

Міністерство освіти і науки України
Національний університет “Запорізька політехніка”

МЕТОДИЧНІ ВКАЗІВКИ
до лабораторних робіт
з дисципліни
“Менеджмент проектів програмного забезпечення”
для студентів
спеціальності 121 “Програмна інженерія”
(всіх форм навчання)

Методичні вказівки до лабораторних робіт з дисципліни
“Менеджмент проектів програмного забезпечення” для студентів
спеціальності 121 “Програмна інженерія” (всіх форм навчання) / Укл.:
Т.В. Федорончак. – Запоріжжя: НУЗП, 2021. – 113 с.

Автори: Т.В. Федорончак, к.т.н., доцент

Рецензент: Н.О.Миронова, к.т.н., доцент

Відповідальний
за випуск: В.І. Дубровін, к.т.н., професор

Затверджено
на засіданні кафедри
програмних засобів

Протокол №
від " " 20 р.

ЗМІСТ

Лабораторна робота № 1	Організація комунікації членів проектної команди за допомогою корпоративного месенджера.....	5
Лабораторна робота № 2	Контроль робочого часу, витраченого на виконання проектних завдань.....	8
Лабораторна робота № 3	Робота з системою управління завданнями при розробці програмного забезпечення.....	13
Лабораторна робота № 4	Управління завданнями проекту за допомогою канбан-дошок.....	29
Лабораторна робота № 5	Прототипування інтерфейсів програмних систем.....	42
Лабораторна робота № 6	Побудова ментальних карт.....	46
Лабораторна робота № 7	Методи побудови мережеских моделей та діаграм передування для ІТ-проектів.....	48
Лабораторна робота № 8	Розрахунок тривалості проекту методом критичного шляху CPM та методом PERT.....	54
Лабораторна робота № 9	Створення проекту та календарне планування робіт.....	63
Лабораторна робота № 10	Планування ресурсів, створення призначень і вартісний аналіз проекту.....	76
Лабораторна робота № 11	Аналіз завантаження ресурсів і оптимізація параметрів проекту.....	90
Лабораторна робота № 12	Контроль за ходом реалізації проекту методом освоєного обсягу.....	95

ВСТУП

Мета дисципліни “Менеджмент проектів програмного забезпечення” – показати проектний підхід до розробки програмного забезпечення та навчити студентів використовувати на практиці основні методи, інструменти та засоби управління проектами.

Внаслідок виконання лабораторних робіт з дисципліни студенти повинні вміти: розробляти загальний план проекту, що вимагає значного об’єму робіт; оцінювати проектні витрати; вимірювати прогрес проекту; використовувати метод засвоєного об’єму для аналізу вартості; динамічно регулювати плани проекту; використовувати засоби керування змінами та системи призначення завдань в проектах з розробки програмного забезпечення, відстежувати затрачений на виконання завдань час, створювати інтелект-карти для відображення вимог проекту та вайрфрейми для візуалізації інтерфейсу користувача.

Студент повинний вміти правильно аналізувати отримані результати. Для самоперевірки при підготовці до виконання і здачі роботи студент повинний відповісти на контрольні питання, приведені наприкінці відповідної роботи.

Звіт з лабораторних робіт повинен містити результати виконання роботи, відповіді на 3 контрольних запитання та змістовний аналіз отриманих результатів та висновки.

Під час співбесіди студент повинний виявити знання про мету роботи, по теоретичному матеріалу, про методи виконання кожного етапу роботи, по змісту основних розділів розробленого звіту з демонстрацією результатів на конкретних прикладах.

Для виконання лабораторних робіт рекомендується використовувати безкоштовну систему призначення завдань Redmine, систему управління проектами за методом канбан-дошок Trello, корпоративний месенджер Slack, систему трекінгу часу Toggle, систему для створення вайрфреймів Balsamic, систему для побудови інтелект-карт XMind та систему управління проектами ProjectLibre або LibrePlan. За власним бажанням студент може використовувати будь-які інші системи керування проектами, що підтримують необхідні можливості.

Загальний залік з практичної частини курсу студент одержує після виконання і здачі всіх лабораторних робіт.

Лабораторна робота № 1

Організація комунікації членів проектної команди за допомогою корпоративного месенджера

Мета роботи

Одержання практичних навичок організації комунікації членів проектної команди.

Основні теоретичні відомості

Slack [1] – це корпоративний месенджер та інструмент для ефективної командної роботи. Його було запущено в 2013 році.

Slack – це цифровий робочий простір команди, що дозволяє організувати ефективну комунікацію членів команди. Slack збирає в одному вікні обговорення в загальних темах (каналах), приватних групах і особистих повідомленнях. Slack надає можливість виконувати швидкий пошук за всіма повідомленнями. Повідомлення в Slack формують архів роботи команди. Крім того Slack надає можливості інтеграції з іншими застосунками, що використовуються в проектах з розробки програмного забезпечення. Так можна інтегрувати в Slack сервіси Google Drive, GitHub, Trello та багато інших. Це дозволяє відстежувати різні сторони проекту за допомогою єдиної платформи без переключення з одного застосунку в інший.

Головне вікно системи відображає основні компоненти (рис. 1.1):

- команди та їх робочі середовища;
- канали;
- повідомлення;
- пошук;
- сповіщення.

Команда – це група людей, які використовують Slack для спілкування. Робоче середовище Slack – це цифровий простір, який команда розділяє для спілкування та отримання роботи.

Робоче середовище Slack складається з каналів, які використовуються для бесід з учасниками. Вони можуть бути організовані навколо відділів компанії, проектів або якихось спільних інтересів. Кількість каналів не обмежена. Кожне робоче середовище містить два канали за замовчуванням: #general і #random. Всі члени

команди входять в ці канали, призначені для об'яв та повідомлень для всієї команди.

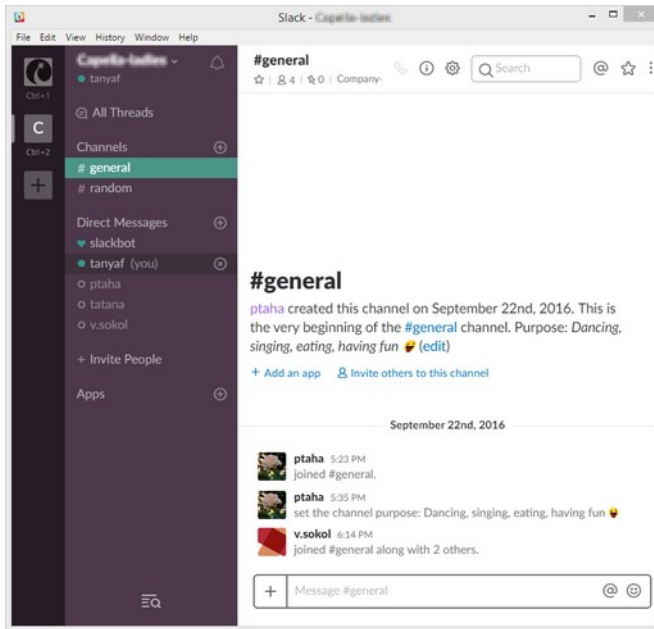


Рисунок 1.1 – Головне вікно системи Slack

В Slack існують два види каналів:

- відкриті (public) доступні всім членам команди;
- закриті (private) доступні окремим членам команди.

Для швидкого спілкування з окремими членами команди, можна надіслати їм прямі повідомлення. Прямі повідомлення можуть включати лише двох співбесідників або групу до 8 людей. Щоб звернути окрему увагу користувача на повідомлення, використовуються згадки за допомогою символу @.

Користувачі можуть ділитися повідомленнями, постами, сніппетами коду, посиланнями, файлами тощо.

Сповіщення використовуються для того, щоб звернути увагу користувача. Сповіщення можна отримувати на робочому столі, на мобільному пристрої або електронною поштою. За допомогою налаштувань, користувач може визначити пріоритети важливості різних типів сповіщень.

Для того, щоб знайти необхідні повідомлення, файли, пости або членів команди, в Slack використовується пошук.

Голосові та відеодзвінки в Slack допомагають команді підтримувати зв'язок між користувачами незалежно від їхнього фізичного розташування. Під час дзвінка можна демонструвати свій робочий стіл для необхідних пояснень. Дзвінки роблять можливою віддалену роботу членів команди.

Slack також надає файловий простір для завантаження файлів команди.

Завдання на лабораторну роботу

1. Створіть команду з 3-5 студентів групи в Slack.
2. Створіть канали для обговорення.
3. Інтегруйте робоче середовище з Google Drive, Trello та GitHub. Продемонструйте результати інтеграції.
4. Підготуйте звіт з лабораторної роботи.

Контрольні питання

1. Поясніть призначення та принципи роботи з системою Slack. Які в неї можливості?
2. Що таке канал? Яких типів вони бувають?
3. Як звернутися в чатах до користувачів?
4. Які можливості налаштування сповіщень в системі Slack?
5. З якими сервісами можна інтегрувати Slack? Навіщо потрібна інтеграція?
6. Що таке боти в Slack?
7. Яким чином налаштувати нагадування в Slack за допомогою команди /remind?
8. Що таке слеш команди? Наведіть приклади.
9. Назвіть інші програмні засоби, що використовуються для організації комунікації всередині команд з розробки програмного забезпечення.

Лабораторна робота № 2

Контроль робочого часу, витраченого на виконання проектних завдань

Мета роботи

Одержання практичних навичок контролю робочого часу, витраченого на виконання проектних завдань.

Основні теоретичні відомості

Для вирішення проблеми контролю робочого часу за завданнями проектів створено безліч програм обліку робочого часу або тайм-трекерів. Ці системи дозволяють визначити реально витрачений час на виконання робіт проекту, а також визначити на які саме роботи витрачає свій час команда. Даний тип програмного забезпечення можуть використовувати люди з погодинною оплатою праці. Це можуть бути такі професії, як програмісти, дизайнери, перекладачі, адвокати. Трекери можуть мати підтримку командної роботи і можуть використовуватися як інструмент контролю з боку керівника проекту за завданнями, що виконуються співробітниками.

Найпростіші тайм-трекери відзначають кількість часу, витраченого на певні види робіт. Більш складні системи збирають також інформація про відвідані сайти, запущені програми, створюють скріншоти робочого столу, відстежують прості системи, створюють звіти і виставляють рахунки.

Toggl

Toggl [2] – це один з найпопулярніших сервісів в своїй категорії. Він має практично всі необхідні можливості, щоб організувати роботу фрілансера або навіть невеликої компанії. Більшість функцій сервісу доступні навіть у безкоштовній версії. Облік робочого часу може виконуватися в браузері, в настільному застосунку та на мобільному пристрої.

Основні функції системи:

- ручний і автоматичний облік робочого часу;
- функція побудови звітів, які показують прогрес і витрати часу;

- можливість поділу команди на групи;
- таймери працюють в автономному режимі і синхронізуються, коли ви знову в інтернеті;
- таймери синхронізуються на всіх підключених пристроях, так що ви можете почати роботу на вашому комп'ютері і продовжити на телефоні.

