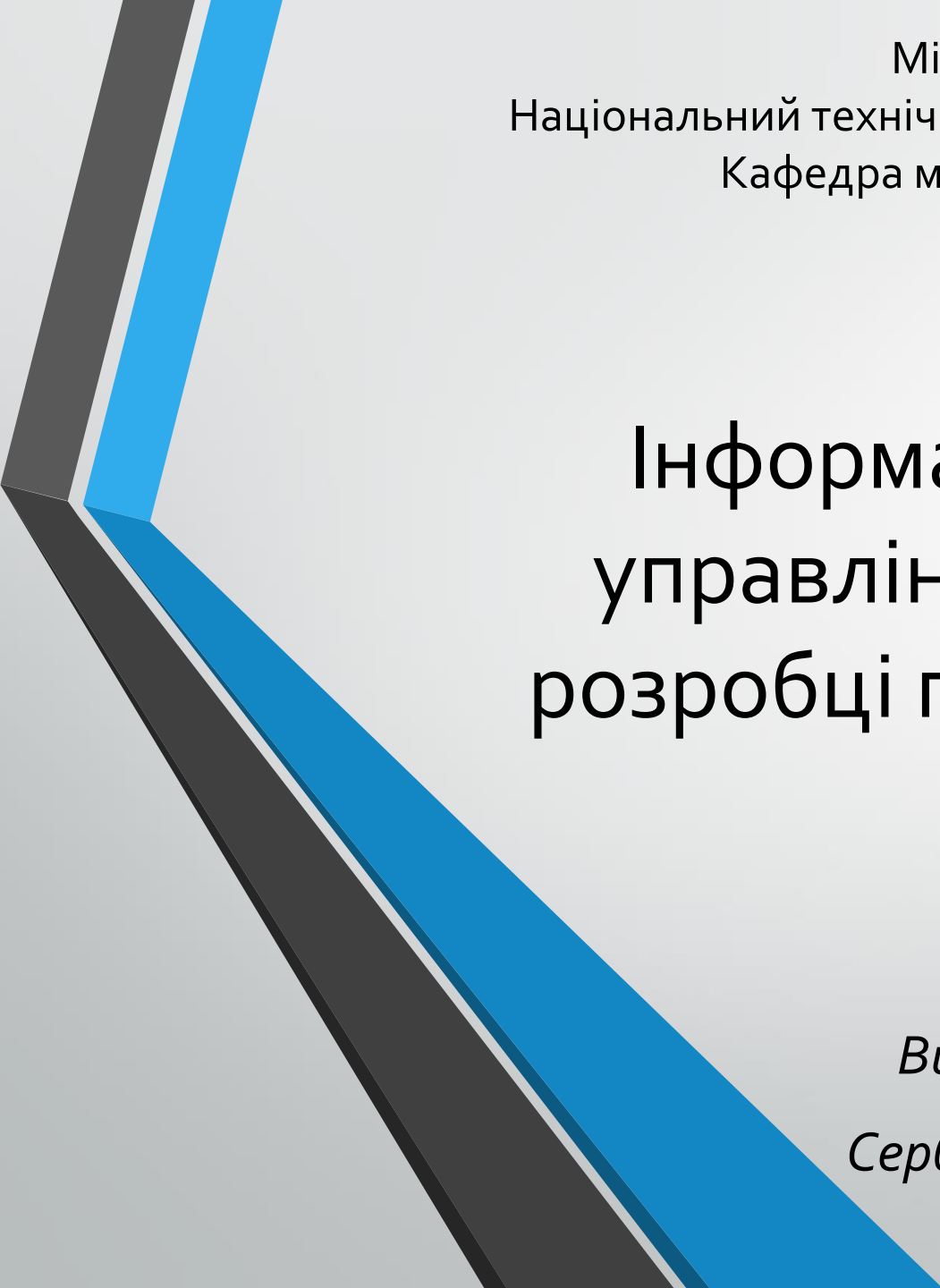




Міністерство освіти і науки України
Національний технічний університет України «КПІ ім. Ігоря Сікорського»
Кафедра математичних методів системного аналізу

Інформаційно-аналітична система управління проектною командою в розробці програмного забезпечення

Виконав:
Сербул Р.С.

Керівник:
проф. Данилов В.Я.

Актуальність роботи

- Процес впровадження систем управління є трудомістким і пов'язаний з великою кількістю ризиків, що може привести до негативних результатів
- "Корпорації стають проектно орієнтованими"
- Аналіз стану проекту в режимі реального часу
- Гнучкість та масштабованість системи
- Автоматизація розгортання інфраструктури системи
- Віддалений моніторинг стану проекту

Основні задачі роботи

- Аналіз існуючих підходів
- Обґрунтування запропонованих в рамках роботи рішень
- Аналітичний модуль системи
- Проектування об'єктно-орієнтованої моделі системи
- Розробка web-додатку системи управління завданнями в вигляді Kanban-дошок
- Формування проектної команди

Мета дослідження:

- розробка веб-додатку для аналізу та управління ІТ-проектом.

Об'єкт дослідження:

- Система управління проектами в розробці програмного забезпечення.

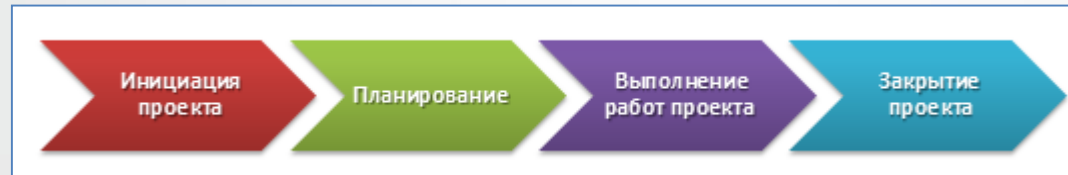
Методи дослідження:

- об'єктно-орієнтоване програмування, методологія Kanban, Agile, Docker-контейнери, інтелектуальний аналіз даних, нейронні мережі.

Актуальність задач управління ІТ-проектами

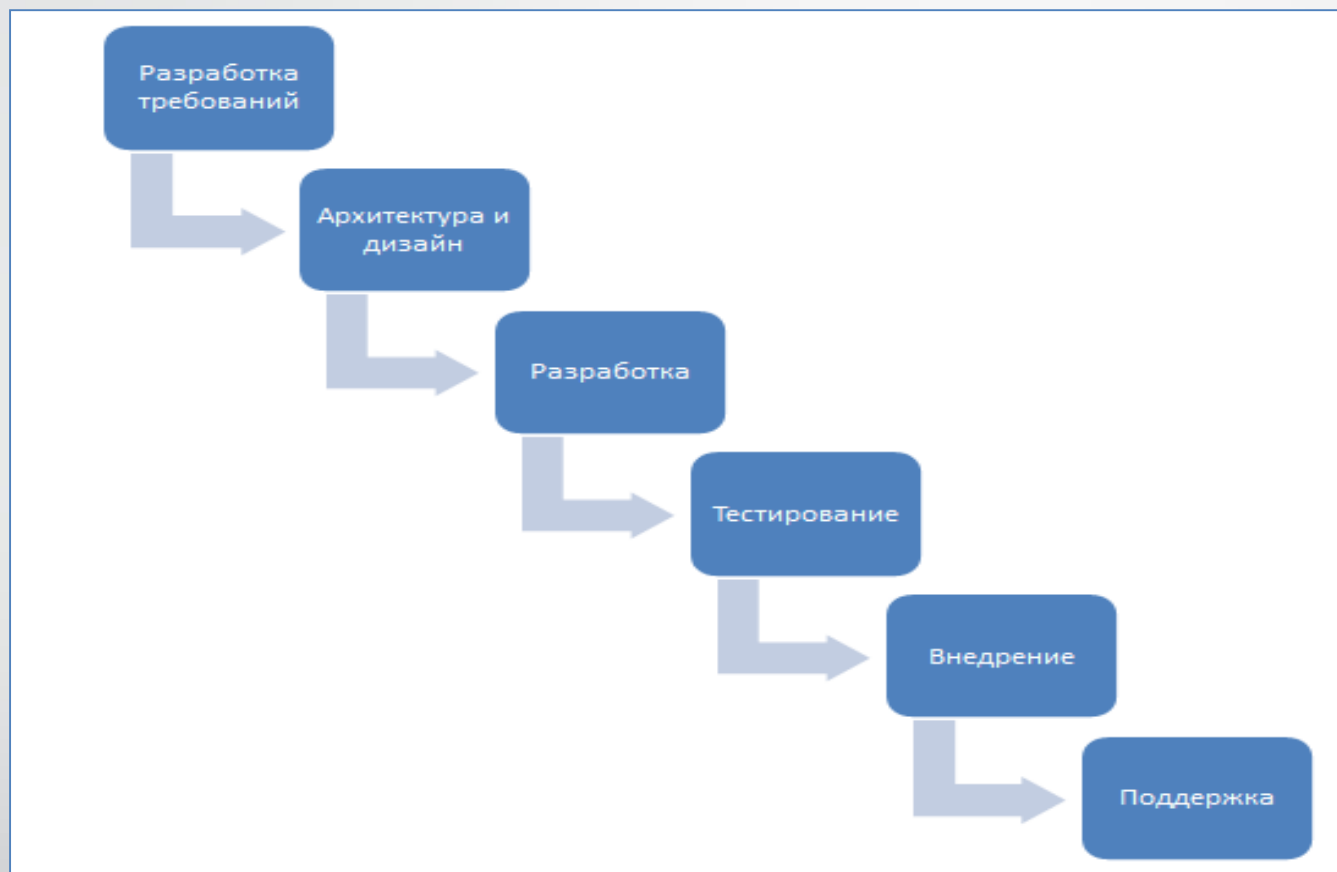
З розвитком інформаційних технологій одна частина організацій стала автоматизувати свою діяльність, а друга частина почала створювати програмне забезпечення для його подальшого постачання організаціям першого типу. І в першому і в другому випадку, організації мають справу з ІТ-проектами.

