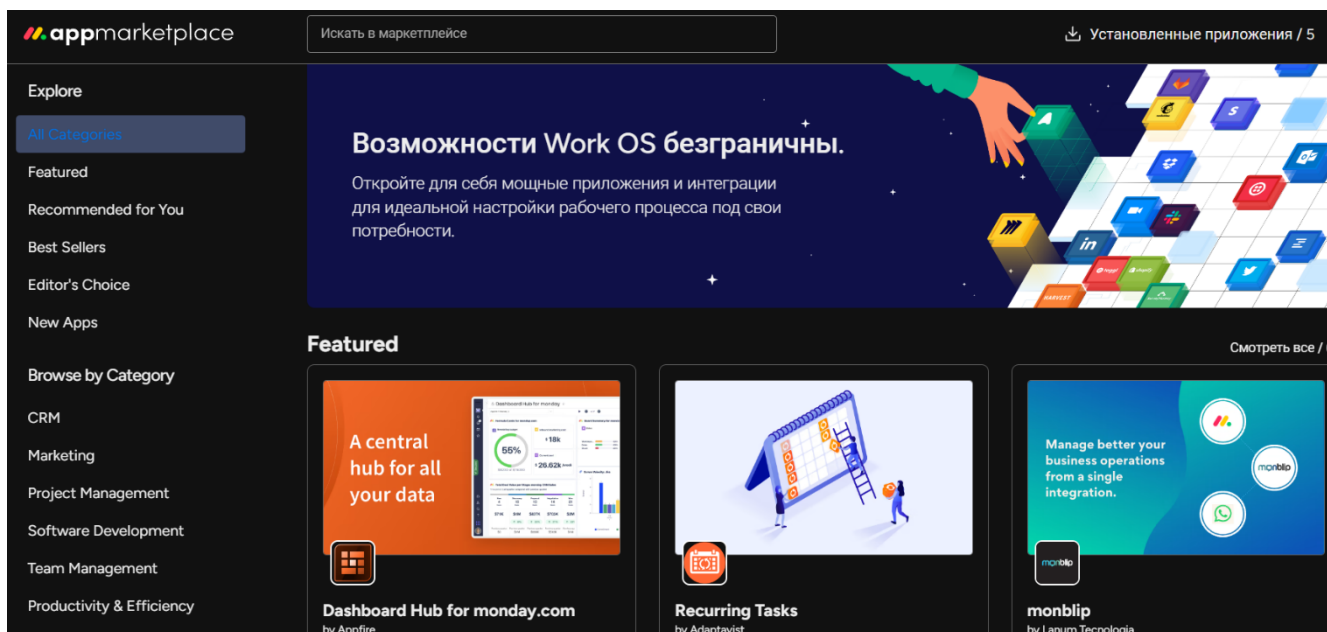


Рисунок 3.11 Інтерфейс панелі з інтеграціями

3.6 Оцінка та аналіз результату порівняльної характеристики продуктів

Щоб програмне забезпечення було корисним для управління проектами, воно має бути зручним для користувача та пропонувати різноманітні корисні функції.

Для цієї мети доступні численні програми, зокрема Trello, Monday, Jira Basecamp та інші. Усі вони постійно вдосконалюються, щоб краще обслуговувати своїх користувачів. Ринок програмного забезпечення повний варіантів, як безкоштовних, так і платних, з різним рівнем доступності інструментів.

Я звузив список найкращих із найкращих, проаналізувавши понад 30 програм, щоб вибрати 12 найбільш функціональних платформ (докладні порівняння деяких з них доступні в попередньому розділі).

Було розглянуто такі програми:

- Gantt Pro
- BaseCamp
- Jira
- Wrike
- Smartsheet
- Ms Project
- Asana
- Odoo
- Trello
- Monday
- Redbooth
- Zoho

Порівняння і аналіз програм робився за наступними параметрами:

- Підтримка ОС
- Обмін файлами
- Візуалізація даних

- Створення звітів
- Система налаштуваних сповіщень
- Прогнозування
- Діаграма Ганта
- Хмарне сховище
- Наявність чату
- Синхронізація зі стороннім ПО
- Online
- Суспільний доступ
- Налаштування фільтрів
- Інтуїтивність інтерфейсу
- Пробний період
- Безкоштовний тариф

Програми керування проектами зазвичай включають діаграму Ганта, яка є гістограмою, яка відображає графік роботи проекту. Ця функція дозволяє відстежувати хід виконання, надаючи інформацію про те, чи проект випереджає чи відстає від графіка, що особливо корисно для великомасштабних проектів зі значними фінансовими наслідками.