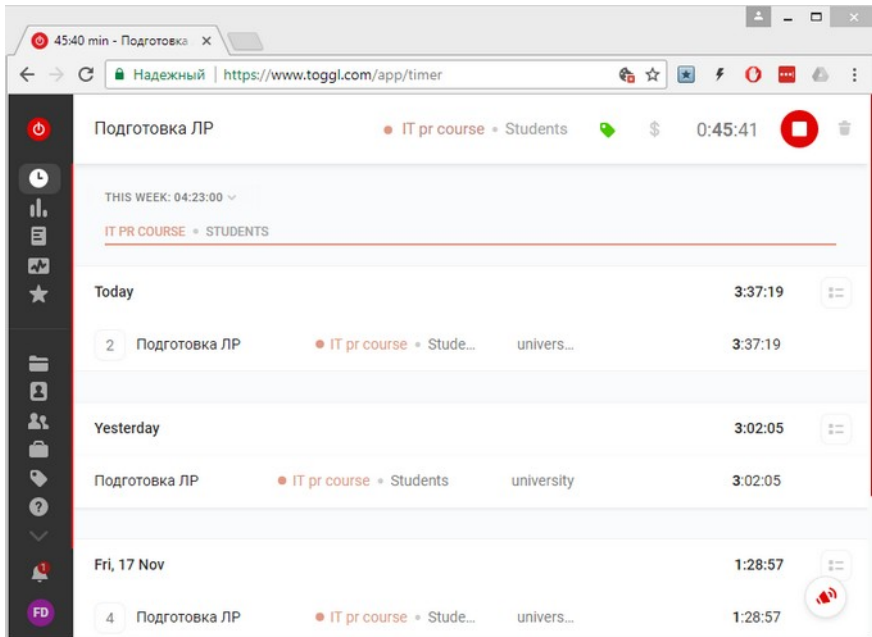


Рисунок 2.1 – Головне вікно системи Toggl в браузері

За допомогою настільного застосунку можна отримати додаткові можливості:

- Timeline – облік переглянутих веб-сайтів та використаних під час роботи програм, розподілених по 15 хвилинним інтервалам; інформація зберігається протягом 6 днів;
- Autotracker – дозволяє визначити програми або ключові слова та проекти, з якими вони пов'язані; якщо програма з ключовим словом знаходиться на передньому плані, вона запускає активатор автоматичного трекеру;
- Idle detection – сповіщає про відсутність активності під час

контролю робочого часу та дозволяє зафіксувати час простою, відхилити його або додати час як новий запис;

– Pomodoro timer – дозволяє налаштувати сповіщення, які дадуть знати про необхідність зробити перерву в роботі.

В платних версіях продукту додаються можливості попереднього поділу проекту на завдання, підрахунку вартості виконаної роботи та розширені форми звітування.

В системі існує поняття робочого простору. Це дозволяє розділити робочі та приватні дані.

Спочатку в системі необхідно створити проекти та клієнтів, для того щоб мати можливість розподіляти витрачений на завдання час за проектами. Додатково можна створити теги, щоб розподіляти записи по категоріях (наприклад, проектування, розробка, тестування, фідбек, написання документації тощо). Це дозволяє легше організовувати та шукати записи в системі.

Після цього можна створювати записи з обліку. Для цього задається назва виконуваного завдання, проект, мітки та запускається таймер. Після закінчення роботи таймер необхідно зупинити. Далі можна створювати нові завдання або запускати таймери для вже існуючих в системі завдань. При цьому при кожному запуску та зупинці таймеру в системі буде додаватися новий запис обліку часу. На випадок, коли ви забули запустити таймер при виконанні роботи, в системі можна додати запис вручну.

На сторінці статистики можна побачити розподіл витраченого часу згрупований за клієнтами, проектами, командами, завданнями, за календарем. Дані статистики можуть бути експортовані із системи.

Завершені або неактивні проекти можна перенести в архів.

Ви можете інтегрувати таймер Toggl з іншими веб-інструментами, з якими ви працюєте, за допомогою Toggl Button Chrome Extension (Asana, Google Диск, Trello, Basecamp). Для Trello це розширення додає в картках кнопку початку роботи таймера, по натисканню якого створюється запис обліку часу в системі Toggl з ім'ям картки.

TopTracker

В якості прикладу системи відстеження часу з більшими можливостями розглянемо TopTracker.

TopTracker [3] – це легкий застосунок, який допомагає вам стежити за тим, як ви витрачаєте свій час. Користувачі можуть включити відстеження для власних проєктів або проєктів, на які вони запрошені іншими членами команди.

Основне відстеження часу можна зробити за допомогою веб-застосунку, проте рекомендується завантажити настільну програму, щоб активувати відстеження рівня активності, описи активності, скріншоти чи знімки веб-камери.

TopTracker – це часовий трекер, спеціально розроблений для фрілансерів та їхніх потреб. Він був ретельно спроектований, щоб надавати фрілансерові повний контроль над тим, що відстежується, і як часто.

Програма відстежує час, витрачений на кожне завдання, і записує, коли відбувається клацання клавіатури або клацання миші, що дозволяє автоматично відслідковувати неактивність користувача і надавати інформацію про основні статистичні показники діяльності. Зауважте, що він ніколи не фіксує щось, що ви насправді набираєте або клацаєте. Застосунок також може за необхідності записувати знімки екрана або знімки веб-камери.

Якщо ви працюєте з конфіденційною інформацією, перед завантаженням можна налаштувати автоматичне розмиття всіх знімків екрана, щоб зображення розпізнавалось, однак текст не можна було читати. Таким чином, члени вашої команди можуть отримати уявлення про те, над чим ви працюєте, не порушуючи безпеку будь-якої конфіденційної інформації.

У застосунку для настільних комп'ютерів ви також можете видалити скріншоти перед їх завантаженням або просто відключити відстеження скріншотів взагалі.

Завдання на лабораторну роботу

1. Зареєструйтеся в системі Toggl та продемонструйте роботу з системою.
2. Відстежте час на виконання однієї з лабораторних робіт.

Контрольні питання

1. Навіщо необхідні системи обліку та контролю робочого часу?

2. Які функції повинні мати системи обліку робочого часу?
3. Опишіть техніку управління часом Pomodoro.
4. Опишіть можливості системи Toggl.
5. Опишіть можливості системи TopTracker.
6. Поясніть необхідність інтеграції систем контролю робочого часу з іншими системами управління проектами.
7. Назвіть інші існуючі системи тайм-трекінгу.

Лабораторна робота № 3

Робота з системою управління завданнями при розробці програмного забезпечення

Мета роботи

Одержання практичних навичок роботи з системою управління завданнями з інтеграцією з системою керування версіями коду.

Основні теоретичні відомості

Redmine [4] – це відкритий серверний веб-застосунок для управління проектами та завданнями (в тому числі для відстеження помилок). Redmine написаний на Ruby на основі веб-фреймворку Ruby on Rails.

Даний продукт надає наступні можливості:

- ведення декількох проектів;
- гнучка система доступу, заснована на ролях;
- система призначення та відстеження завдань;
- діаграми Гантта і календар;
- ведення новин проекту, документів і управління файлами;
- оповіщення про зміни за допомогою RSS-потоків і електронної пошти;
- вікі та форуми для кожного проекту;
- облік часових витрат;
- інтеграція з системами контролю версій коду;
- створення записів про помилки на основі отриманих листів.

Redmine створює окремий простір для кожного проекту. Це простір описує стан проекту (призначені завдання, учасники проекту, останні новини та ін.) і надає доступ до засобів управління проектом.

Redmine використовує термін *issue* (проблема, питання) для позначення завдання розробки.

Існує кілька стандартних *категорій* для завдань (*трекерів*):

- Feature – нова функція;
- Bug – помилка, невірна робота існуючого коду;
- Support – підтримка.

Адміністратор системи може додати нові категорії завдань під конкретний проект.

Користувачам системи призначаються ролі, які визначають, що

кожен користувач може зробити в рамках кожного проекту.

Наведемо стандартні ролі системи в порядку убування їх привілеїв:

- адміністратор сайту управляє сайтом і обліковими записами користувачів, налаштовує дозволи для ролей тощо;
- менеджер проекту може створювати нові проекти, управляти простором проекту (наприклад, управляти учасниками проекту і їхніми ролями), управляти документами проекту, новинами, вікі і форумом;
- розробник управляє і редагує завдання, виконує фіксацію змін в репозиторій, редагує вікі, управляє файлами проекту, змінює репозиторій;
- генератор звітів створює і переглядає завдання, протоколює час витрачений на завдання, переглядає репозиторій, переглядає вікі, бере участь в обговореннях на форумі.

Тільки адміністратор системи має доступ до глобальних налаштувань системи, в тому числі до активації модулів системи, таких як наприклад, репозиторії та їх налаштування.

Користувачам призначається роль в кожному проекті, в якому він бере участь, наприклад, «менеджер в проекті А», «розробник в проекті Б» або «клієнт в проекті В». Користувач може мати кілька ролей. Призначення ролі для окремого завдання в даний момент неможливо.

Потік робіт (workflow) – це зміна статусів завдань в системі, це те як Redmine відстежує завдання від створення до закінчення виконання. За замовчуванням послідовність дій над завданням включає наступні статуси (рис. 3.1):

- New – нова;
- In Progress – в роботі;
- Feedback – зворотний зв'язок;
- Resolved – вирішена;
- Closed – закрита;
- Rejected – відхилена.

За замовчуванням, будь-хто, хто увійшов в систему, може створити завдання, але тільки Менеджер може видалити, відхилити або відкрити заново закриті завдання.

В налаштуваннях системи можна налаштувати правила переходу завдань з одного статусу в інший.

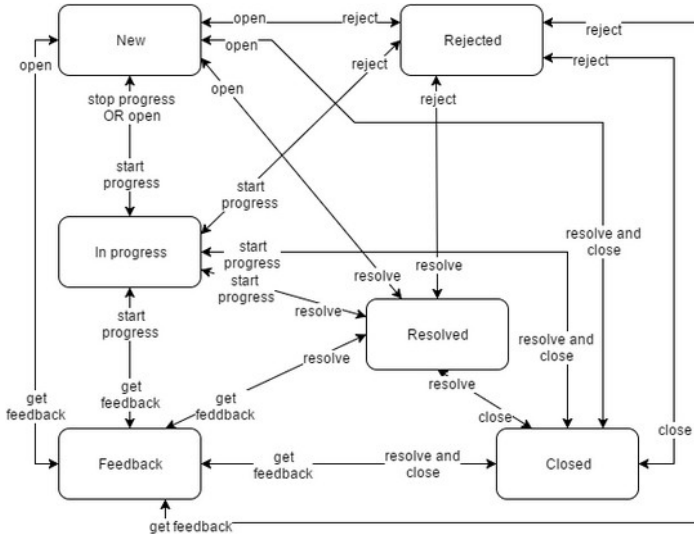


Рисунок 3.1 – Стандартний потік робіт системи Redmine

Проект

Список проектів (з можливістю додати новий) доступний з верхньої панелі меню (пункт Projects). За замовчуванням, створювати проекти в системі можуть тільки адміністратори і менеджери проектів.

При створенні проекту (рис. 3.2) необхідно вказати ім'я та унікальний ідентифікатор, який буде використаний як частина адреси url простору проекту, опис та домашню сторінку проекту. Проект може бути створений у вигляді підпроекту вже існуючого проекту.

Після створення в налаштуваннях проекту необхідно вказати виконавців проекту (рис. 3.3). Хоча б один користувач повинен бути призначений в якості менеджера проекту.