Моделі життєвого циклу ІТ проекту: їх характеристики і особливості



У найзагальніших рисах схема життєвого циклу ІТ-проекту може бути представлена чотирма фазами: ініціація, планування, виконання та закриття

Водоспадна модель життєвого циклу ІТ-проекту



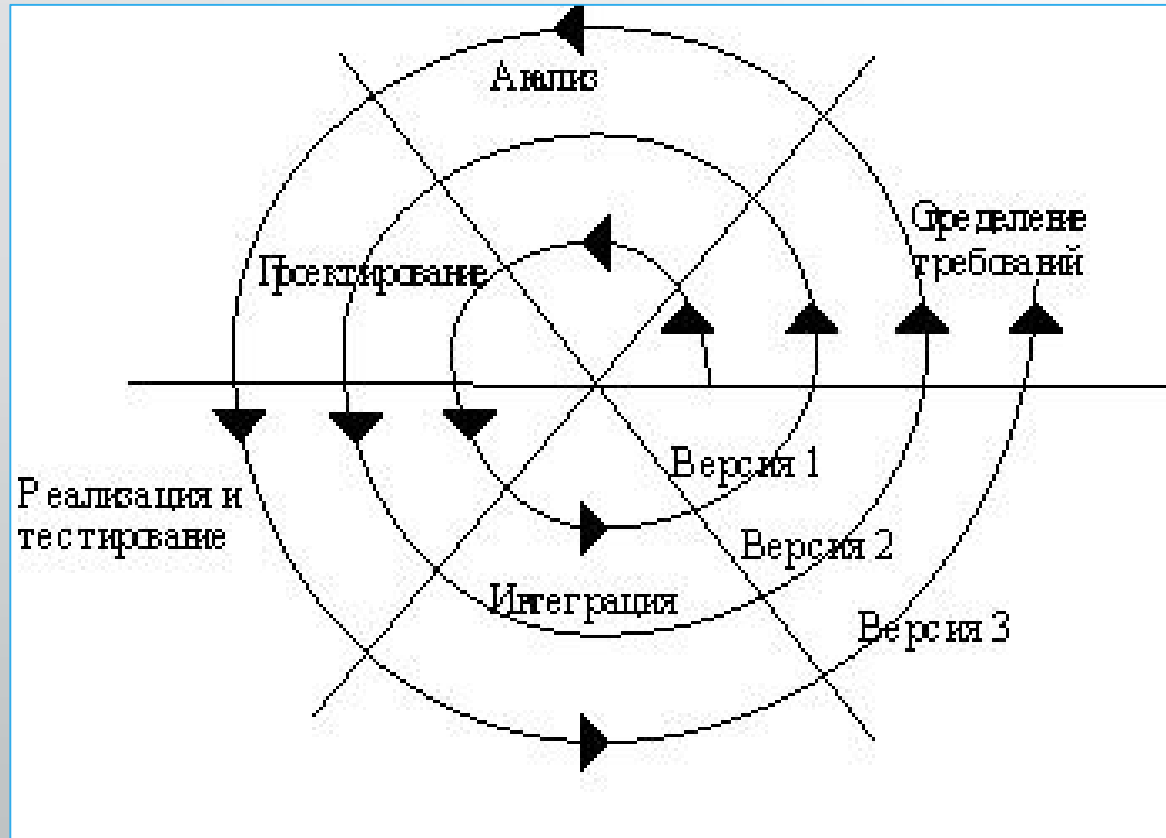
Ключовим моментом Водоспадної моделі є її послідовний характер, що означає, що кожна фаза не може початися, якщо не завершена попередня

Ітеративна модель життєвого циклу ІТ-проекту



Кожна ітерація додає новий функціонал в програмний продукт

Спиральна модель життєвого циклу ІТ-проекту

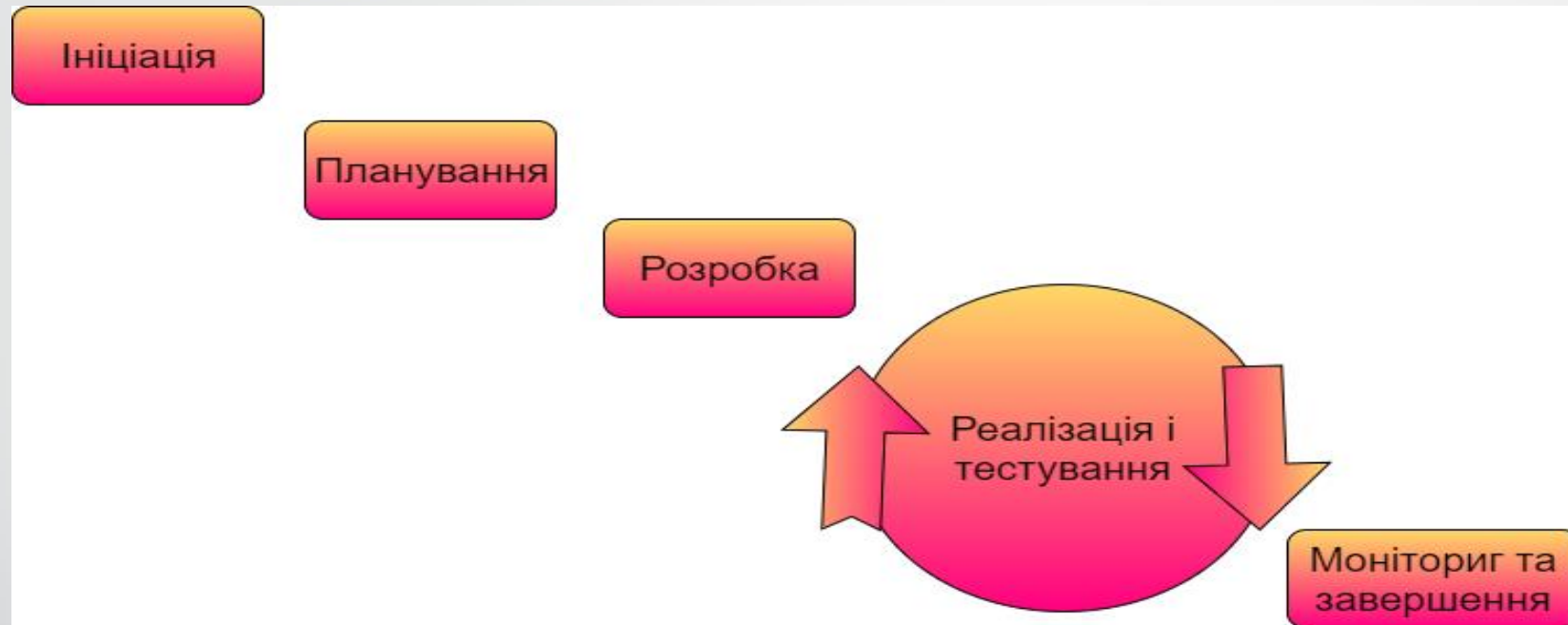


Кожен «виток спіралі» представляє собою закінчену частину продукту, відповідно, з кожним новим «витком» проект досягає більш глибокий рівень деталізації і конкретизації

Сучасні підходи та методології в управлінні ІТ-проектам

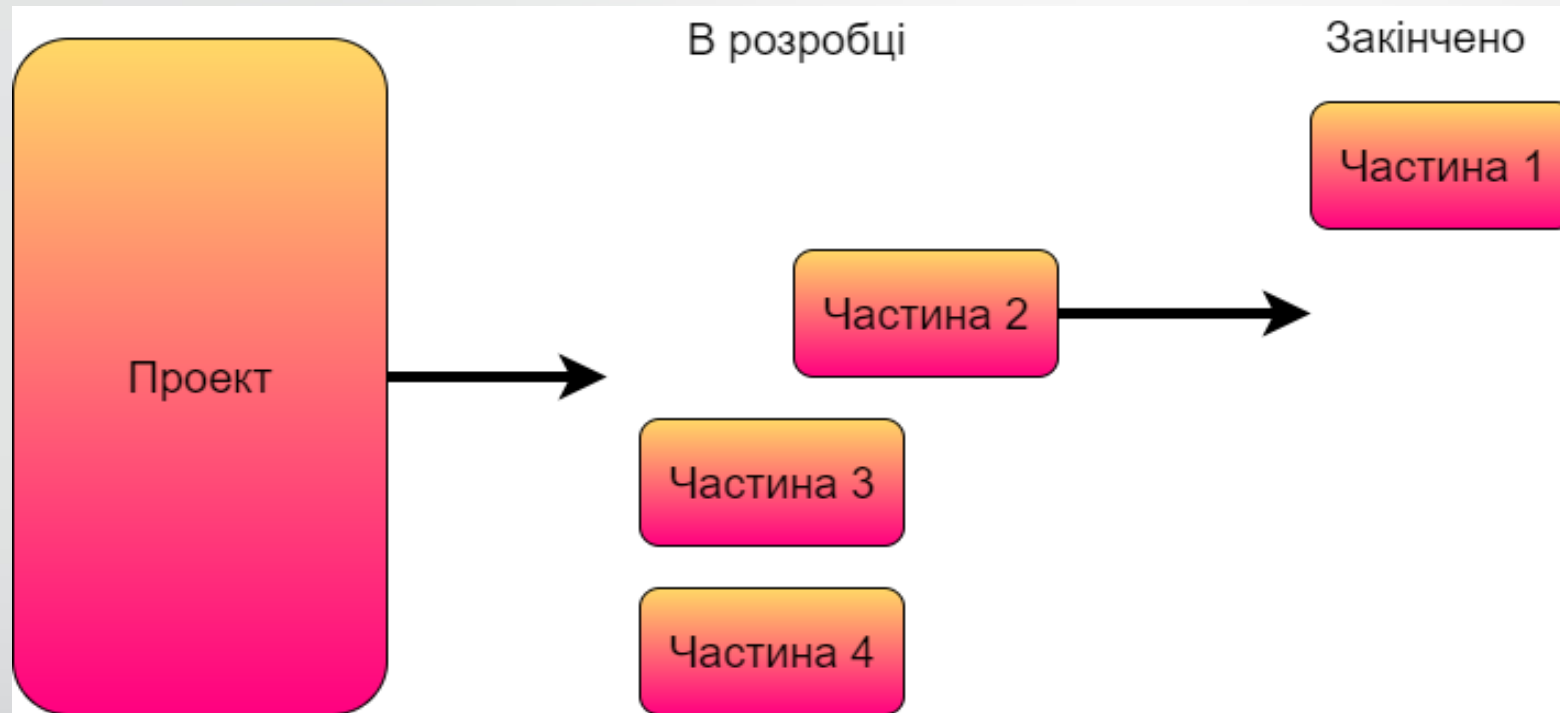
- Класичний проектний менеджмент
- Agile
- Scrum
- Lean
- Kanban