У сучасну епоху Інтернет переповнений величезною кількістю інформації, що робить роботу в Інтернеті та хмарне зберігання дуже затребуваною функцією багатьох програм. Ця можливість дозволяє користувачам змінювати проекти, відстежувати їх прогрес з будь-якого місця з доступом до Інтернету та працювати віддалено під час відряджень.

Співпраця над проектами також стає легшою завдяки спільному використанню додатків, усуваючи потребу в передачі даних між комп'ютерами. Відстеження витрат за певний період показує, скільки грошей було інвестовано в проект, залишки коштів і чи працює компанія з прибутком чи збитком. Ця функція також дозволяє автоматично прогнозувати, що зручно для прогнозування майбутніх витрат.

Після пробного періоду користувачі можуть придбати ліцензійну версію програми, вартість якої залежить від розміру та бюджету компанії. Візуальне представлення інформації у вигляді діаграм спрощує її розуміння, особливо для нетехнічних користувачів. Для україномовних споживачів наявність україномовної версії програми є перевагою.

Пропонування пробної або безкоштовної версії програми є вигідним, оскільки дозволяє користувачам експериментувати з нею в освітніх цілях або вибрати програму, яка найбільше відповідає їхнім потребам. Безкоштовна версія особливо вигідна для малого бізнесу.

Можливість спілкуватися в чаті та обмінюватися файлами в програмі є корисною, усуваючи потребу в установці додаткового програмного забезпечення. Також важливо враховувати операційні системи, які підтримує програма, чим більше операційних систем вона підтримує, тим більше користувачів може залучити.

Щоб заощадити час і швидко знайти інформацію, фільтри дисплея є корисним інструментом, який користувачі можуть використовувати.

Синхронізація з іншим програмним забезпеченням забезпечує безперебійну передачу даних з одного пристрою на інший, прискорюючи завершення проекту та гарантуючи, що учасники отримають важливі оновлення на своїх мобільних пристроях.

Створення звітів за будь-який період від початку проекту дозволяє вести звітну документацію та здійснювати моніторинг проекту.

Інтуїтивно зрозумілий інтерфейс є перевагою, що дозволяє нетехнічним користувачам швидко ознайомитися з програмою та почати працювати в ній без тривалого навчання.

Вирішальним фактором, який слід враховувати при виборі програми управління проектами, є її вартість. Існує кілька варіантів для мало бюджетних проектів, таких як RedBooth, Trello, MS Project і Asana. Однак навіть невеликі проекти можуть використовувати безкоштовні програми, такі як GanttPro, BaseCamp, Wrike, Trello, Asana та Odoo. Крім того, більшість програм пропонують

пробний період, що дозволяє користувачам визначити, яка програма найкраще підходить для їхніх конкретних потреб.

Виокремивши, ці критерії для детальної оцінки була розроблена таблиця (Рисунок 3.12) з порівняльною характеристикою. В цій таблиці за кожний пункт, кожна програма отримувала по 1 балу за кожний параметр.

	Gantt Pro	BaseCamp	Jira	Wrike	Smartsheet	MS Project	Asana	Odoo	Trello	Monday	Redbooth	Zoho
	Windows Mac Web	All platforms	All platforms	Android Web	Android Web	Windows Web	Android Web	All platforms	All platforms	All platforms	Android Web	All platforms
Підтримка ОС	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Обмін файлами	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Візуалізація даних	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Створення звітів	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Система налаштовуваних словесень	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Прогнозування	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Діаграма Ганта	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Хмарне сховище	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Наявність чату	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Синхронізація зі стороннім ПО	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Online	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Суспільний доступ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Налаштування фільтрів	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Інтуїтивність інтерфейсу	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Пробний період	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Безкоштовний тариф	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Вартість	20\$	100\$	14\$	10\$	10\$	9\$	25\$	15\$	10\$	20\$	5\$	40\$
Кількість переваг	12	11	14	12	10	12	13	11	13	12	10	12