Простір обраного проекту (рис. 3.3) відображає його назву у верхній частині і містить панель вкладок, яка надає доступ до наступних пунктів для управління проектом і його підтримки:

- Overview – огляд проекту (відстеження завдань, учасники проекту, останні новини, витрачений час);
- Activity – останні активності в проекті;
- Issues – завдання проекту;

- Gantt – діаграма Гантта проекту;
- Calender – календар;
- News – новини, анонсовані менеджером проекту, або важлива загальна інформація про статус проекту;
- Documents – робочі документи (документація, специфікації, замітки, інструкції);
- Wiki – вікі для опису проекту, підтримує історію;
- Files – файли, що використовуються в розробці (наприклад, для скачування членами проекту або файли проекту в вигляді релізу);
- Forums – форум для обговорення проекту його учасниками;
- Settings – налаштування.

Redmine

Home My page Projects Administration Help

Logged in as user My account Sign out

Search: Jump to a project

New project

Name * test project

Description

Lets test redmine and the way it works.

Identifier * test-project

Length between 1 and 100 characters. Only lower case letters (a-z), numbers, dashes and underscores are allowed, must start with a lower case letter.
Once saved, the identifier cannot be changed.

Homepage

Public ☒

Subproject of

Inherit members ☐

Modules

☒ Issue tracking ☒ Time tracking ☒ News

☒ Documents ☒ Files ☒ Wiki

☒ Repository ☒ Forums ☒ Calendar

☒ Gantt

Trackers

☒ Bug ☒ Feature ☒ Support

Create Create and continue

Рисунок 3.2 – Створення проекту

Redmine

Home My page Projects Administration Help

Logged in as user My account Sign out

Search: Jump to a project

Test project

+ Overview Activity Issues Gantt Calendar News Documents Wiki Files Settings

Settings

Information Modules **Members** Versions Issue categories Wiki Repositories Forums Activities (time tracking)

New member

User / Group	Roles	
user1 user1	Manager	Edit Delete
user2 user2	Developer	Edit Delete
user3 user3	Reporter	Edit Delete

Рисунок 3.3 – Додання виконавців створеного проекту

Завдання

Завдання є центральним поняттям всієї системи, що описує якусь роботу, яку необхідно виконати, або проблему, яку необхідно вирішити. Завдання прив'язане до певного проекту, належить певному користувачеві, може бути пов'язана з певною версією тощо.

Створюване завдання призначається на одного з членів проектної команди. Робота, що проводиться над завданням, можуть відстежувати члени команди, яких додано в спостерігачі.

Форма створення завдання наведена на рис. 3.4.

The screenshot shows the 'New issue' form in the Redmine application. The interface is in a blue-themed layout. At the top, there's a navigation bar with links like 'Home', 'My page', 'Projects', 'Administration', and 'Help'. Below this is a search bar and a 'Jump to a project' dropdown. The main content area has a tabbed interface with 'Overview', 'Activity', 'Issues', 'Gantt', 'Calendar', 'News', 'Documents', 'Wiki', 'Files', and 'Settings'. The 'Issues' tab is active, showing the 'New issue' form. The form includes fields for 'Tracker' (set to 'Feature'), 'Subject' (set to 'Feature number one'), and a large 'Description' text area containing the text 'First feature task in redmine test project'. Below these are fields for 'Status' (set to 'New'), 'Priority' (set to 'Normal'), and 'Assignee' (set to 'user1 user1'). There are also fields for 'Parent task', 'Start date' (09.09.2016), 'Due date' (15.09.2016), 'Estimated time' (10 hours), and '% Done' (0%). A 'Files' section allows uploading files (maximum size 5 MB). A 'Watchers' section lists 'user1 user1', 'user2 user2', and a search bar for adding more watchers. At the bottom, there are buttons for 'Create', 'Create and continue', and 'Preview'. The footer indicates the application is powered by 'Streamline Redmine Stack' and includes copyright information for 2006-2016.

Рисунок 3.4 – Створення завдання

При створенні завдання вказується наступна інформація:

- Tracker – категорія завдання;
- Subject – короткий опис завдання;
- Description – детальний опис завдання;
- Status – стан завдання (зазвичай нове завдання має статус new і по мірі виконання роботи над завданням статус буде змінюватися);
- Priority – пріоритет завдання (low, normal, high, urgent, immediate);
- Assignee – учасник проекту, на якого призначається завдання;

- Start date и due date – планові дати початку і завершення завдання ;
- Estimated time – оцінка часу виконання;
- Files – файли з відомостями щодо завдання, наприклад, скріншоти екрану;
- Watchers – учасники проекту, які будуть отримувати оповіщення про зміну статусу завдання.

Існуючі завдання відображаються списком на вкладці Issues.

Коротке зведення (summary) відображає статистику про завдання в різних блоках по: трекерам, пріоритетах, користувачам, на яких призначені завдання, авторам, версіями, категоріям. Кожен з блоків може бути обраний для більш деталізованого перегляду. В цьому випадку будуть відображені не тільки завдання в станах закрито / відкрито / всього, але і по всім іншим налаштованим статусам завдань.

На сторінці перегляду завдання (рис.3.5) відображається поточний стан роботи, відомості про те, що вже зроблено в процесі виконання завдання в хронологічному порядку, а також надаються інструменти для управління завданням.

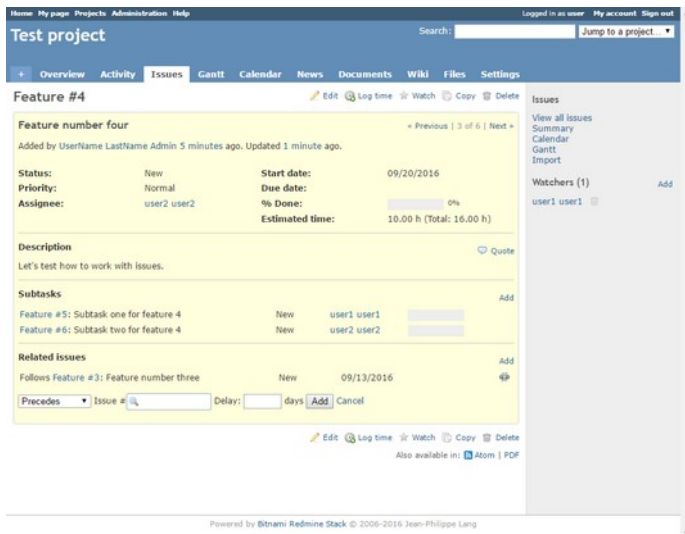


Рисунок 3.5 – Сторінка перегляду завдання

Завдання може бути розбито на підлеглі завдання (subtasks), що дозволяє регулювати відношення батько-нащадок між завданнями.

Для батьківського завдання дата початку – найраніша з дат початку підлеглих завдань, дата завершення – найпізніша з дат завершення підлеглих завдань, пріоритет – найвищий з пріоритетів підлеглих завдань, оцінка часу виконання – сума оцінок часу підлеглих завдань, витрачений час – сума часу витраченого на підлеглих завдань, відсоток виконання – зважене середнє значення відсотка виконання підлеглих завдань.

Для завдання можна додати пов'язані з ним завдання (Related issues).

Зв'язування завдань дозволяє розробникам позбутися дубльованих завдань шляхом їх об'єднання один з одним, або позначити послідовність вирішення завдань.

При зв'язуванні завдань можна встановити різні варіанти відносин:

- related to / пов'язане з – просто додає посилання на інше завдання;

- duplicates / дублює – пов'язує завдання так, щоб при закритті завдання оригіналу буде закрито і завдання дублікат:

- наприклад, якщо завдання В дублює А, закриття В залишить А відкритою, а закриття А автоматично закритим В;

- duplicated by / дублюється – зворотній зв'язок дублювання;

- blocks / блокує – пов'язує завдання так, що закриття блокованого завдання неможливо, поки не буде закрито завдання-блокіратор:

- наприклад, якщо завдання В блокує завдання А, А не може бути закрито, поки не закрито В;

- blocked by/ блокується – зворотнє від блокує;

- precedes / передує – пов'язує завдання, визначаючи такий порядок їх виконання, де завдання А повинне бути завершене за х днів до того як завдання В повинна бути запущене:

- наприклад, якщо завдання А передує завданню В, ви не зможете встановити дату початку завдання В такою, що дорівнює або менше дати закінчення завдання А;

- follows / іде слідом – зворотнє від передує.

- наприклад, якщо завдання В слідує за завданням А (наприклад А закінчується 21.04, а В починається 22.04) і ви встановлюєте +2 дні до закінчення завдання А, то дати початку і закінчення завдання В зрушити на +2 дні теж ;

– скопійовано з – пов'язує завдання, отримані шляхом копіювання;

— copied to — скопійовано в.

Над завданням можна проводити наступні дії:

– Edit – додання опису прогресу по завданню, відсотка виконання, зміна статусу завдання ;

- Log time – учасник проекту може протоколювати час, витрачений на завдання;

- Watch – учасник проекту може стати спостерігачем завдання.

При натисканні на посилання Edit відображається вікно для редагування відомостей про завдання (рис. 3.6), в якому можна змінити статус, змінити виконавця, відсоток виконання, додати замітки про роботу, файли тощо.

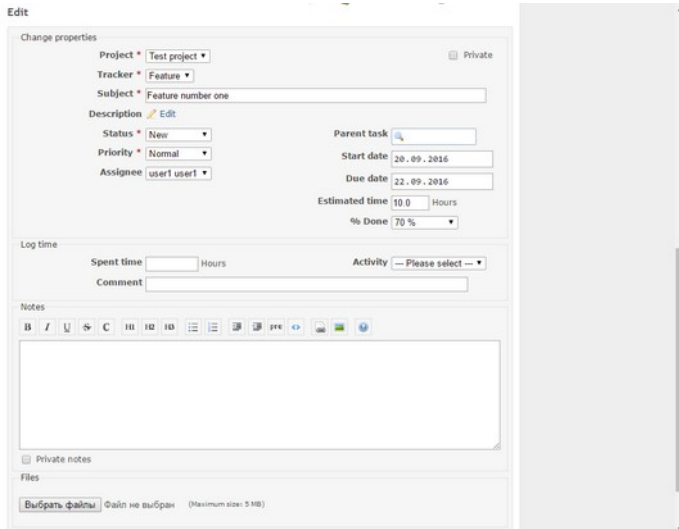


Рисунок 3.6 – Редагування завдання

Персональна сторінка користувача

Персональна сторінка користувача (My page) дозволяє

відобразити різні блоки, які містять інформацію про проекти, в яких він залучений. За замовчуванням активовані два блоки (рис.3.7):

- Issues assigned to me – призначені для виконання завдання ;
- Reported activities – відомості про завдання, на які користувач призначений спостерігачем.

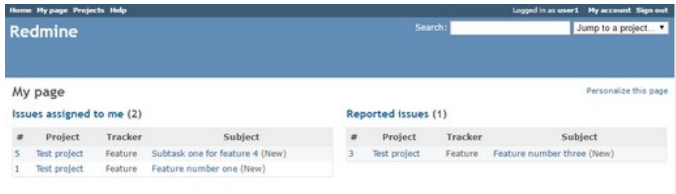


Рисунок 3.7 – Персональна сторінка користувача

Відстеження прогресу

Для відстеження прогресу за завданнями використовуються кілька засобів:

- Calendar – в календарі зазначаються початкова та кінцева дати завдань (рис. 3.8);
- Gantt chart – на діаграмі Гантта відображаються початкові і кінцеві дати завдань, їх статуси, зв'язки і відсоток виконання (рис. 3.9).