Класичне проектне управління



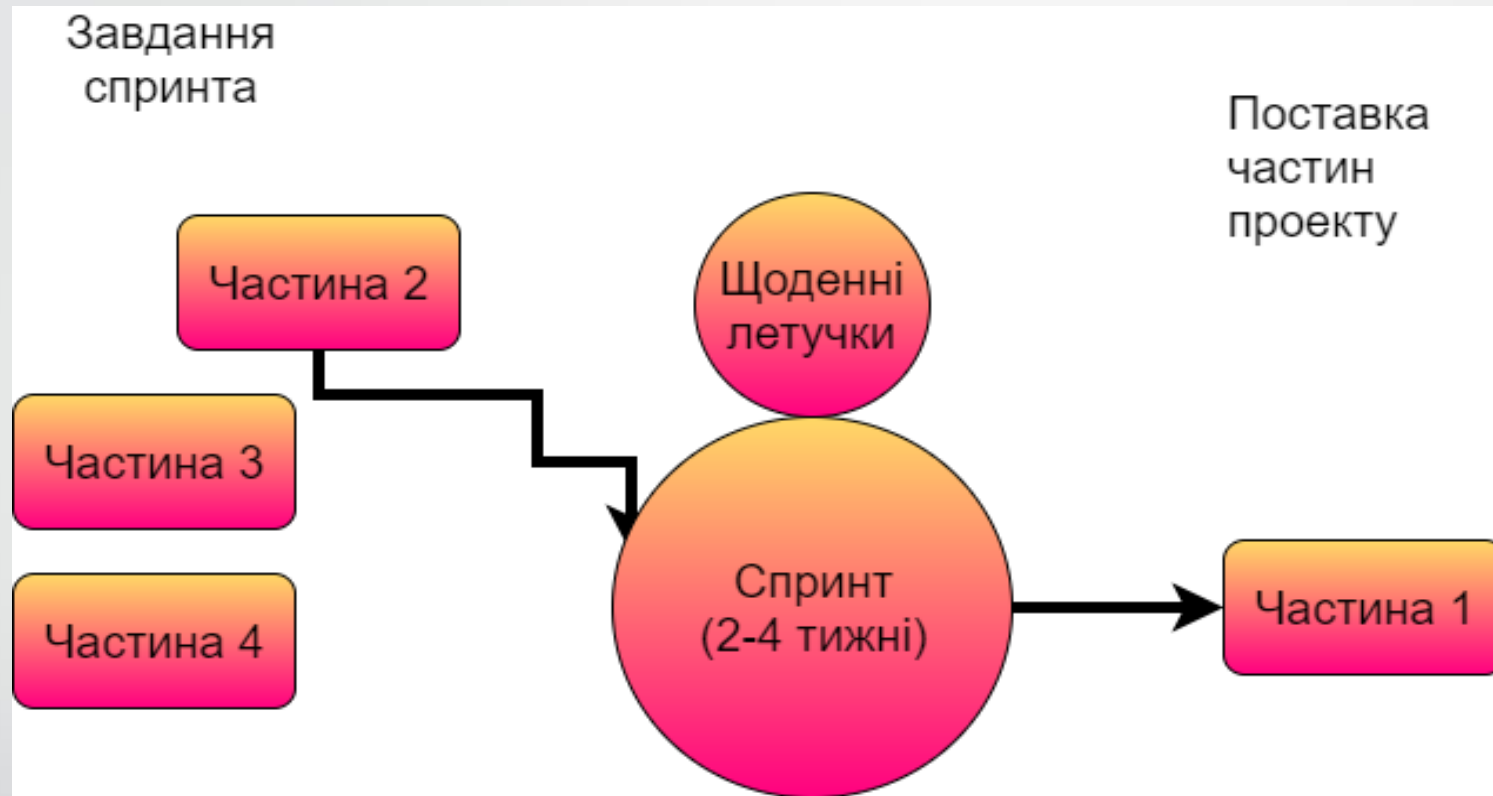
Даний підхід орієнтований на проекти, в яких є строгі обмеження по послідовності виконання завдань.

Гнучка методологія Agile



Проект розбивається не на послідовні фази, а на маленькі підпроекти, які потім «збираються» в готовий продукт

Фреймворк Scrum



Scrum дуже вимогливий до команди проекту. Вона повинна бути невеликою (5-9 чоловік) і кроссфункціональною - тобто члени команди повинні володіти більш ніж однією компетенцією, необхідної для реалізації проекту

Концепція управління Lean



Етапи Lean і їх гнучкість дозволяють бути впевненими в тому, що кожна частина проекту реалізується так, як потрібно, але Lean не пропонує чіткого робочого процесу для реалізації «шматочків» проекту, що сприяє розтягуванню термінів проекту.

Метод управління розробкою Kanban



Lean виглядає трохи абстрактним сам по собі, але в комбінації з Kanban його стає набагато простіше використовувати для побудови власної системи управління проектами

Архітектура системи по методології Kanban

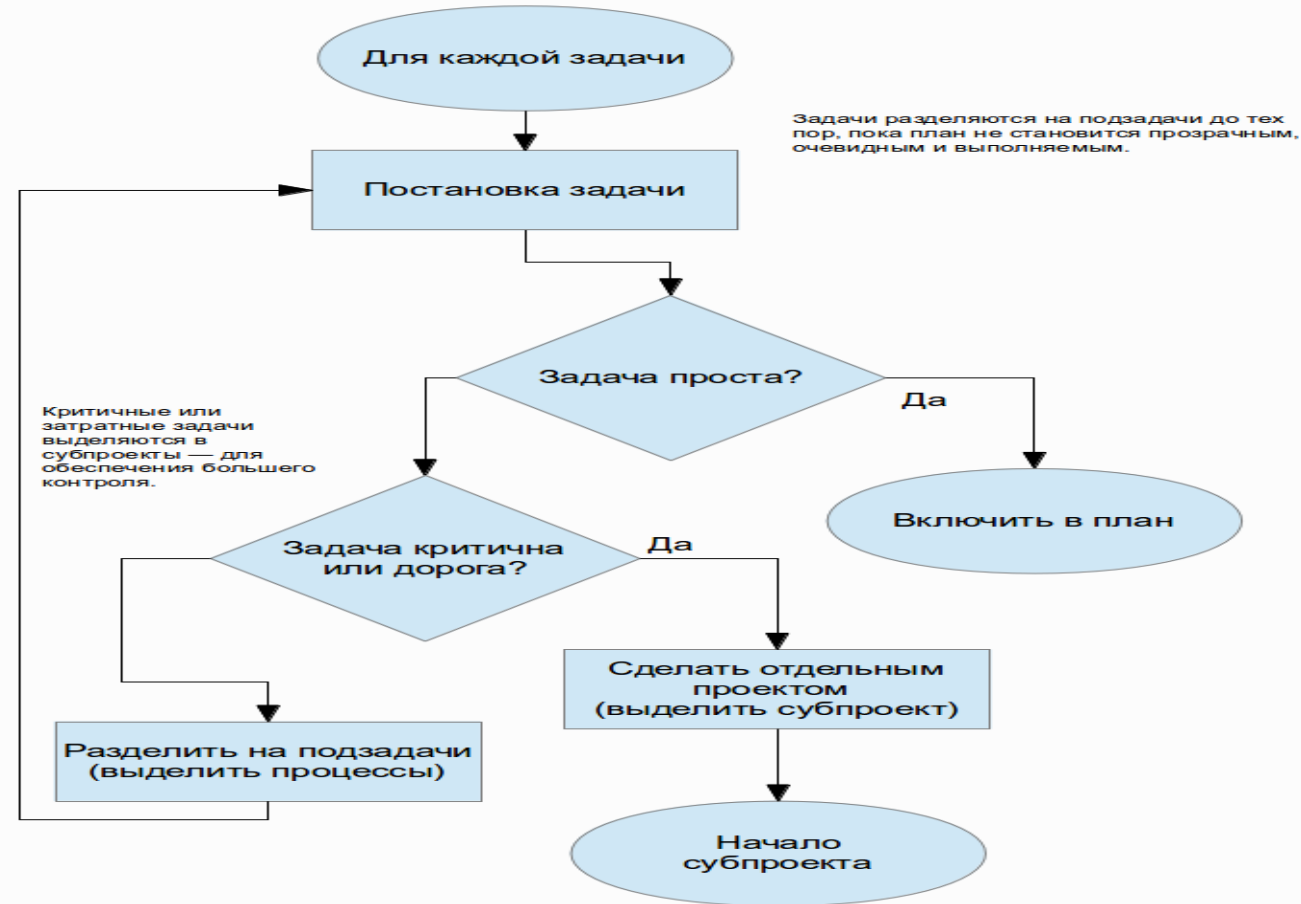
Kanban - методологія управління виробництвом, проектами, найбільш гнучкий з підходів Agile, в основі якого лежать кілька принципів, яких необхідно дотримуватися:

- Візуалізація процесу розробки. Візуальне представлення процесу розробки допомагає швидко визначити, яке завдання на якій стадії виконання знаходиться. Це може бути вкрай корисно в разі великого проекту, що складається з безлічі завдань.
- Мінімізація WIP. Обмеження числа завдань, що виконуються на кожній стадії розробки, допомагає ефективно розподіляти наявні ресурси і не викликати простоїв в роботі.
- Вимірювання і оптимізація життєвого циклу розробки. Можливість вносити зміни в процес розробки є одним з важливих компонентів Agile методології.