Рисунок 3.12 Детальна порівняльна характеристика task трекінгових систем

В четвірку лідерів входять:

- Jira = 14 балів
- Asana = 13 балів
- Trello = 13 балів
- Monday = 12 балів

Висновки до розділу 3

Після проведеного дослідження і аналізу існуючих продуктів на ринку, моя персональна рекомендація полягає в тому, щоб кожна ІТ компанія підбирала менеджмент системи виключно під свої потреби, та відштовхувалась від своїх пріоритетів. Безпосередньо в тій компанії, що працюю я – незмінним базисом є Jira.

Керівники можуть делегувати свої повноваження та призначати відповідальних осіб, визначати пріоритетні завдання та визначати конкретні

терміни роботи над проектом. Управління проектами за допомогою встановлених стандартів та інструментів дозволяє значно збільшити ймовірність успішного завершення проекту, хоча це може супроводжуватися додатковими витратами, такими як зарплата керівника проекту, документація та звітність.

Бонусом використання програмного забезпечення для керування завданнями є скорочення термінів і витрат проекту завдяки уникненню непродуктивної роботи. У сучасному конкурентному бізнес-середовищі організації прагнуть оптимізувати свої процеси та зменшити витрати, щоб ефективно використовувати свої трудові ресурси та усунути втрату та спотворення актуальної інформації.

Однак не кожна компанія здатна успішно впровадити ці зміни. Ідеальним вирішенням цієї проблеми є використання програмного забезпечення для керування завданнями. Всі завдання та дії, що виконуються для їх вирішення, розміщені в одному місці, і система доступна через WEB практично в будь-який час. Керівництво компанії завжди бачить реальний стан справ. Розподіл відповідальності та встановлення пріоритетів за допомогою програмного забезпечення для керування завданнями здійснюється максимально швидко. Програмне забезпечення для управління завданнями допомагає модернізувати роботу над поставленими завданнями, роблячи її більш ефективною, незалежно від чисельності штату компанії та місця розташування її співробітників.

Кожному учаснику проекту надається список завдань і рівень їх пріоритетності та терміновості порівняно з іншими завданнями. Це програмне забезпечення є зручною системою для організації групової роботи над завданнями та управління проектами.

Менеджери проектів і члени команд повинні володіти балансом соціальних і технічних навичок, включаючи самоуправління, комунікацію, технічну експертизу в управлінні проектами та своїй галузі. Для досягнення необхідного рівня організаторських здібностей і лідерства необхідне навчання. Поєднання навичок допомагає у вирішенні проблем, виконанні рутинних завдань і забезпеченні необхідних змін у проекті.

ВИСНОВКИ

У будь-якому проекті важливо мати систему керування завданнями. Один із підходів - створити модель з відкритим вихідним кодом, що може бути вдосконалена через практичне тестування. Для поліпшення моделі, рекомендується визначити пріоритетність комплексного управлінського підходу, проаналізувати наслідки змін, використовувати універсальні засоби розробки та розглянути альтернативні погляди на впровадження.

При виборі універсального фреймворку варто враховувати різні домени, досвід реалізації та можливості розвитку фреймворку. Ці аспекти допоможуть вам вибрати та створити методи застосування, що відповідають запропонованій практичній моделі.

Однією з головних переваг гнучких методологій є те, що вони сприяють співпраці та комунікації між членами команди, що веде до кращого розуміння проекту та його цілей. Це, у свою чергу, призводить до вищого рівня продуктивності та більш ефективного використання ресурсів.

Крім того, методології Agile дозволяють безперервно вдосконалюватися, заохочуючи команди обмірковувати свою роботу та вносити необхідні зміни.

Методологія водоспаду — це традиційний підхід до управління проектами, який широко використовується в різних галузях. Він дотримується підходу зверху вниз, коли кожен етап проекту завершується перед переходом до наступного. Цей підхід найбільш ефективний у створенні конкретних кінцевих продуктів, таких як медичні чи будівельні проекти. Однак успіх проекту водоспаду залежить від обсягу виконаної попередньої роботи, особливо дослідження, яке дозволяє точніше оцінити завершення проекту.