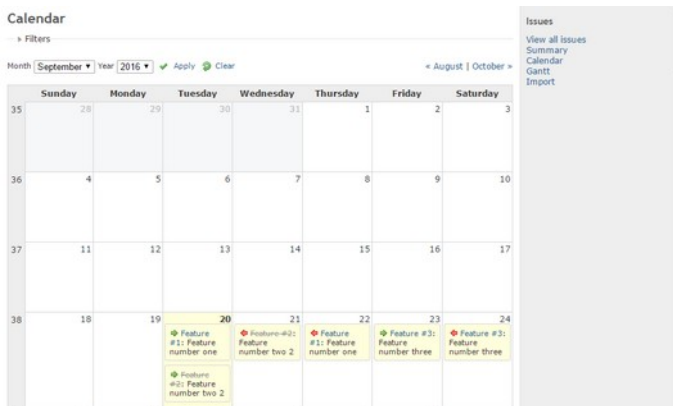


Рисунок 3.8 – Календар проекту

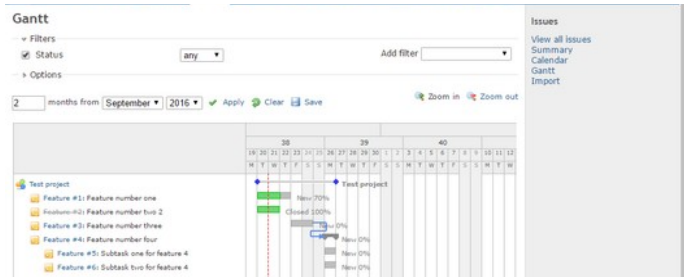


Рисунок 3.9 – Діаграма Гантта

Відстеження часу

Відстеження часу (Log time) дозволяє користувачам вказати, скільки годин роботи витрачено на певне завдання проекту. Кожен запис витраченого часу (рис. 3.10) може бути рознесений за типом діяльності (проектування або розробка) і доповнений коментарями.

Слід зауважити, що витрачений час фіксується в десятковому форматі!

Ці дані можуть бути використані, наприклад, для аналізу внеску кожного учасника в проект або для оцінки фактичної трудомісткості і вартості розробки.

Рисунок 3.10 – Протоколювання витраченого часу

Після створення набору записів про витрачений час сторінка Подрообиці трудовитрат (spent time) (рис. 3.11) відобразить витрачений час на проект (вкладка overview -> бічна панель spent time -> details).

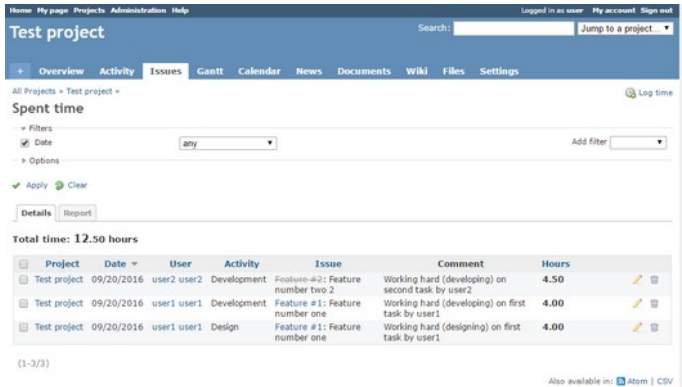


Рисунок 3.11 – Відстеження часу, витраченого на завдання

Оперативний план (Roadmap)

Оперативний план забезпечує засноване на версіях відображення завдань, що гнучко налаштовується і допомагає планувати і управляти розробкою проекту. Він забезпечує згрупований (заснований на версіях) огляд поточного стану проекту.

Пункт меню оперативного плану відображається в меню проекту тільки якщо в проекті включений модуль завдань і через настройки проекту створена хоча б одна версія (рис. 3.12).

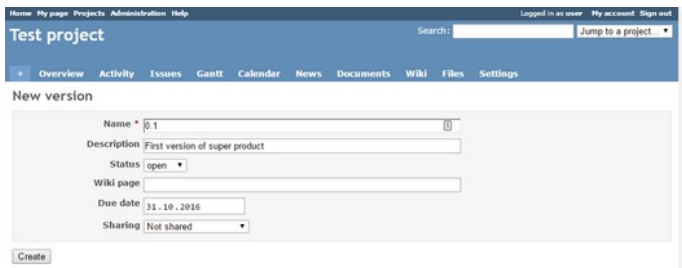


Рисунок 3.12 – Створення версії програмного продукту

Вкладка "Оперативний план" (Roadmap) надає огляд поточного стану вашого проекту, який містить наступні "блоки" інформації для всіх (не завершених) версій (рис. 3.13):

- ім'я версії;
- дата, на яку заплановано завершення версії;

- прогрес бар, що показує загальний відсоток активних/завершених/закритих завдань на основі відсотка готовності, очікуваний час і статуси завдань, що прив'язані до відповідної версії;
- вміст вікі сторінки, що прив'язано до версії;
- список прив'язаних до версії завдань.

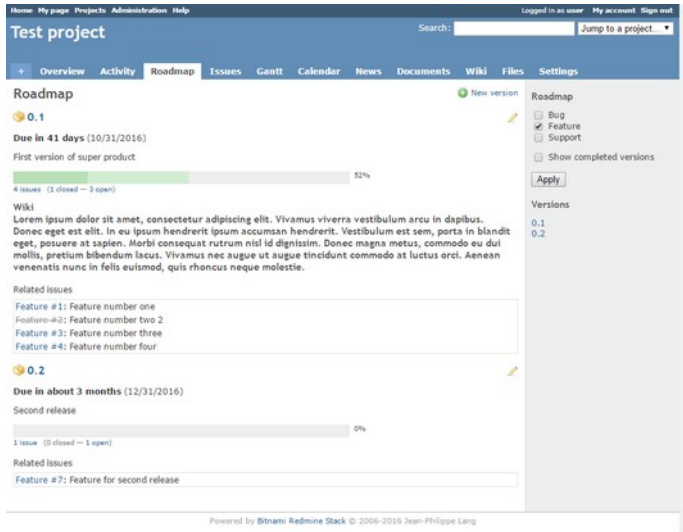


Рисунок 3.13 – Оперативний план проекту

Якщо натиснути на назву версії, відобразиться детальне подання поточного стану конкретної версії (рис. 3.14).

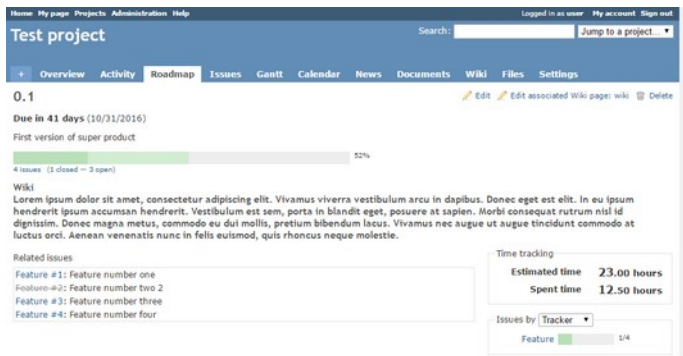


Рисунок 3.14 – Детальний опис версії продукту

Інтеграція з системами контролю версій

Redmine може бути інтегрований з системами контролю версій: Subversion, Darcs, Mercurial, CVS, GIT. Ці системи можуть бути використані в якості сховищ коду і для відстеження змін в коді.

Для роботи з репозиторіями відповідні пакети (SVN, GIT тощо) повинні бути встановлені на тій же машині, що і Redmine. Для Subversion репозиторію підтримуються всі стандартні протоколи (http:, svn:, file: ...), тобто можна просто вказати URL існуючого віддаленого репозиторію. Включити підтримку репозиторіїв може тільки адміністратор системи (рис. 3.15). Інтеграція полягає у відстеженні змін у зовнішньому репозиторії, їх фіксації в базі даних, аналізі коментарів до змін з метою їх прив'язки до певних завдань.

Спочатку необхідно створити новий репозиторій (рис. 3.16).

Для сховища необхідно встановити відповідність користувачів сховища та системи Redmine (рис. 3.17).

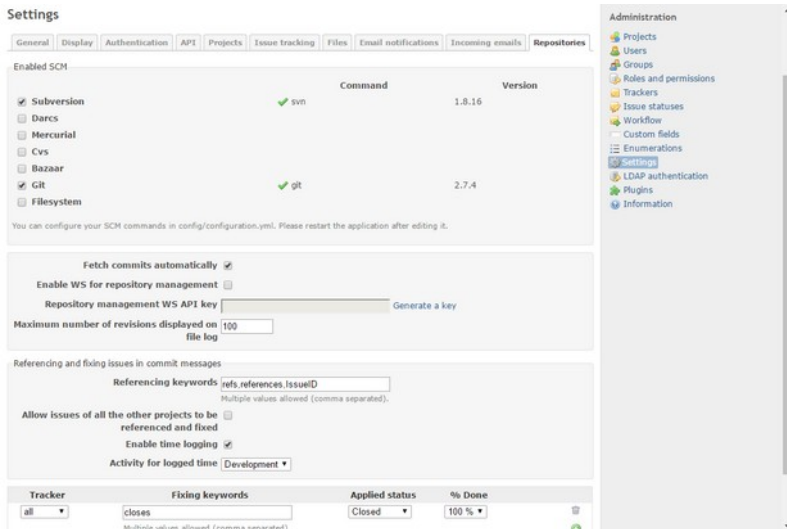


Рисунок 3.15 – Налаштування систем контролю версій коду адміністратором системи

Home My page Projects Administration Help

Test project

Logged in as user My account Sign out

Search:

Overview Activity Roadmap Issues Gantt Calendar News Documents Wiki Files Settings

New repository

SCM Subversion

Main repository ☒

Identifier testrepo

Length between 1 and 255 characters. Only lower case letters (a-z), numbers, dashes and underscores are allowed. Once saved, the identifier cannot be changed.

URL https://github.com/tanyaf/datasciencecourse.git

Examples: file:///, http:///, https://, svn://, svn+[(tunnelscheme)]/

Login tanyaf

Password

Create Cancel

Рисунок 3.16 – Підключення репозиторію

Home My page Projects Administration Help

Test project

Logged in as user My account Sign out

Search:

Overview Activity Roadmap Issues Gantt Calendar News Documents Wiki Files Repository Settings

Repository

Select or update the Redmine user mapped to each username found in the repository log. Users with the same Redmine and repository username or email are automatically mapped.

Login	User
tanyaf	user1 user1

Save

Рисунок 3.17 – Налаштування користувачів репозиторію

Вкладка Сховище (Repository) використовується для огляду сховища проекту і дозволяє переглядати останні фіксації (рис. 3.18).

Home My page Projects Administration Help

Test project

Logged in as user My account Sign out

Search:

Overview Activity Roadmap Issues Gantt Calendar News Documents Wiki Files Repository Settings

testrepo

Name	Size	Revision	Age	Author	Comment
branches		1	over 2 years	tanyaf	hello world added
trunk		4	less than a minute	tanyaf	This commit refs: #1 and fixes #2 Implement fea...