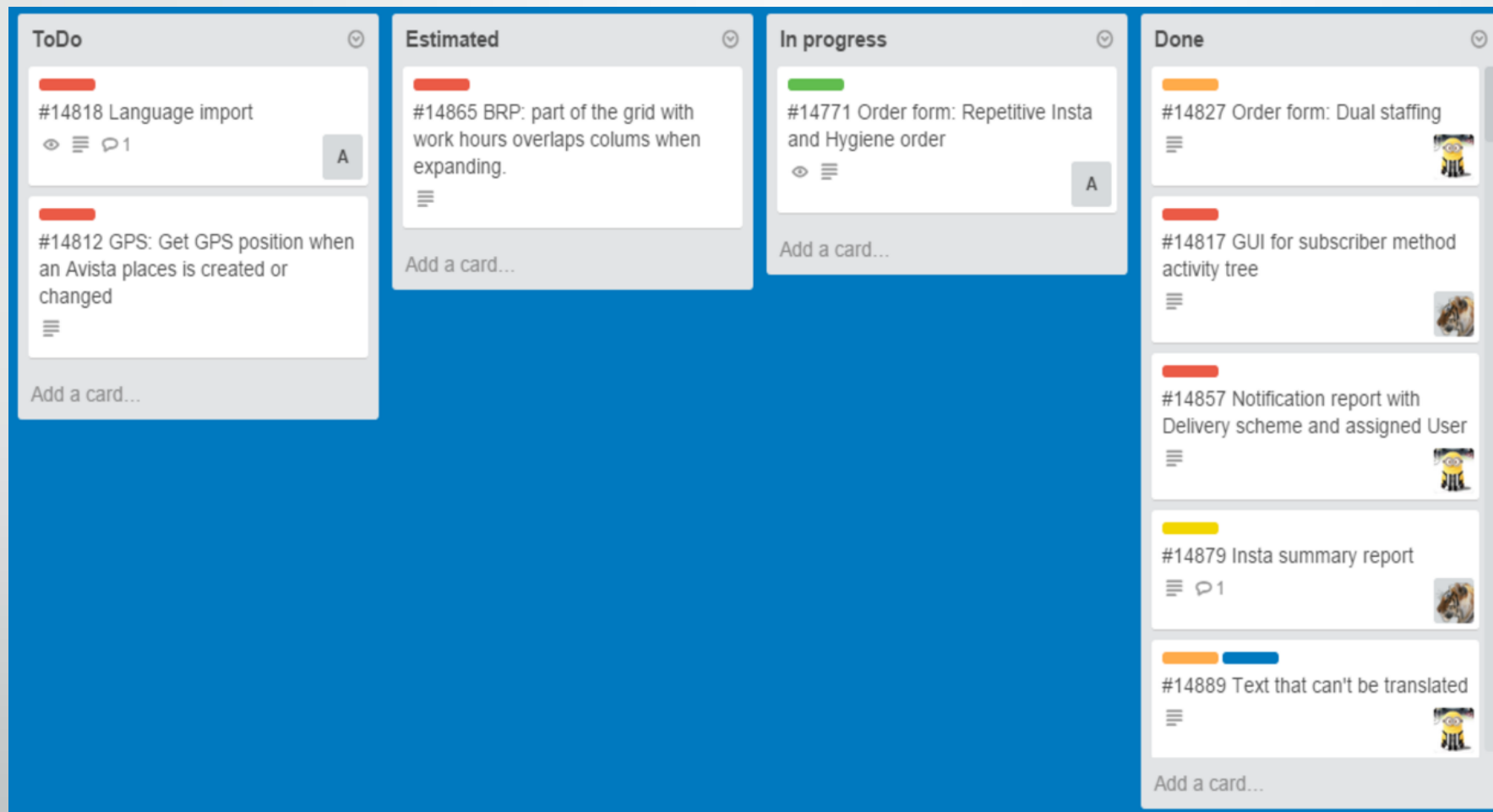
Поділ процесу розробки на етапи

- Беклог проекту
- Вимоги
- Дизайн
- Розробка
- Тестування
- Деплоймент
- Закінчено