Гнучкі методології також мають деякі недоліки. Однією з головних проблем є те, що їх може бути важко реалізувати, особливо для команд, які звикли до більш традиційних підходів. Крім того, методології Agile вимагають високого рівня співпраці та спілкування, що може бути складним завданням для віддалених команд або команд, які розкидані в різних місцях. Незважаючи на його ефективність,

негнучкість методу каскаду ускладнює реагування на невизначеність і зміни, що може призвести до затримок і перевитрат.

Як наслідок, більш гнучкі методи, такі як Agile, стають все більш популярними, особливо в галузі розробки програмного забезпечення. Гнучкі методології зосереджені на співпраці, постійному вдосконаленні та здатності швидко реагувати на зміни. І водоспад, і методологія Agile мають свої переваги та недоліки. Хоча водоспадний метод ефективний у створенні конкретних кінцевих продуктів, його негнучкість може бути перешкодою. У сучасному стрімкому бізнес-середовищі дуже важливо вміти адаптуватися до мінливих ринкових умов, потреб клієнтів та інших факторів. Гнучкі методології дозволяють командам швидко й ефективно реагувати на ці зміни, що може призвести до конкурентної переваги.

Методологія Agile містить набір із 12 принципів які були описані у висновках другого розділу, які підкреслюють різні переваги цього підходу. Ці принципи служать керівництвом для компаній, які прагнуть застосувати гнучкі методи роботи, і мають на меті запровадити ефективніший спосіб роботи.

Простота необхідна для досягнення максимальної продуктивності шляхом усунення непотрібної роботи. Пильна увага до технічної досконалості та гарного дизайну підвищує маневреність. Увага до деталей має вирішальне значення. Працююче програмне забезпечення є основним мірилом прогресу, і воно не повинно бути ідеальним, але воно має бути функціональним. Важливо надавати функціональне програмне забезпечення за коротші терміни, послідовно протягом усього проекту, а не лише в кінці. Головним пріоритетом є забезпечення задоволеності клієнтів шляхом постійного своєчасного надання високоякісного програмного забезпечення. Потреби та задоволеність клієнта стоять на першому плані всіх рішень. Періодичне обмірковування того, як покращити та відповідне коригування поведінки є життєво важливим для ефективності команди.

Зворотній зв'язок важливий для коригування та покращення продуктивності. Щоденна співпраця між бізнесменами та розробниками має вирішальне значення для успіху проекту. Впровадження змін, навіть на більш пізніх стадіях розробки, заохочується в гнучких процесах, оскільки це може підвищити

конкурентоспроможність клієнта. Змінам не слід чинити опір, а радше прийняти їх і використовувати. Команди, що само організуються, створюють найкращі архітектури, вимоги та проекти. Якщо дозволити членам команди визначити свої ролі в поєднанні з принципом 5, це дає оптимальні результати. Спілкування віч-на-віч є найефективнішим способом обміну інформацією всередині команди розробників, щоб уникнути непорозумінь. Вмотивовані люди є наріжним каменем успіху проекту. Створення сприятливого середовища та довіра членам команди виконувати свою роботу залучає та утримує талановитих і працьовитих людей. Гнучкі процеси забезпечують сталий розвиток, і всі зацікавлені сторони повинні підтримувати постійний темп.