Latest revisions

#	Date	Author	Comment
4	09/20/2016 06:15 PM	tanyaf	This commit #1 and fixes #2 Implement feature #1 #2 redmine test
3	06/15/2014 01:32 PM	tanyaf	Merge branch 'master' of https://github.com/tanyaf/datasciencecourse
1	06/15/2014 01:21 PM	tanyaf	hello world added
2	06/03/2014 10:22 AM	tanyaf	Initial commit

View differences

View all revisions

Also available in: Atom

Рисунок 3.18 – Інтеграція з репозиторієм

При отриманні даних зі сховищ коментарі до фіксацій стану проекту (комітів) проглядаються на наявність ключових слів, що позначають завдання проекту. Це дозволяє автоматично додавати в систему відомості про стан завдань.

Ключові слова за замовчуванням (регістронезалежні):

- для прив'язування до завдання: refs, references, IssueID;
- для завершення завдання: fixes, closes.

В системі за замовчуванням не встановлено статус, який застосовується для автозавершення завдання, тому його необхідно задати. Зробити це може лише адміністратор системи (рис. 3.15).

Приклад коментаря:

`This commit refs #1, #2 and fixes #3`

Після ключового слова ідентифікатори завдань повинні бути розділені пробілом, комою чи символом &.

Якщо встановлений прапорець "включити відстеження часу" (Налаштування -> Репозиторії), розробники можуть відображати час, витрачений на завдання через повідомлення, які супроводжують коміт. В такому випадку необхідно вказати номер завдання і ключову комбінацію @ <time>, яка містить години і хвилини виконання.

Наприклад:

`refs #1234 @15m`

Пов'язані редакції (Associated revisions) будуть відображені на сторінці завдання (в основному праворуч від першої відповіді).

Завдання на лабораторну роботу

1. Створіть git репозиторій. Репозиторій може бути створений на власній машині або в системах Github або Bitbucket.
2. Створіть команду з 3-5 студентів як користувачів в системі Redmine, один з яких буде менеджером проекту, а інші – розробниками.
3. Створіть проект та декілька завдань в ньому. Призначте завдання на виконавців, додайте спостерігача.
4. Створіть версію продукту та додайте до нього завдання.
5. Підключіть репозиторій до проекту та укажіть відповідність користувачів двох систем.
6. Продемонструйте потік робіт в системі. Іншими словами, змініть статуси завдань, наприклад, наступним чином: нова -> в процесі -> зворотній зв'язок -> закрита.
7. Додайте записи про витрачений на завдання час.
8. Змініть статуси завдань за допомогою фіксацій в репозиторії.

9. Додайте відомості про витрачений час за допомогою фіксації змін в репозиторії.
10. Підготуйте звіт з лабораторної роботи.

Контрольні питання

1. Для чого необхідні системи управління завданнями?
2. Що таке трекер і яких типів вони бувають?
3. Що таке потік робіт?
4. Поясніть основні принципи роботи з системою призначення завдань.
5. В чому полягають можливості інтеграції системи призначення завдань і системи контролю версій коду?
6. Яким чином в системі Redmine можна змінювати статуси робіт?
7. Які ще існують системи призначені для відстеження задач, що використовуються в проектах розробки програмного забезпечення?

Лабораторна робота № 4

Управління завданнями проекту за допомогою канбан-дошок

Мета роботи

Одержання практичних навичок управління завданнями проекту за допомогою канбан-дошок.

Основні теоретичні відомості

Канбан-дошки

Trello [5] – це популярна он-лайн система управління проектами за методом японських канбан-дошок. Ця система була створена Fog Creek Software в 2011 році. На рис. 4.1 наведено головне вікно системи.

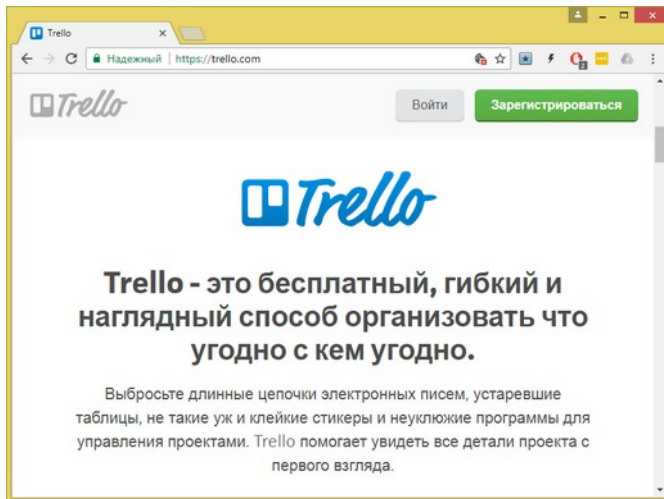


Рисунок 4.1 – Головне вікно системи Trello

Trello використовує парадигму керування проектами, відому як канбан. Kanban – це система розпорядку для ошадливого та «якраз вчасного» (Just in time) виробництва, розроблена в компанії Toyota.

На рис. 4.2 наведено схематичне подання канбан-дошки, що може бути використана при розробці програмного забезпечення.



Рисунок 4.2 – Візуальне подання канбан-дошки

Trello – це максимально простий інструмент, який легко впровадити в робочий процес без довгої адаптації з боку персоналу. Його можна використовувати як особистий органайзер, щоденник, список, колективний to-do менеджер тощо.

Трьома структурними елементами Trello є:

- дошка (board);
- список (list);
- картка (card).

В Trello проекти зображуються дошками, що містять списки. Списки містять картки, якими зображуються завдання. Картки повинні переходити з попереднього списку до наступного, таким чином зображаючи рух завдання від ідеї до завершення.

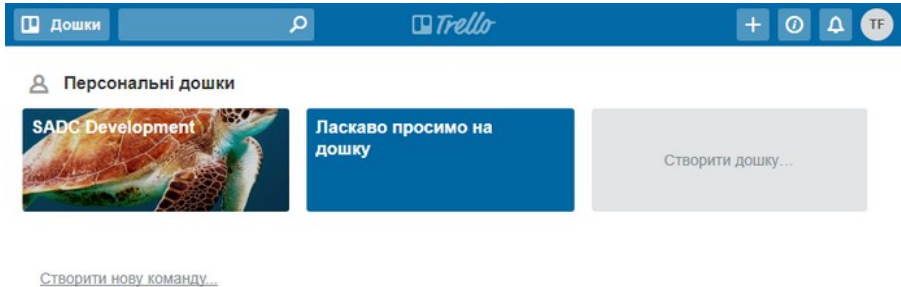


Рисунок 4.3 – Персональна сторінка користувача системи

Дошка – це один робочий екран, який логічно розділений на списки. Списки, в свою чергу, є вертикальними рядками для зберігання карток. Картки – це спеціальні форми для опису завдань. Їх можна рухати як всередині одного списку, так і вільно переміщувати між списками або дошками. Списки теж можна переміщати. Для будь-якого завдання можна призначити людей, відповідальних за її виконання. Користувачі та дошки можуть об'єднуватись в команди.

Є три типи дошок з різним рівнем доступу:

- приватна доступна тільки на особисте запрошення власника дошки;
- командна доступна всім учасникам команди проекту;
- публічна може бути доступна всім.

Закриті дошки і непотрібні списки з картками зберігаються в спеціальному архіві. Звідти їх можна повернути назад або остаточно видалити. В системі можна створювати необмежену кількість завдань, дошок і списків, а також додавати будь-яке число учасників.

Команди дозволяють об'єднати дошки та учасників в єдиному місці. Нові дошки за замовчення можуть бачити всі члени команди, щоб кожен міг мати змогу долучитися. За необхідності можна зробити дошку приватною або публічною.

Дошки дозволяють систематизувати та організувати інформацію про проекти. Вони надають зручний огляд прогресу поставлених завдань. На дошці завдання представляються картками, які розподіляються за списками. Списки потрібні для організації карток на різних стадіях прогресу проекту. Тоді перехід картки між списками можна розглядати як робочий потік або процес. Або ж списки можуть просто служити категоріями чи сховищем ідей та інформації.

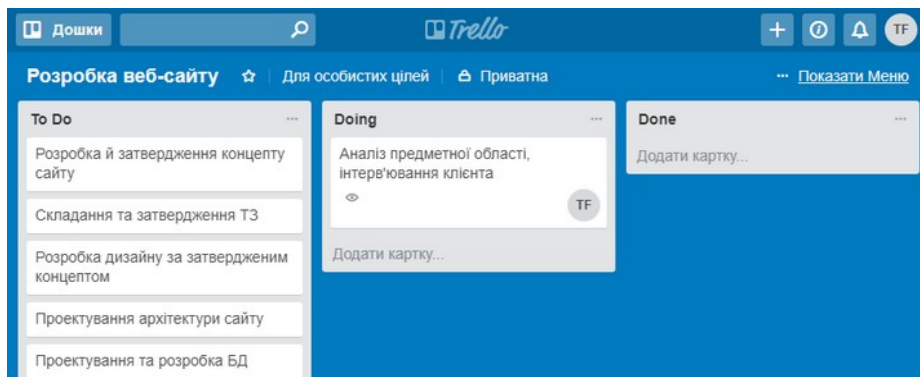


Рисунок 4.4 – Зовнішній вигляд дошки

Як правило, завдання розбиваються на наступні три списки або типи:

- треба зробити (To Do),
- робиться (Doing),
- готово (Done).

Це сама елементарна структура, яку можна змінювати на власний розсуд у відповідності з особливостями робочого процесу вашої команди.

Наприклад, команда розробки може додати до своєї робочої дошки такі списки: «Невиконане», «Виконується», «Проходить тестування», «Підтверджено» і «Завершено».

Для створення команди або дошки натисніть кнопку «плюс» (+) в шапці вікна Trello лівіше від вашого імені. Посилання «Показати меню» в шапці відкриває меню налаштувань дошки.

Додайте списки для створеної дошки.

Додайте на свою дошку учасників, щоб мати можливість назначати їм завдання та дати можливість спільної роботи на дошці. Натисніть посиланні «Додати учасників» в меню дошки та оберіть учасників своєї команди або запросіть нових учасників за адресами електронної пошти.

Додайте картки для поставлених завдань. Для цього натисніть посилання «Додати картку» в першому списку. Картка повинна мати лаконічне ім'я. За домовленістю в імені картки на початку у круглих дужках указують оцінку розміру або трудомісткості завдання (5), а наприкінці у квадратних дужках вказують реальні затрати [2].

ваша команда знала про оновлення її статусу. Таким чином її учасники будуть знати, над чим йшла робота. Якщо згадати конкретного учасника дошки або команди у коментарі за допомогою символу @, цей учасник отримає сповіщення у Trello. Щоб повідомити всіх на картці або дошці, використайте згадку «@card» або «@board».

Журнал оновлень картки містить історію коментарів і дій на картці й створює шкалу подій, що позначають певний прогрес, наприклад переміщення картки в інший список або завершення чимось пункту списку.