Алгоритм поділу задач на підзадачі



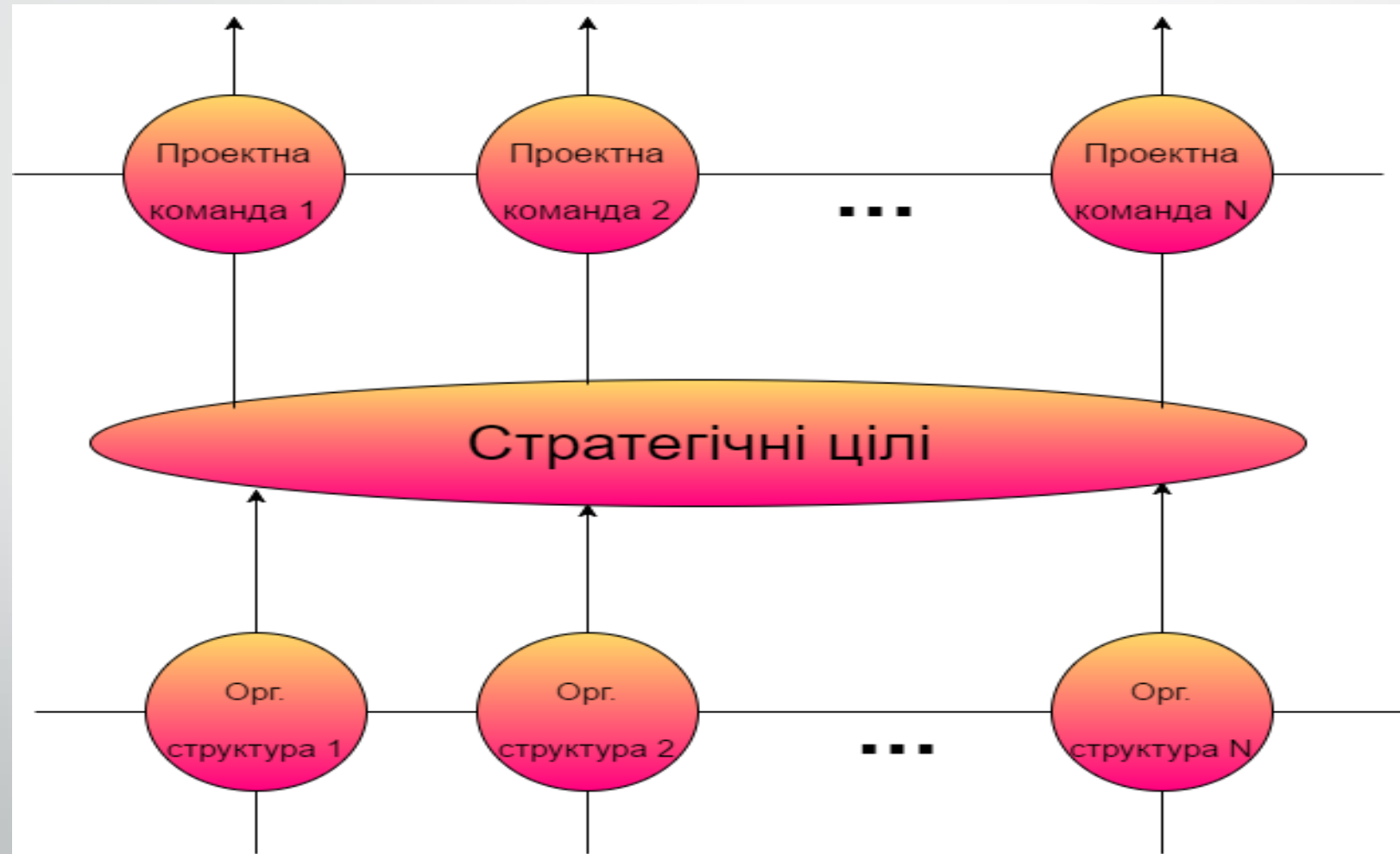
Візуалізація за допомогою Канбан-дошки



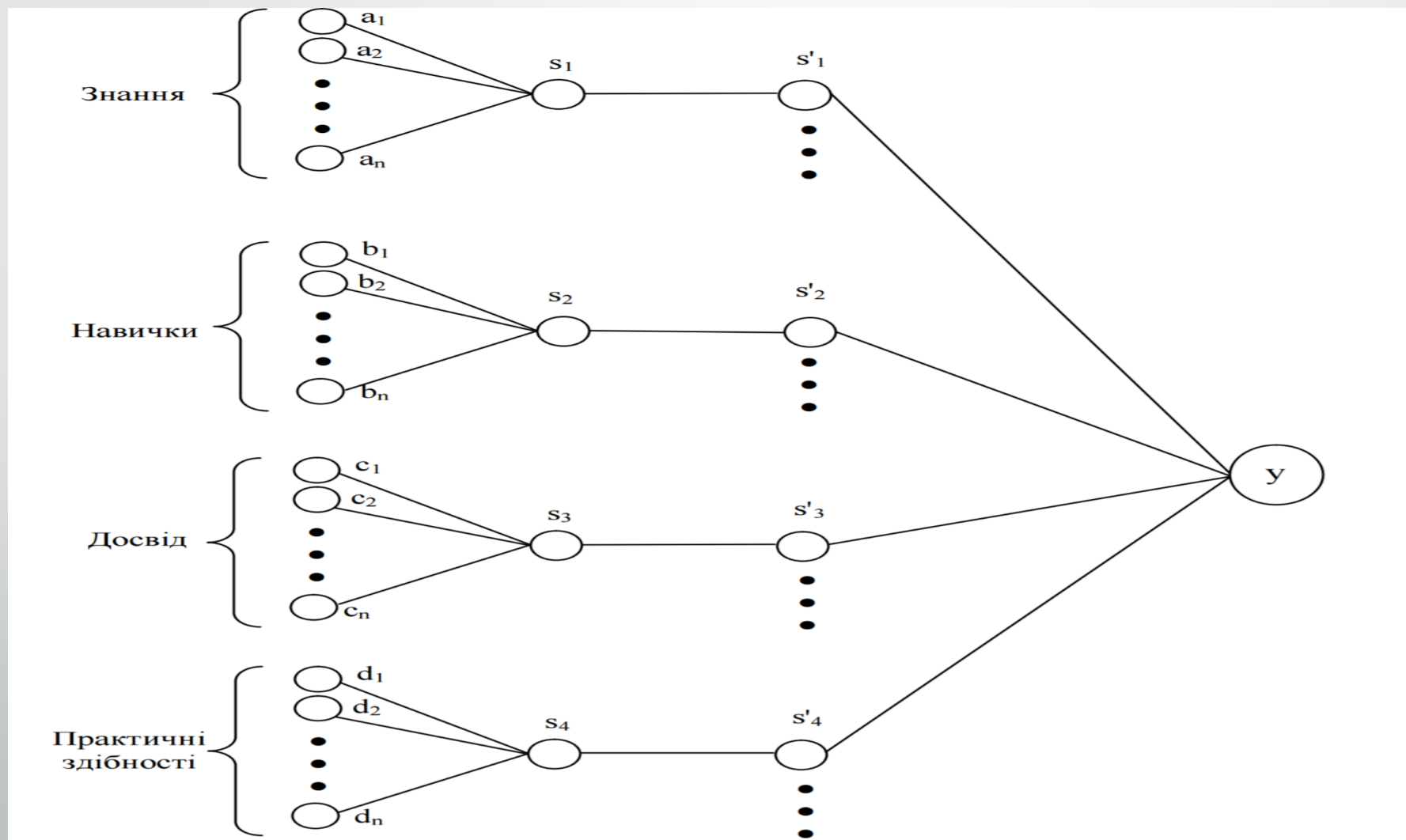
Метрики в Kanban

- Cycle Time - час, яке завдання перебувала в розробці від моменту, коли нею почали займатися, до моменту, коли вона пройшла фазу кінцевої поставки.
- WIP - кількість завдань які одночасно знаходяться в роботі. Розділяється за різними стадіями роботи над завданням.
- Lead Time - час від появи завдання до її кінцевої поставки. Включає час циклу і час очікування в черзі на реалізацію.
- Wasted Time - час, яке завдання проводить в різних чергах, а не безпосередньо в роботі.
- Effectiveness - відсоток часу, який витрачається безпосередньо на роботу з завданням, а не на очікування в різних чергах.
- Throughput - кількість завдань, яке може виконувати команда в одиницю часу (день, тиждень, місяць).

Організаційна модель проектних команд в розробці ПЗ



Модель відбору персоналу в проектну команду з використанням теорії нейронних мереж



Загальна архітектура розробленої системи

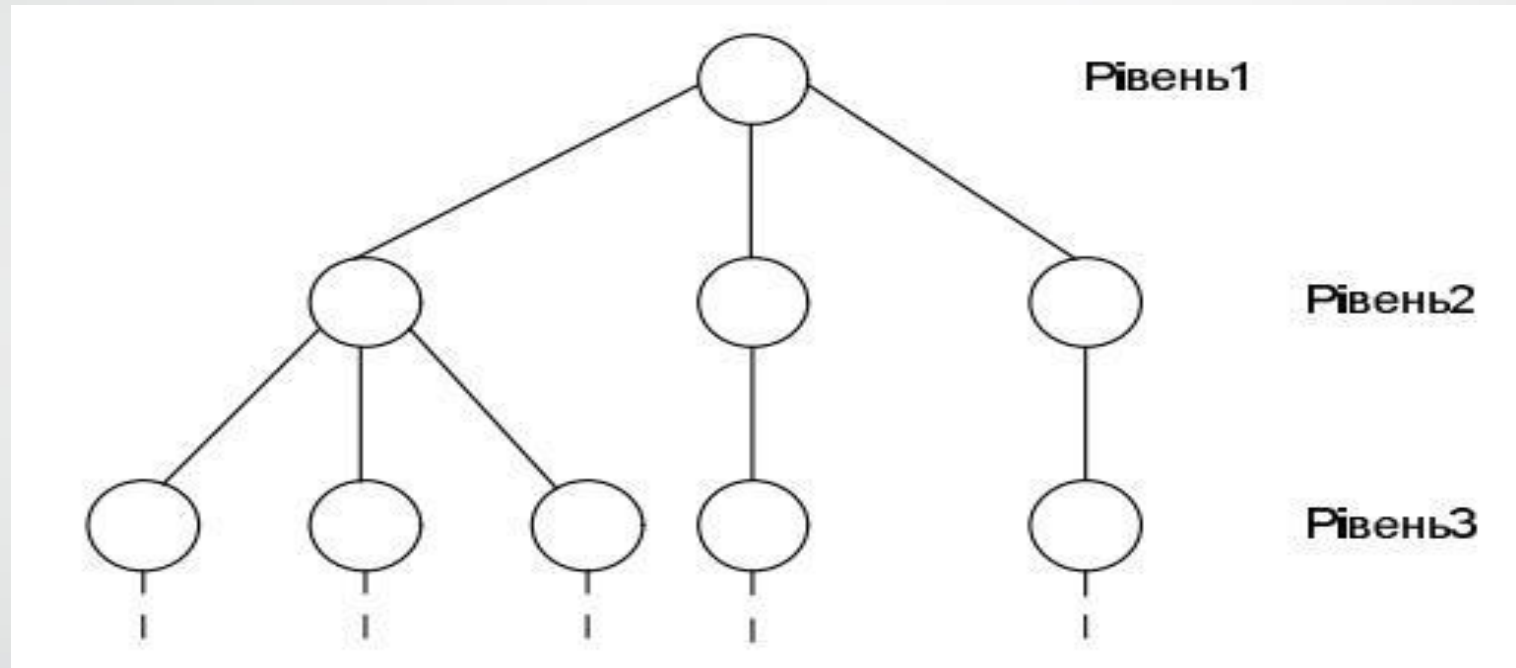


Модель розробленої системи

Складність корпоративних інформаційних систем є не випадковим процесом, а більш за все їх властивістю. Вона визначається рядом причин, серед яких можна назвати наступні:

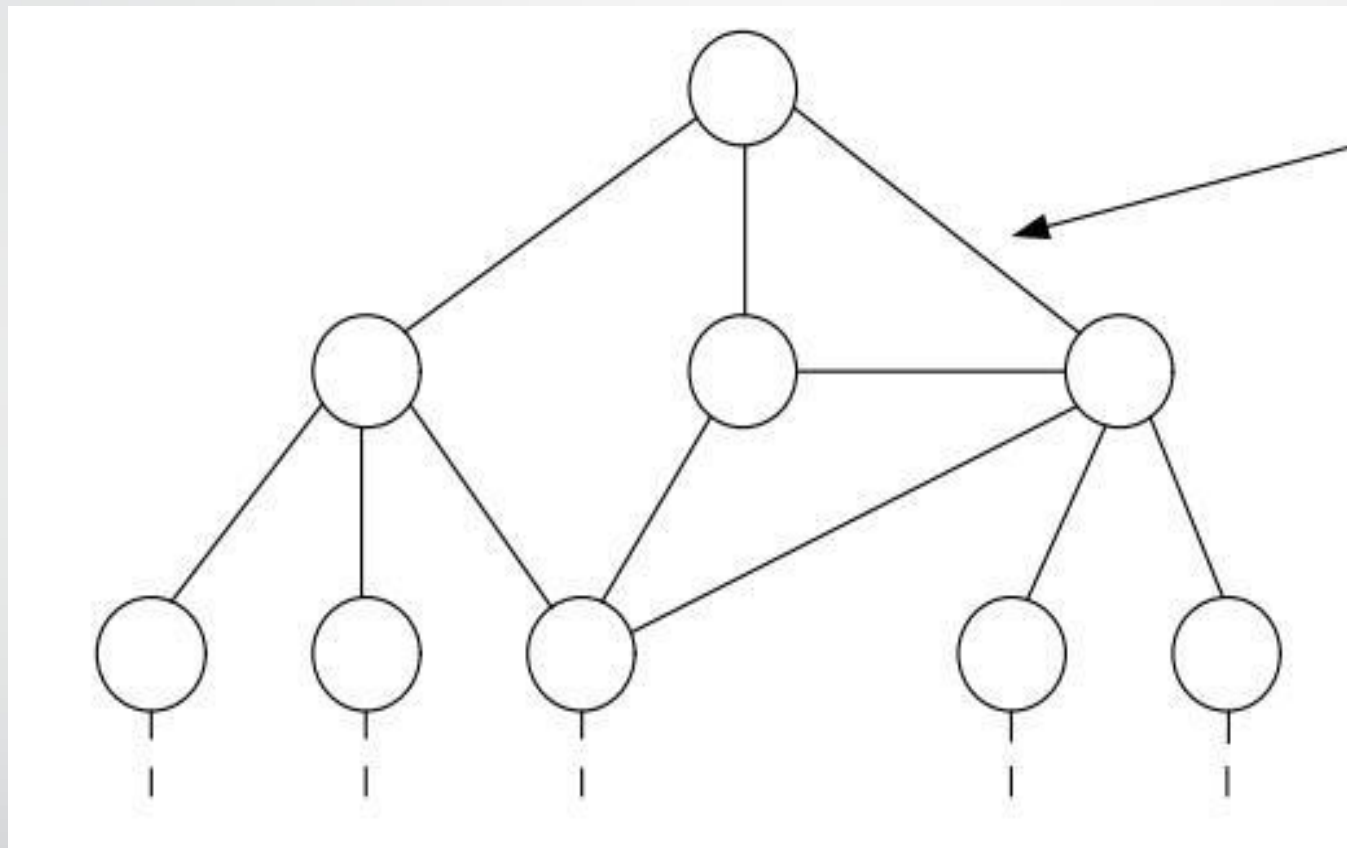
- складність розв'язуваної задачі;
- складність розробки ІС;
- складність забезпечення таких параметрів, як адекватність, масштабованість, надійність, економічна ефективність і безпека;
- складність опису окремих підсистем ІС.