Програмне забезпечення для керування завданнями пропонує низку переваг, у тому числі можливість скоротити терміни проекту та зменшити витрати за рахунок мінімізації непродуктивної роботи. У сучасному конкурентному бізнес-середовищі організації постійно прагнуть оптимізувати свої процеси та скоротити витрати, щоб максимально використовувати свої трудові ресурси, мінімізуючи втрати та спотворення важливої інформації. Однак не кожна організація здатна ефективно впровадити ці зміни. Саме тут програмне забезпечення для керування завданнями стає ідеальним рішенням. Централізувавши всі завдання та дії в одному місці та зробивши систему доступною через Інтернет, компанії можуть мати чітке уявлення про свій прогрес у режимі реального часу. За допомогою програмного забезпечення для керування завданнями розподіл обов'язків і пріоритетність завдань виконується швидко й ефективно, незалежно від розміру чи розташування робочої сили компанії. Це програмне забезпечення допомагає модернізувати керування завданнями, роблячи його ефективнішим і ефективнішим для компаній будь-якого розміру.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Придатко О. В., Смотр О. О., Мартин Є. В., Придатко В. В., Солотвінський І. В. Оптимізація методів теорії масового обслуговування для вирішення прикладних завдань розвитку регіональних систем безпеки життєдіяльності // Системи обробки інформації. Харків, 2019. № 2 (157). С. 146–153.
2. Андерсон, Д. Kanban – Успішні еволюційні зміни для вашого технологічного бізнесу, 2010. Blue Hole Press.
3. Ларман, С. & Басілі, В. Історія ітеративного та поступового розвитку, IEEE Computer 2003. 36(6), 47-56 с.
4. Пападопулос, Г. Перехід від традиційного до гнучкого програмного забезпечення методології також на великих, розподілених проектах. Процедурно-соціальна та поведінкова науки 2015., 175 с., 455-463 с.
5. Agile manifesto [Електронний ресурс] – Режим доступу до ресурсу: <http://agilemanifesto.org>.
6. Henrik Kniberg. Scrum and XP from the trenches, 2007. – 140 с.
7. Classical Waterfall Model – Режим доступу до ресурсу: <https://www.geeksforgeeks.org/software-engineering-classicalwaterfall-model/>.
8. Боегм, Б., Будьте готові до гнучких методів з обережністю. IEEE Computer 2002 р., 35, 64–69.
9. Кемпбел-Келлі, М. Історія історії програмного забезпечення. IEEE літописи Історія Computing 2007 р., 29 (4), 40 с.
10. Чеші, М., Сіллітті, А., Суккі, Г., Панфіліс, С.Д., Управління проектами в планбазованих і гнучких компаніях. Програмне забезпечення IEEE 2005 р., 22 (3), 21-27 с.
11. Чоу, Т., Чао, Д.Б., Дослідження критичних факторів успіху в гнучкому програмному забезпеченні проекти. Журнал систем і програмного забезпечення 2008 р. 81, 961-971 с.

12. Кокберн, А., Agile Розробка програмного забезпечення. Аддісон-Уеслі, Бостон, Массачусетс 2002 р.
13. Dingsøyr, T., Дібо, Т., Моє, Н.Б., Agile Розробка програмного забезпечення: Поточні дослідження та майбутні напрямки. Спрінгер, Берлін/Гейдельберг 2010 р.
14. Ейнштардт, К.М., Грайбнер, М.Е. Теорія побудови з випадків: можливості та виклики. Журнал Академії менеджменту, 2007 р. 50 (1) с., 25-32 с.
15. Кох, А.С., Agile Розробка програмного забезпечення: Оцінка методів для вашої організації. Artech House, Нортвуд, Массачусетс. 2005 р.
16. Ларман, С., Масштабування Lean Agile\Large Scaling Lean & Agile: Large, Multisite – Режим доступу до ресурсу: http://qconlondon.com/dl/qconlondon2011/slides/CraigLarman_ScalingLeanAgileLargeMultisiteOrOffshoreDelivery.pdf
17. Малік, Х., & С'ю Хок, О. Огляд Agile методології в програмному забезпеченні розробки. Міжнародний журнал досліджень та оглядів у галузі прикладних наук, 2009 р., 1 (1) с.
18. Месо, П., Джейн, Р., Гнучка розробка програмного забезпечення: принципи адаптивних систем і кращі практики. Управління інформаційними системами 2006 р. 23 с., 19-30 с.
19. Мо, Н.Б., Дінгсойр, Т., Диба, Т.,. Командна модель для розуміння гнучкої команди: кейс проекту Scrum. Інформаційно-програмні технології 2010 р., 52 с., 480–491 с.
20. Нерур, С., Махапатра, Р.К., Мангаларадж, Г., Проблеми міграції на гнучких методології 2005 р., Комунікації АСМ 48 (5) с., 72-78 с.