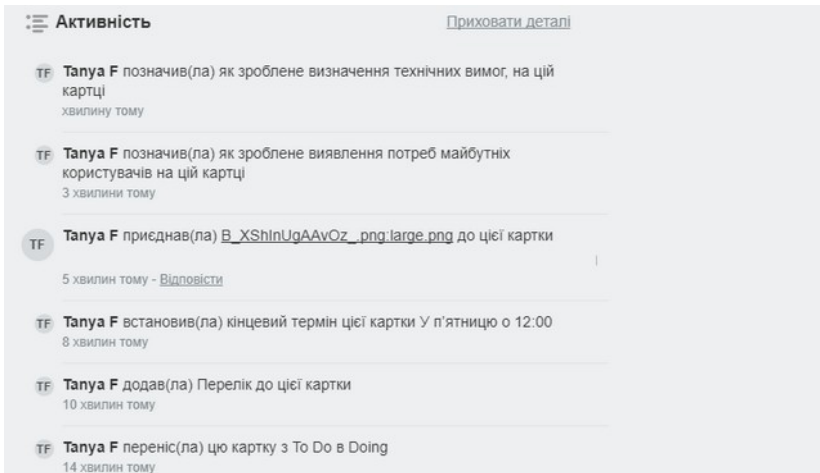


Рисунок 4.6 – Журнал оновлень картки

Додайте учасників дошок до карток, щоб призначити їм завдання. Учасники, яких було додано до карток, будуть отримувати сповіщення про дії на цих картках. Додати учасників до картки можна простим перетягуванням аватарів з розділу меню «учасники». Натисніть кнопку «Учасники» на відкритій картці та виберіть користувачів, яких ви хочете додати.

Мітки категорій і тегів використовуються для фільтрації карток та надання їм пріоритетів. Наприклад, Critical, Major, Minor, Trivial.

Зображення у форматах .jpeg, .gif або .png можна зробити обкладинками картки, і вони будуть відображатися на її передній частині.

Розділ швидких дій на картці дозволяє редагувати мітки, змінити учасників, перемістити картки в інші списки і дошки (наприклад, перемістити до наступного етапу розробки), змінити термін виконання, архівувати картку.

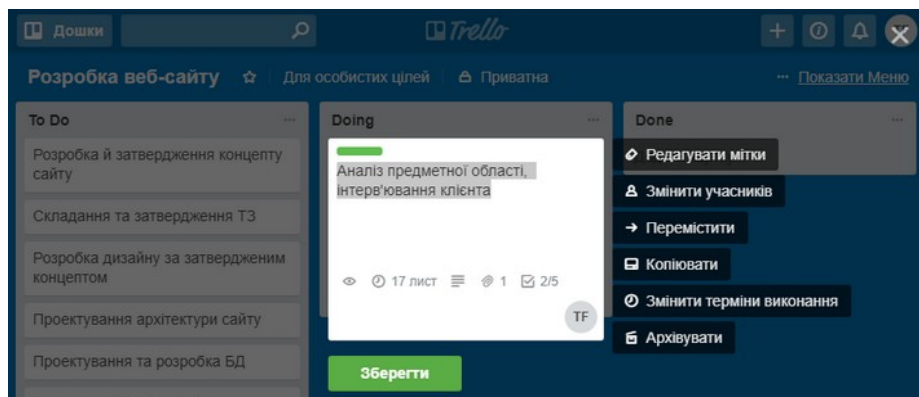


Рисунок 4.7 – Швидке редагування картки

На правій стороні дошки Trello знаходиться меню керування, що дозволяє керувати учасниками, змінювати налаштування, а також вмикати підсилювачі.

Пункт меню «Фільтрувати картки» дозволяє відфільтрувати картки на дошці по імені, міткам, учасникам і датам завершення.

Підсилювачі або покращувачі дошок містять додаткові функції і можливості інтеграції дошок Trello із застосунками, якими ви користуєтесь під час роботи (наприклад, GitHub, Slack, Google Drive). Кожна дошка має один безкоштовний підсилювач. Команди Business Class та Enterprise отримують на своїх дошках необмежену кількість підсилювачів.

Trello працює не лише в браузері. Для роботи з системою також розроблені мобільні застосунки для операційних систем Android та iOS.

Якщо команда розробки використовує Slack для організації комунікації, рекомендується зв'язати команди Trello і Slack, щоб спростити співпрацю в цих двох застосунках. Це дозволить створювати і редагувати картки Trello, призначати учасників команди, а також додавати коментарі і дати завершення безпосередньо з каналів

Slack без переключення між застосунками. Можна також налаштувати сповіщення оновлень з дошок Trello на зазначених каналах Slack, отримати ключову інформацію і взаємодіяти з дошками і картками Trello, що прив'язано до Slack. Застосунок Trello для Slack є безкоштовним.

Історії користувача

Історія користувача (user story) – це спосіб опису вимог до системи, що розроблюється, у вигляді одного чи більше речень, звичайною мовою предметної області, які описують чого користувач хоче досягти. Історії користувача використовуються в гнучких методологіях розробки для з'ясування функцій та можливостей системи, що будуть реалізовуватись.

Кожна історія обмежена в розмірі та складності, адже історії користувача записуються на невеликих картках і мають бути реалізовані в короткі строки від декількох днів до декількох тижнів. Історії повинні легко читатися та бути зрозумілими користувачам, розробникам та всім стейкхолдерам. Вони не деталізовані на самому початку проекту і не пропонують конкретних варіантів досягнення мети. Історії детально розробляються «точно в строк», уникаючи таким чином занадто ранньої визначеності і надмірної обмеженості при формулюванні рішення.

Хороша історія користувача повинна відповідати критерію «INVEST»:

- Independent – незалежна від інших історій;
- Negotiable – деталі отримують шляхом обговорювання всією командою;
- Valuable – цінна для кінцевого користувача;
- Estimable – команда має можливість надати приблизну оцінку її складності та розміру;
- Small – достатньо мала, щоб бути реалізованою в короткий термін;
- Testable – легко протестувати результат, має критерії приймання.

Історії користувача визначаються споживачами програмного забезпечення і висвітлюють їх вимоги до функціонування системи. Історії користувача висвітлює *хто* є користувачем, *що* йому потрібно і

чому.

Історії користувача – це швидкий спосіб оперування вимогами користувача, без необхідності застосування занадто формалізованих документів. Наміром з яким використовують історії користувача є швидке реагування на швидко змінювані вимоги реального світу.

Зазвичай історії користувача відповідають такому шаблону:

"Як <роль користувача>, я хочу <ціль/бажання/дія> щоб <вигода>"

Також використовують скорочений варіант:

"Як <роль>, я хочу <ціль/бажання/дія>"

За необхідності для історій рисують графічне пояснення: мокапи, ескізи користувацьких екранів.

Історії користувача скоріше служать як нагадування замовнику про ідеї для подальших обговорень продукту з командою проекту. В ході обговорень початкові ідеї, викладені в історіях користувача, обростають деталями. Такими деталями є все, що допоможе команді під час реалізації історії – уточнення, обмеження, важливі критерії готовності. Історії поповнюються деталями в міру необхідності, еволюціонуючи від коротких висловлювань до деталізованих і узгоджених вимог з вбудованими критеріями готовності.

Для кожної історії користувача визначається відповідна процедура прийняття, яка за допомогою тестування чи іншим способом визначає чи задоволені вимоги історії користувача.

Користувач розставляє пріоритети серед історій користувача, визначаючи найбільш важливі для системи, які будуть першими реалізовані. Для цього вони повинні бути розбиті на завдання і оцінені розробниками. Пріоритети завдань можуть бути розставлені по системі пріоритетів MoSCoW (Must, Should, Could, Won't). Історії з високим пріоритетом гратимуть ключову роль в корисності системи, в той час як інші будуть скоріш необов'язковим функціоналом, який привертає увагу користувачів або полегшує їхню роботу.

Крім того для кожної історії виконується оцінка її складності або розміру. В методології SCRUM для цього використовуються пункти (points), а процес призначення оцінок називається покером планування.

Перелік історій користувача не є фіксованим. Впродовж виконання проекту історії можуть уточнюватися, видалятися та додаватися, змінювати пріоритети.

Припустимо у замовника є потреба в реалізації системи, яка б

дозволила користувачам зберігати, обмінюватися і продавати свої фотографії. Очікується, що прибуток від системи буде досягатися за рахунок відсотка з продажів.

Цей короткий опис - бачення (vision) системи.

Спочатку необхідно ідентифікувати групи користувачів системи. Знання про майбутніх користувачів допоможе сфокусуватися на потребах кожного з них.

Різними аспектами системи будуть користуватися декілька узагальнених ролей:

- користувачі, які зберігають, обмінюються, продають свої фотографії.
- адміністратори, які будуть забезпечувати підтримку системи для блага інших користувачів.

Можливо, з часом ми зможемо визначити ще якісь ролі. Поки ми випускаємо їх з поля зору. Щоб почати достатньо наявних.

Маючи ролі користувачів і їх основне завдання, спробуємо описати найважливіші історії, які могли б нам розповісти про майбутню систему.

Історії користувачів, які можна виділити для даної системи.

1. Як користувач я можу зберігати свої фотографії в системі, щоб мати можливість показати або продати їх іншим користувачам.
2. Як адміністратор я можу управляти фотографіями користувачів, так щоб контент сайту був легальним.

Під час обговорення першої історії, замовник і команда приходять до того, що користувачі системи повинні бути авторизовані системою перед виконанням будь-яких дій з фотографіями. Це призводить до появи нової користувацької ролі гостя - групі людей, які неавторизовані системою або взагалі поки не мають облікового запису.

3. Як гість я можу зареєструватися в системі для отримання призначеного для користувача облікового запису та подальшої роботи.

4. Як гість я можу увійти в систему під раніше створеним обліковим записом, для подальшої роботи.

5. Як користувач я можу видалити свій обліковий запис і перестати бути користувачем системи.

6. Як користувач я можу змінити дані свого облікового запису.

7. Як користувач я можу зробити деякі поля свого облікового запису видимими для інших користувачів.

Розглянемо історію номер 3, одну з ідентифікованих вище.

Під час обговорення цієї історії з командою замовнику задають питання про те яка інформація потрібна для створення облікового запису. Обговорюючи різні варіанти, замовник і команда приходять до того, що для першої версії системи досить буде перевіреного електронної адреси плюс імені користувача та його пароля.

До історії дописується цієї коментар. Тепер історія виглядає так:

3. Як гість я можу зареєструватися в системі для отримання призначеного для користувача облікового запису та подальшої роботи. Потрібен перевірений email і вибрані користувачем ім'я і пароль.

Під час подальших обговорень хтось із тестувальників запитує про мінімальну довжину пароля і перевірки на унікальність імені. Продовжуючи дискусію, команда і замовники приходять до думки, що необхідно описати основні критерії готовності історії, щоб команда розуміла очікування і знала, коли оголошувати історію готовою:

3. Як гість я можу зареєструватися в системі для отримання призначеного для користувача облікового запису та подальшої роботи. Потрібен перевірений email і вибрані користувачем ім'я і пароль.

Тест 1: користувач не може ввести пароль менше 6 символів.

Тест 2: користувач не може ввести ім'я менше 3 і більше 20 символів.

Тест 3: користувач повинен мати унікальне ім'я в системі.

Тест 4: після реєстрації користувач повинен отримати лист на email для активації свого облікового запису.

Тест 5: користувач не може увійти в систему, якщо обліковий запис не було активовано.

Тест 6: при успішному вході система вітає користувача текстом «Ласкаво просимо, ім'я користувача».

Наведемо приклад, як могли б бути відсортовані історії вищеописаного проекту за пріоритетам.

1. Як гість я можу зареєструватися в системі для отримання призначеного для користувача облікового запису та подальшої роботи.

2. Як гість я можу увійти в систему під раніше створеним обліковим записом, для подальшої роботи.

3. Як користувач я можу зберігати свої фотографії в системі, щоб мати можливість показати або продати їх іншим користувачам.

4. Як адміністратор я можу управляти фотографіями користувачів, так щоб контент сайту був легальним.