Об'єктна модель системи



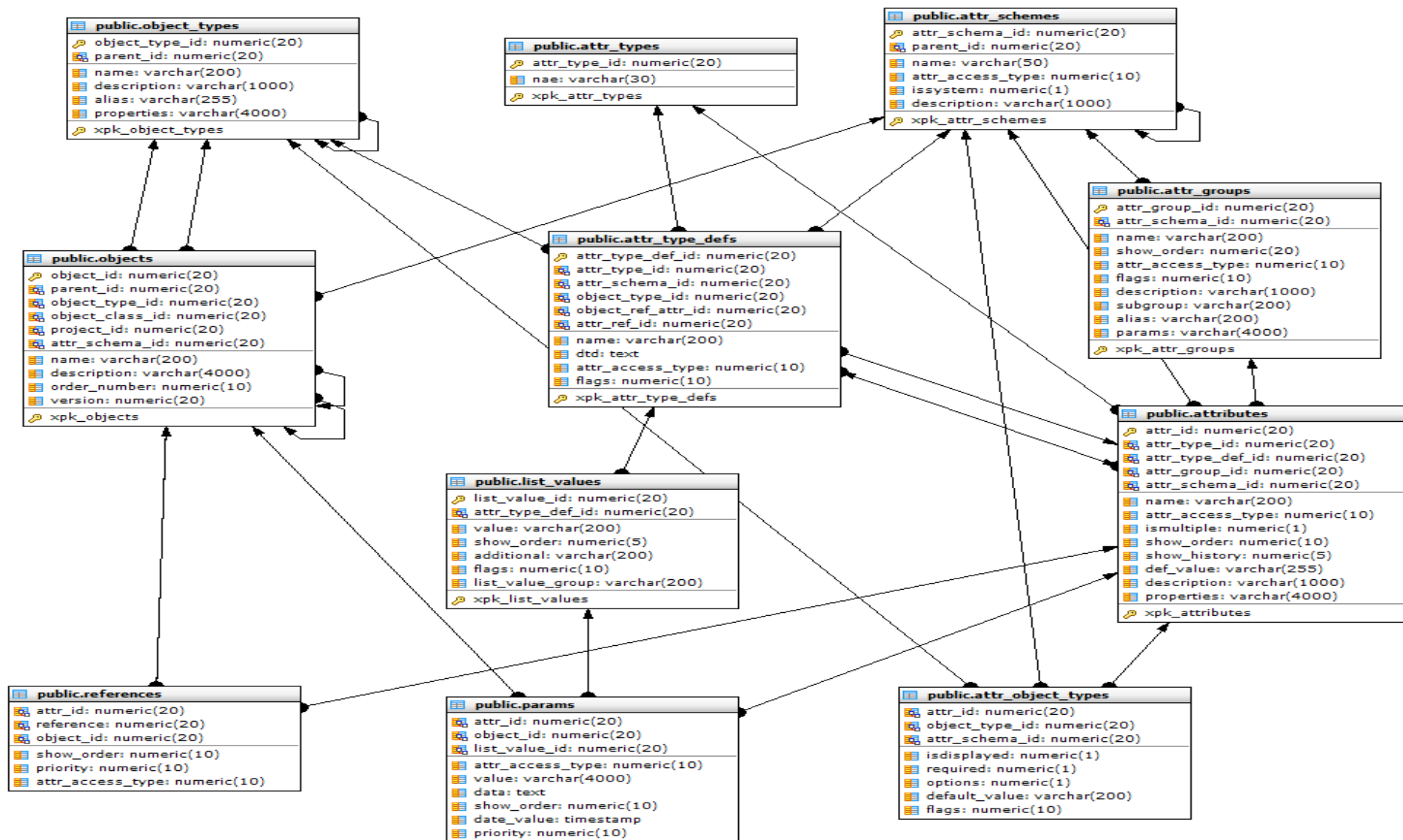
Як правило, системи, які базуються на об'єктно-орієнтованій моделі даних, найбільше функціональні, гнучкі, але, разом з тим, і найскладніші

Мережева модель представлення даних



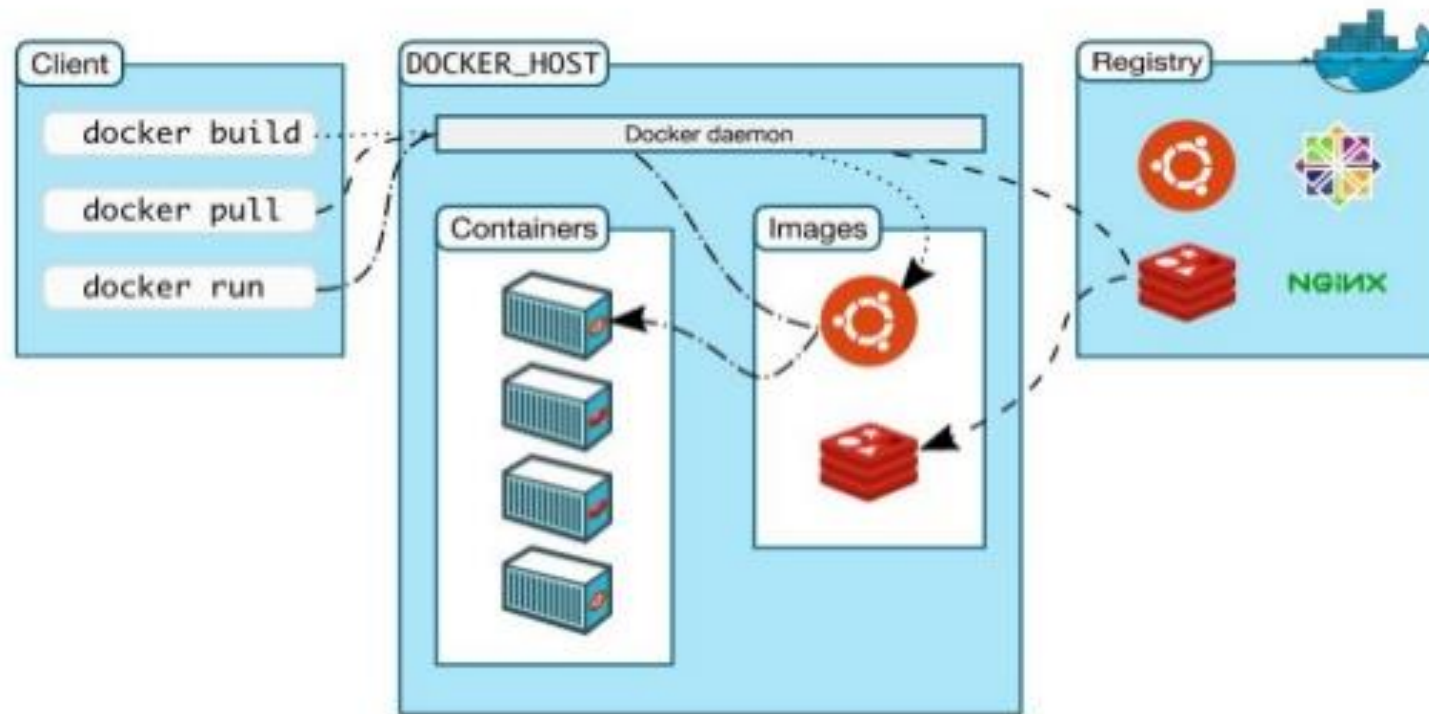
Предметну область будь-якої задачі програмування будемо вважати представленою у вигляді сукупності функціональних елементів (об'єктів), що обмінюються в процесі виконання програми повідомленнями. Процес такого представлення називається об'єктною декомпозицією

Схема спроектованої бази даних

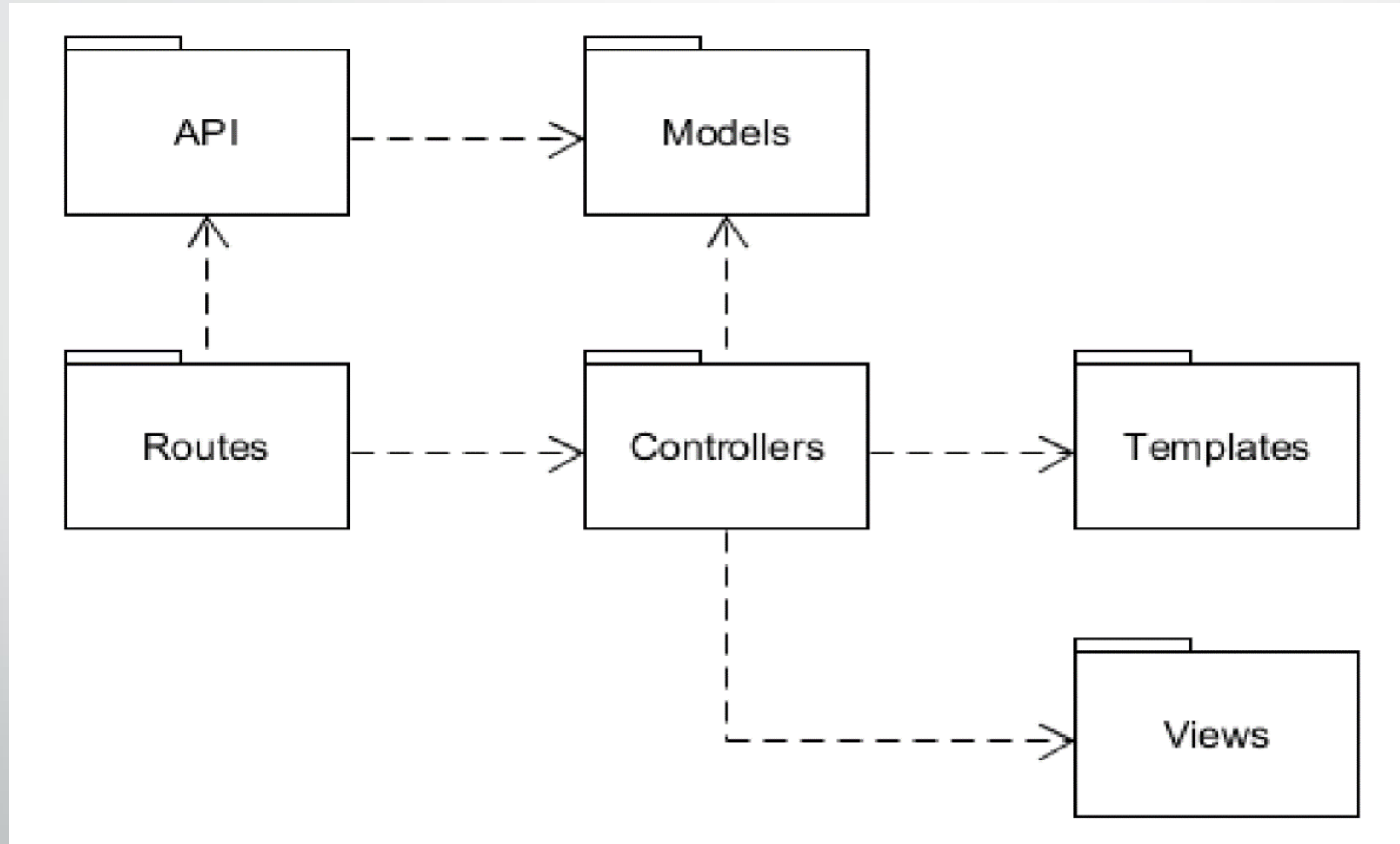


Автоматизація розгортання та управління додатками в Docker

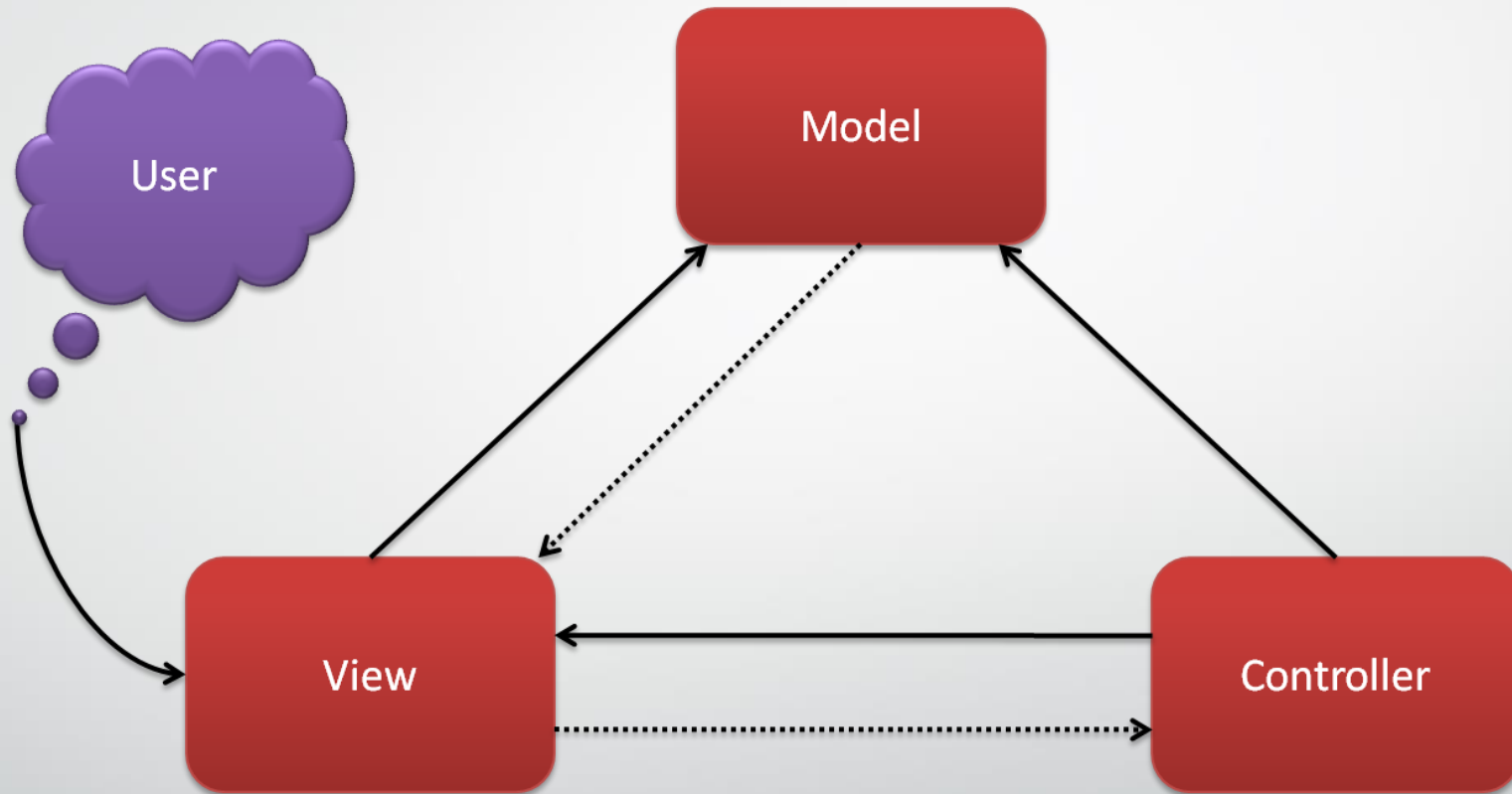
Docker Components



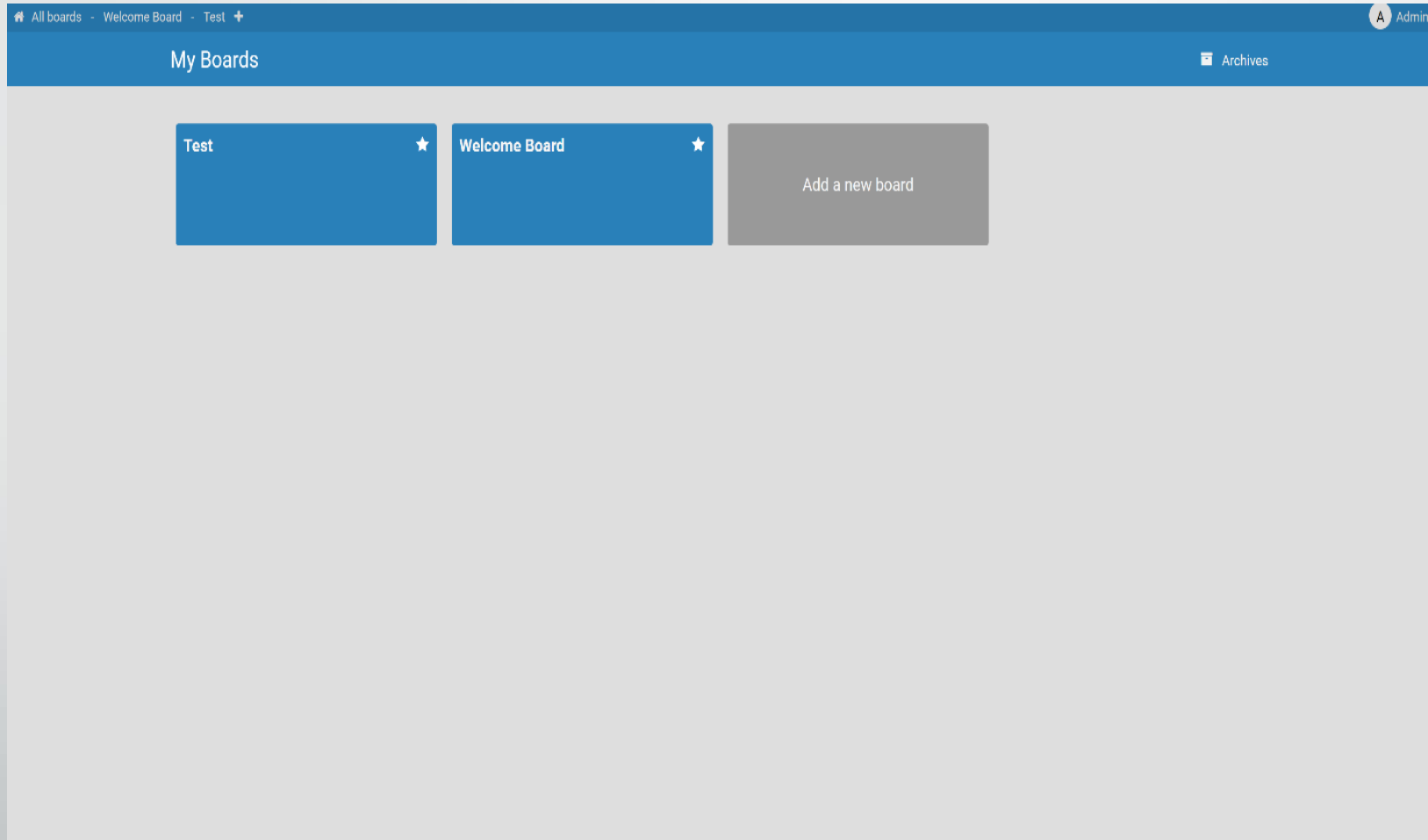
Діаграма пакетів серверної частини веб-додатку



Структура візуального шару



Головне вікно розробленої системи



Kanban-дoшкa

The screenshot displays a Kanban board interface. At the top, a blue header bar contains navigation links: "All boards", "Test Board", and "Test". On the right of the header is a user profile icon labeled "Admin". Below the header, the board title "Test Board" is shown with icons for "Private" and "Tracking". The main workspace is divided into two columns: "New List1" and "New List2".

- New List1** contains two cards: "qwe" (with a green status indicator) and "Ticket2" (with an orange status indicator). Both cards have an "A" icon in the bottom right corner. Below the cards is a "+ Add a card" button.
- New List2** contains one card: "Ticket1" (with a green status indicator) and an "A" icon. Below the card is a "+ Add a card" button.