5. Як користувач я можу змінити дані свого облікового запису для

корегування невірних даних.

6. Як користувач я можу зробити деякі поля свого облікового запису видимими для інших користувачів.

7. Як користувач я можу видалити свій обліковий запис і перестати бути користувачем системи.

Як можна побачити, історії відсортовані в порядку, який, по-перше, логічний з точки зору замовника і команди, а по-друге цінність історій зменшується зверху вниз. Таким чином, якщо, наприклад, на половині проекту настає нестача ресурсів (скажімо, після реалізації історії для адміністратора системи), замовники зможуть отримати вигоду від продукту, так як найбільш важливі історії вже будуть реалізовані.

Одна з описаних на ранній фазі проекту історій в якийсь момент може здатися занадто великою в порівнянні з іншими, що ускладнює розуміння її пріоритету. Така велика історія потребує розбиття на менші історії.

Наприклад:

Як користувач я можу зберігати свої фотографії в системі, щоб мати можливість показати або продати їх іншим користувачам.

У цьому випадку замовник і команда можуть спробувати розбити її на кілька дрібніших історій, кожна з яких може отримати свій пріоритет.

Як користувач я можу зберігати свої фотографії в системі, щоб мати можливість показати їх іншим користувачам.

Як користувач я можу зберігати свої фотографії в системі, щоб мати можливість продати їх іншим користувачам.

Завдання на лабораторну роботу

1. Створіть команду з 3-5 студентів групи в Trello.
2. Створіть дошку для розробки програмного продукту.
3. Створіть необхідні списки.
4. Оберіть прикладну область розробки і шляхом обговорення командою визначте щонайменше 10 історій користувача. Для 3-5 історій визначте детальний опис, критерії приймання, а також перелік завдань, які необхідно виконати для реалізації історії. Визначте пріоритети історій та оцініть їхню складність.
5. В Tello створіть картки, що відображають розроблені історії

користувача.

6. Продемонструйте принцип командної роботи з системою.
7. Підготуйте звіт з лабораторної роботи.

Контрольні питання

1. Що таке канбан-дошка? Як її використовувати для управління проектами?
2. Поясніть принципи роботи з системою Trello. Які в неї можливості?
3. Що таке дошка, список та картка? Як їх використовувати?
4. Що таке підсилювачі в Trello і навіщо вони?
5. Що таке журнал оновлень картки?
6. Як організувати роботу команди за допомогою Trello.
7. Поясніть принцип роботи з карткою на канбан-дошці? Чому у картки дві сторони?
8. Які принципи створення списків на канбан-дошці?
9. Що таке історія користувача? В якому форматі вони формулюються.
10. Що таке критерії готовності та критерії приймання для історії користувача.
11. Що таке гнучкі методології розробки програмного забезпечення?
12. Опишіть методологію розробки SCRUM, її основні характеристики.

Лабораторна робота № 5 **Прототипування інтерфейсів програмних систем**

Мета роботи

Одержання практичних навичок прототипування інтерфейсів програмних систем.

Основні теоретичні відомості

При створенні технічного завдання також визначається як користувачі будуть взаємодіяти з майбутньою системою.

Метою User Experience Design (UX design) є дизайн взаємодії з користувачем. Для цього створюються вайрфрейми.

Вайрфрейм (wireframe) - це образ дизайну низької точності. не сильно деталізований рисунок дизайну в чорно-білому вигляді з упором не на візуальну складову, а на структуру і зміст. Його створюють на початковому етапі роботи над проектом, щоб скласти загальну картину майбутнього проекту.

Основна мета створення вайрфреймів полягає в тому, щоб зобразити загальну функціональність самого продукту. Вони будуть відображати такі речі, як навігація, основні групи змісту, кнопки, керуючі елементи і загальний макет. Це спосіб візуалізації того, як елементи інтерфейсу будуть відображатися для користувачів на найважливіших екранах.

За допомогою вайрфреймів можна визначити фундаментальну структуру майбутнього продукту, щоб отримати від зацікавлених сторін і розробників підтвердження, що система правильно працює, перш ніж приступити до створення візуального дизайну.

Вайрфрейм повинен показувати:

- основні групи контенту (що?);
- структуру інформації (де?);
- опис і базову візуалізацію взаємодії між інтерфейсом і користувачем (як?).

Класичні вайрфрейми будуються з сірих і білих блоків. Допускається використання символів. Замість зображень в вайрфреймах рекомендується використовувати плейсхолдери.

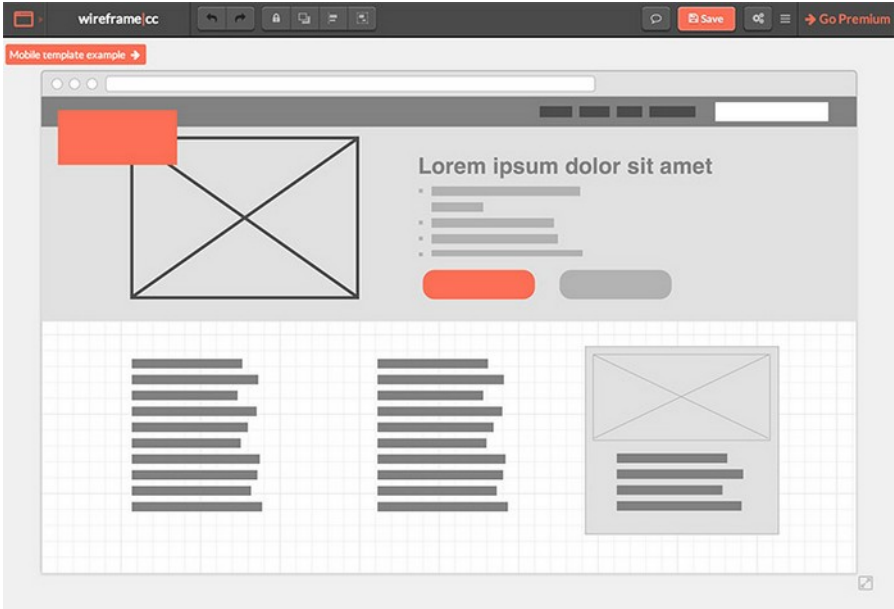


Рисунок 5.1 – Приклад вайрфрейму

Середній час на створення вайрфреймів має бути обмеженим.

Вайрфрейми зазвичай використовуються як документація до проекту. Оскільки вони є статичним і фіксованим способом відображення інтерфейсу, їх потрібно відповідно описати (від короткого пояснення, до складної технічної документації, якщо це необхідно).

Також вайрфрейми корисні як швидкі начерки для внутрішньої комунікації в команді.

Для розробки вайрфреймів існує велика кількість програмних продуктів. Наприклад, Balsamic, Ninjamock, wireframe.cc, Lucidchart і ін.

Набір вайрфреймів може бути злінкований для побудови прототипу для демонстрації переходів між екранами.

Для візуалізації досвіду користувача в продукті застосовуються користувацькі сценарії (user flows).

User flows (також відомі як користувацькі сценарії, UX-сценарії, Wire, UI або UX сценарії) - це наочні матеріали, які ілюструють весь шлях користувача в продукті цілком.

User flow - це гібрид класичної блок-схеми і елементів візуального інтерфейсу. В основі user flow лежать завдання, які виконує користувач, а також можливі альтернативні виходи.

User flow виглядає як схематично зображені сторінки сайту і переходи між ними які побудовані для одної або декількох груп цільової аудиторії. User Flow показує які дії здійснює відвідувач користуючись системою, відповідно до вашого сценарію.

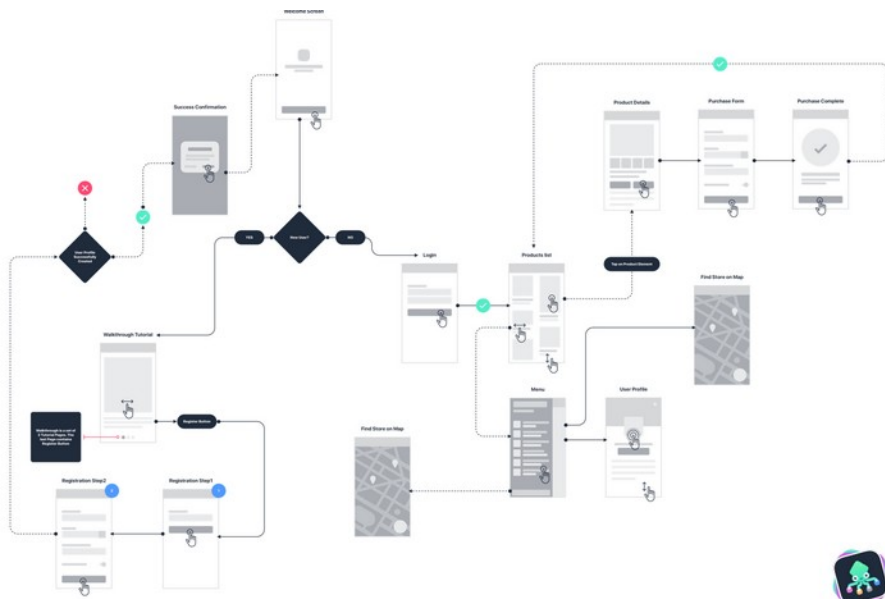


Рисунок 5.2 – Приклад User flow

Завдання на лабораторну роботу

1. Оберіть програмний продукт для побудови вайрфреймів.
2. Оберіть тему для розробки веб застосунку та підготуйте 4-5 вайрфреймів. Зліпуйте вайрфрейми між собою.
3. Оберіть тему для розробки мобільного застосунку та підготуйте 4-5 вайрфреймів. Зліпуйте вайрфрейми між собою.



Контрольні питання

1. Що таке вайрфрейми? Для чого вони використовуються?
2. Чим вайрфрейми відрізняються від мокапів?
3. Які елементи використовуються для побудови вайрфреймів?
4. Оберіть програмний засіб для побудови вайрфреймів та дайте характеристику його функціональних можливостей.
5. Що таке користувацькі сценарії? Для чого вони використовуються і як будуються?

Лабораторна робота № 6 **Побудова ментальних карт**

Мета роботи

Одержання практичних навичок побудови ментальних карт для візуалізації функціональних вимог до проекту.

Основні теоретичні відомості

Діаграма зв'язків, відома також як інтелект-карта, карта думок (Mind map) або асоціативна карта - це інструмент візуального відображення інформації, що дозволяє ефективно структурувати і обробляти її.

Інтелект-карта являє собою деревоподібну схему, на якій зображені слова, ідеї, завдання або інші поняття, пов'язані гілками, що відходять від центрального поняття або ідеї. Гілки, що мають форму плавних ліній, позначаються і пояснюються ключовими словами або образами.

Використовуючи mindmap, створюють технічні завдання, консультанти-аналітики створюють проекти, дизайнери придумують концепції, тренери створюють презентації, менеджери складають плани і т.д.

Інтелект-карти є розробкою Тоні Бьюзена, британського письменника, лектора і консультанта з питань інтелекту, психології навчання і проблем мислення. На думку розробника, інтелект-карта - це потужний графічний метод, що надає універсальний ключ до вивільнення потенціалу, прихованого в мозку.

Сам процес створення інтелект-карт отримав назву майндмеппінга (mindmapping) і на сьогоднішній день визнаний зручною і ефективною технікою візуалізації мислення та альтернативної записи - на відміну від традиційної лінійної форми з використанням тексту, списків, таблиць і схем.