To the right of the columns is a "+ Add a list" button. On the far right, a sidebar contains three sections:

- Members**: Shows a user icon "A" with a "+ Add member" button.
- Labels**: Shows a "test" label with a green background, followed by four unlabeled color swatches (yellow, orange, purple, blue), and a "+ Add label" button.
- Activities**: A list of recent actions:
 - Admin joined Ticket2. a few seconds ago
 - Admin added Ticket2 to this board. a few seconds ago
 - Admin removed Basics from this board. a minute ago
 - Admin joined Ticket1. a minute ago
 - Admin added Ticket1 to this board. a minute ago
 - Admin added New List2 to this board. 2 minutes ago
 - Admin moved qwe from undefined to New List1. 12 days ago
 - Admin moved qwe from New List1 to undefined. 12 days ago
 - Admin moved qwe from undefined to New List1. 12 days ago

At the bottom of the sidebar is a "Keyboard shortcuts" link.

Картка в розгорнутому вигляді

**** Required Custom Fields - Customer**

in list Live (Mar 26 - 31, 2012)

24

4/5

Mar 28 at 12:00 PM (past due)

Ticket System

- [Spec](#)
- Branch: JN_customfieldrequiredness
- [Pull Request](#)

[Edit the card description.](#)

Post-launch checklist

0%

☐

review use of required custom fields for accounts with high widget-ticket volume

[Add item](#)

Pre-sprint checklist

100%

☒ ~~Wireframe Settings-dialog~~

☒ ~~Error notice on contact form (use standard validation error message)~~

☒ ~~JR: add screenshots (wireframes) for each requirement~~

☒ ~~What do required fields look like on the contact form?~~

[Add item](#)

Labels

[Change](#)

Deployed (Production)

Members

[Assign...](#)

Attachments

No attachments.

[Upload](#)

Actions

[Vote](#)

[Add checklist...](#)

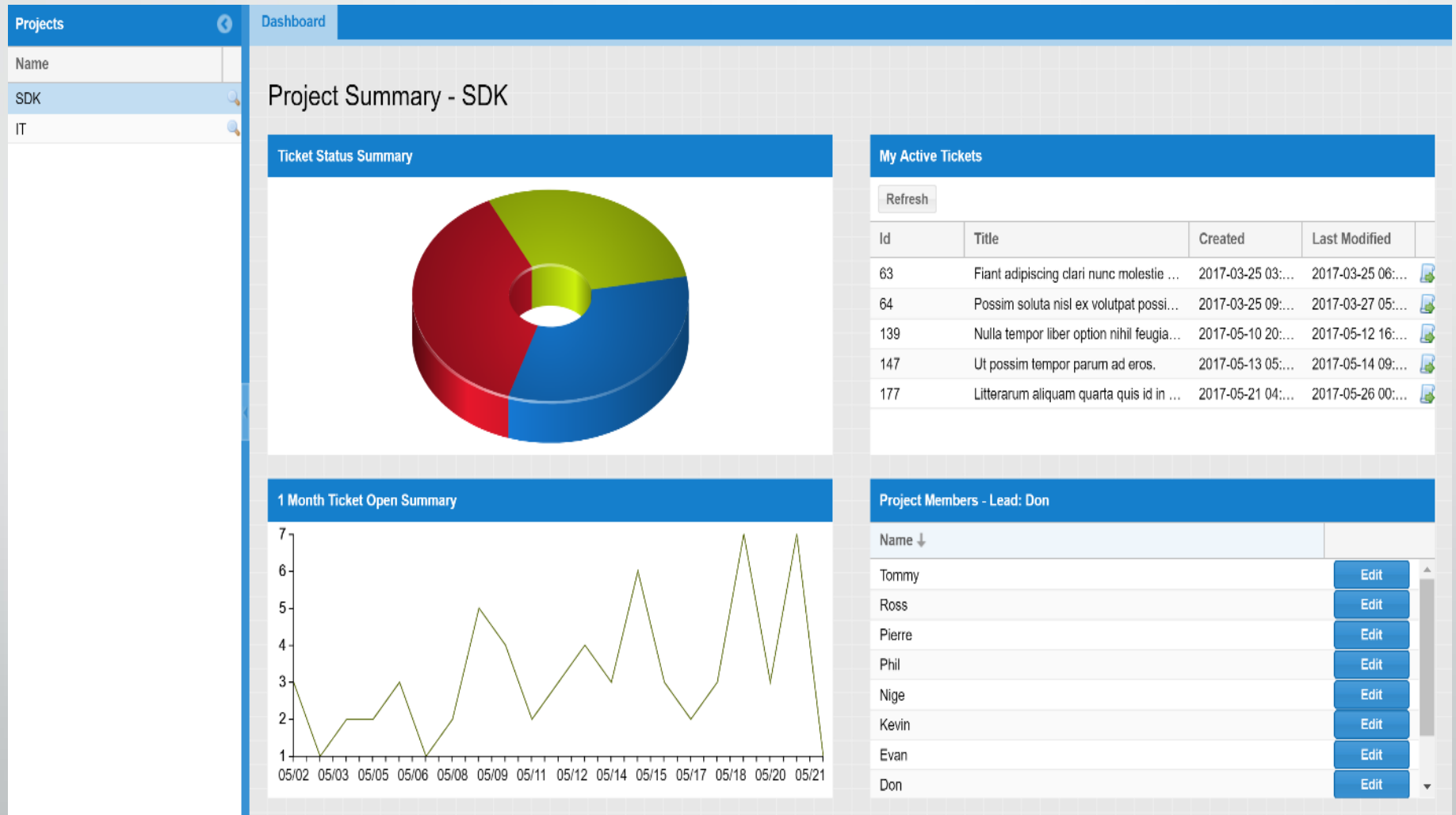
[Due date...](#)

[Labels...](#)

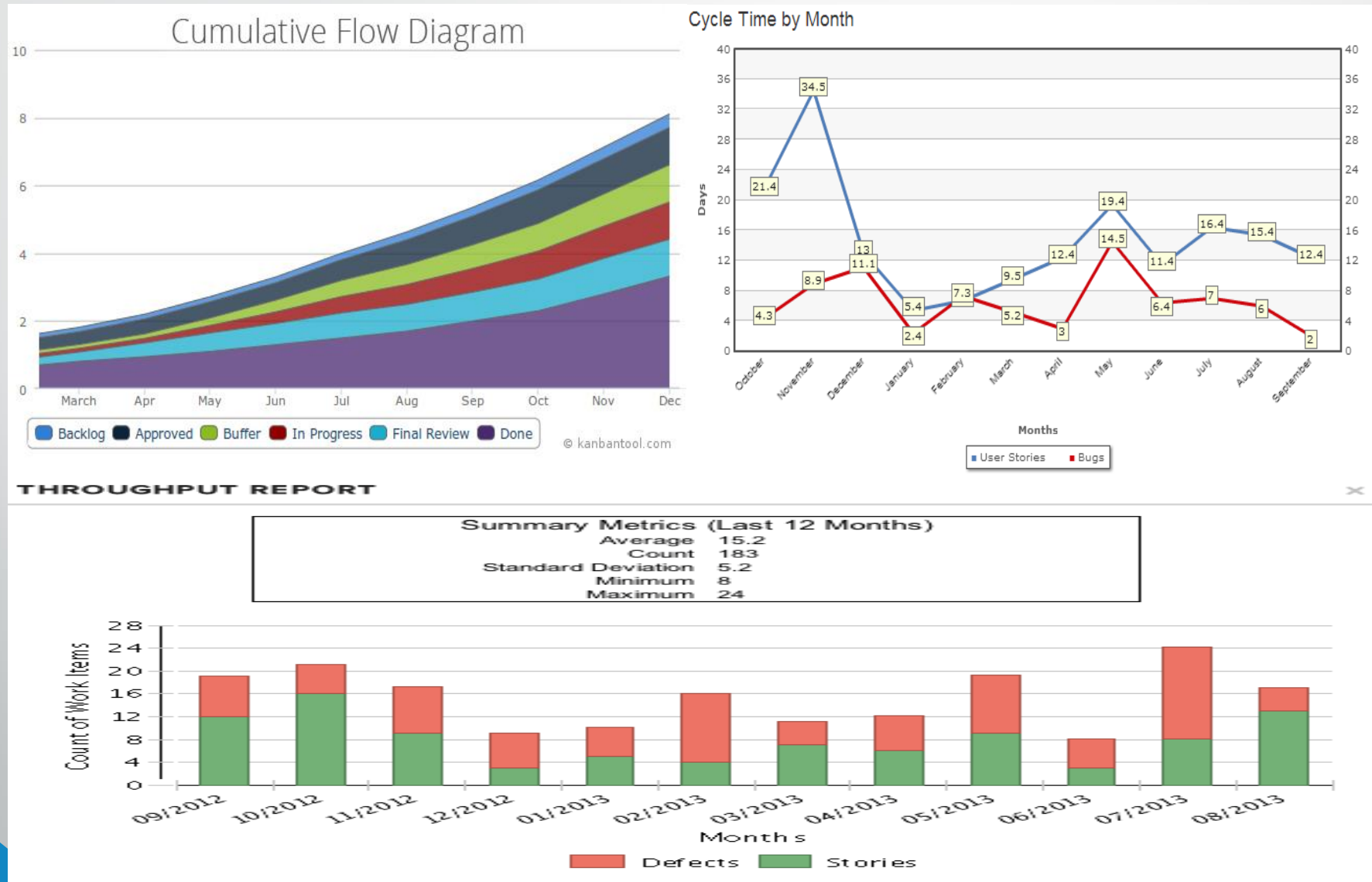
[Move...](#)

[Archive](#)

Інформаційний дашборд



Аналітичний модуль системи



Висновки

- запропоновано сучасні метрики моніторингу стану проекту;
- розроблено метод формування проектних команд для тимчасових віртуальних структур, що використовує теорію нейронних мереж;
- спроектовано та розроблено загальну архітектуру системи;
- розроблено гнучку базу даних;
- впроваджено систему автоматичного розгортання та управління додатками Docker;
- створено серверну частину додатку на програмній платформі Node.js;
- система впроваджена в міжнародній компанії NetCracker.



Дякую за увагу!