Більшість проектів починаються з мети, наприклад, створення нового веб-сайту, розробка нової функції або вдосконалення користувацької роботи, щоб знизити рівень відмови. Цілі легко визначити. Виявлення, як досягти мети - це важка частина. Інтелект-карти полегшують розподіл великих цілей на реальні завдання. Вони прискорюють творчість, допомагають генерувати та фіксувати ідеї, а

також дозволяють зважити кілька варіантів, щоб обрати найкраще з можливих рішень.

При розробці програмного забезпечення інтелект-карти використовують для:

- визначення вимог проекту;
- створення дорожньої карти продукту;
- визначення маркетингової стратегії;
- визначення пріоритетів і цілей.

Інтелект-карти ідеально підходять для менеджерів, UX дизайнерів, бізнес-аналітиків та команд розробників.

Для побудови інтелект-карт існує безліч програмних засобів. Наприклад, XMind, MindMup, Freemind, MindMeister та інші.

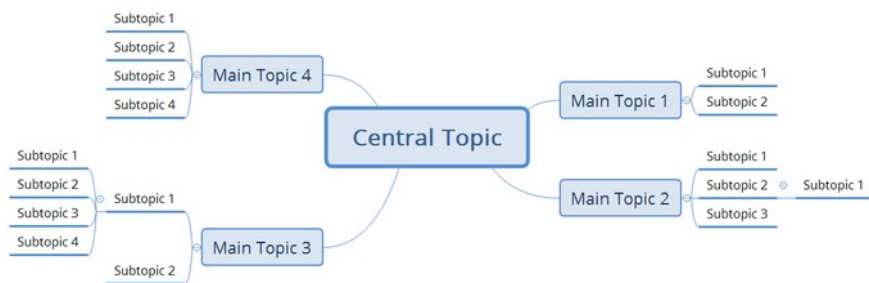


Рисунок 6.1- Принцип побудови інтелект-карт

Завдання на лабораторну роботу

1. Оберіть тему для розробки програмного забезпечення і для побудови інтелектуальних карт.
2. Створіть інтелектуальну карту для відображення функціональних вимог до програмного продукту.

Контрольні питання

1. Що таке інтелектуальна карта?
2. Для чого використовуються інтелектуальні карти?
3. Як будуються інтелектуальні карти?
4. Побудуйте невелику інтелектуальну карту на обрану викладачем тему.

Лабораторна робота № 7

Методи побудови мережевих моделей та діаграм передування для програмних проєктів

Мета роботи

Використовуючи задані параметри передування задач побудувати мережевий графік проєкту у вигляді стрілочної діаграми та у вигляді діаграми передування.

Основні теоретичні відомості

Мережеві графіки служать фундаментальними засобами моніторингу та контролю робіт проєкту. Разом з планом і бюджетом проєкту вони є найголовнішим інструментом управління проєктами.

Існують 3 основні форми відображення мережевої моделі проєкту: форма «Роботи на дугах», форма «Роботи в вузлах» і форма «Діаграма Ганта».

Стрілочні діаграми (Arrow Diagramming Method, ADM, «**роботи на дугах**», Activity on Arrow, AoA) використовують стрілки для подання робіт і пов'язують їх одна з одною в вузлах, що показують їх логічні взаємозв'язки. Стрілочна діаграма використовує такі основні елементи: робота, подія, залежність.

Робота – процес, що вимагає витрат часу і ресурсів і призводить до певного результату. Робота позначається стрілкою, над якою вказують назву роботи та за необхідності в дужках або знизу під стрілкою її тривалість. Початок стрілки позначає початок роботи, а голова – закінчення. Іноді довжина стрілки означає тривалість роботи.

Подія – факт закінчення однієї або декількох робіт, необхідний і достатній для початку наступних робіт. Будь-яка робота на мережевому графіку обмежена двома подіями. Вони пов'язують роботи в єдину мережу. Подія не має тривалості і не потребує ресурсів адже вона просто фіксує результат роботи. Подія є вузлом на діаграмі і позначається кружком, в якому проставляється унікальний номер.

В проєкті завжди є одна початкова подія, що відображає факт його початку, та одна кінцева подія, що відображає той факт, що всі роботи проєкту завершено.

Залежність (фіктивна робота) вводиться для відображення технологічної або організаційної залежності робіт і не вимагає витрат

ні часу, ні ресурсів. Залежність відображає логічну залежність однієї роботи від іншої і зображується пунктирною стрілкою. Фіктивна робота позначається пунктирною стрілкою.

Як приклад застосування залежностей можна навести наступні випадки:

- коли наступна робота вимагає закінчення декількох попередніх робіт, фіктивна робота дозволяє показати існуючу логічну залежність;

- коли дві роботи виконуються паралельно, використовується додаткова подія та фіктивна робота, щоб усунути неоднозначність та показати завершення однієї з робіт.

Таким чином мережева модель у вигляді стрілочної діаграми зв'язується через події, тобто через конкретні результати, а роботи виникають як наслідок, тобто як відповідь на питання «Що потрібно зробити для отримання наступного результату». В цих діаграмах результат домінує над процесом.

Побудуємо стрілочну діаграму за вихідними даними, наведеними в табл. 7.1.

Таблиця 7.1 – Вихідні дані про роботи проекту

Робота	Попередня робота
A	-
B	A
C	-
D	B,C
E	C
F	E
G	E

На рис. 7.1 наведено результат побудови стрілочної діаграми.

При побудові мережевого графіка (рис. 7.1) було враховано такі припущення і обмеження:

- роботи A і C не мають попередніх, тобто вони починаються з вихідної події графіка;

- у роботи E попередником є робота C, а у роботи D дві попередні роботи B і C, тому між третьою і четвертою подіями показана залежність;

- роботи F і G мають одну і ту ж попередню роботу E, тому

введено додаткову подію 6 для усунення неоднозначності.

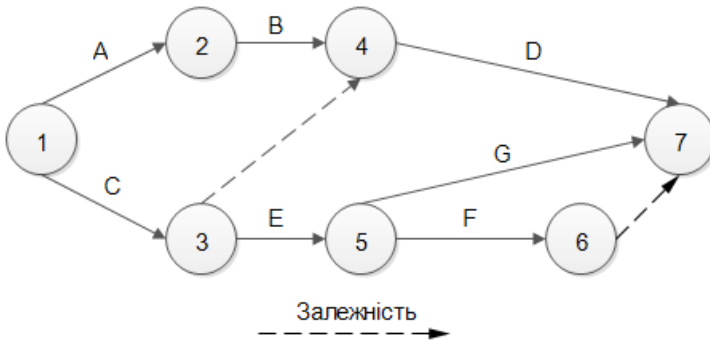


Рисунок 7.1 – Мережевий графік у вигляді стрілочної діаграми («роботи на дугах»)

Розглянемо деякі приклади побудови стрілочних діаграм, які потребують використання залежностей.

Нехай є параметри передування задач проекту (табл. 7.2):

Таблиця 7.2 – Приклад №1 передування робіт

Робота	Попередня робота
A	-
B	-
C	A
D	A,B

Щоб відобразити таку залежність між роботами, необхідно використати фіктивну роботу, адже роботи C та D мають різний набір попередніх робіт. Фрагмент стрілочної діаграми, що відображає таку залежність наведено на рис. 7.2.

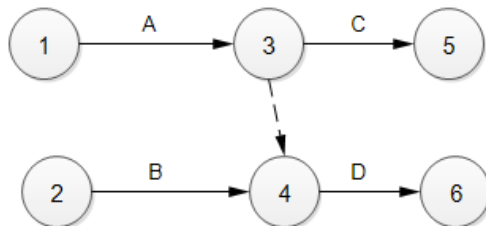


Рисунок 7.2 – Стрілочна діаграма для прикладу №1

Розглянемо наступний приклад (табл. 7.3).

Таблиця 7.3 – Приклад №2 передування робіт

Робота	Попередня робота
A	-
B	-
C	-
D	A,B
E	B
F	B,C

Стрілочна діаграма, що відображає таку залежність між роботами наведена на рис. 7.3. Зверніть увагу, що роботи D, E, F не розділяють один й той самий набір попередніх робіт. Тому виникає необхідність використати дві фіктивні роботи, щоб показати необхідну послідовність робіт проекту.

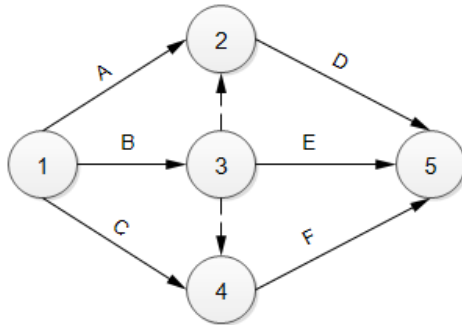


Рисунок 7.3 – Стрілочна діаграма для прикладу №2

Розглянемо наступний приклад (табл. 7.4). Стрілочна діаграма, що відображає таку залежність між роботами наведена на рис. 7.4.

Таблиця 7.4 – Приклад №3 передування робіт

Робота	Попередня робота
A	-
B	-
C	A
D	A,B
E	B

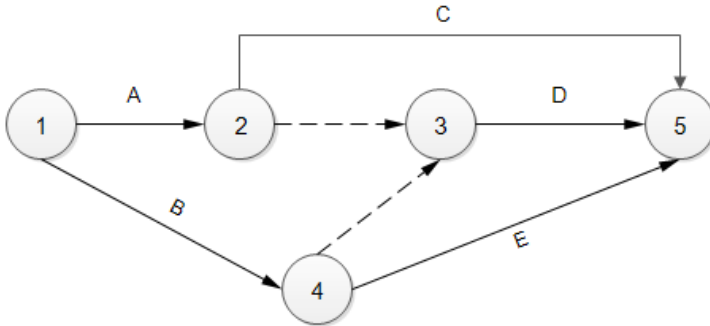


Рисунок 7.4 – Стрілочна діаграма для прикладу №3

Діаграми передування (Precedence Diagramming Method, PDM, «**роботи в вузлах**», Activity on Node, AoN) використовують прямокутники (вузли) для подання робіт, а залежності між ними відображаються стрілками, що показують взаємозв'язок робіт тобто логічну залежність однієї роботи від іншої. В цих діаграмах процес домінує над результатом.

Діаграма передування, побудована на основі початкових даних, представлених в табл. 7.1, наведена на рис. 7.5.

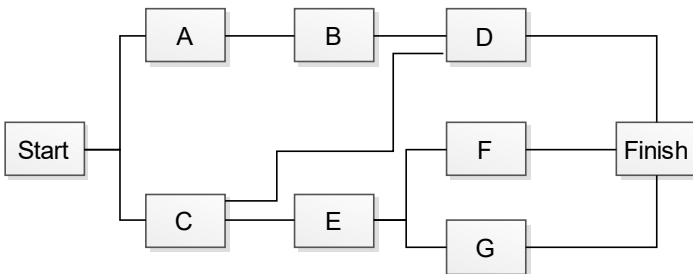


Рисунок 7.5 – Діаграма передування проекту («роботи в вузлах»)

Завдання на лабораторну роботу

1. Ознайомитися з основними правилами побудови мережних діаграм проектів.
2. Для побудови діаграм використати одну з безкоштовних онлайн систем для побудови діаграм, як наприклад Draw.io.
3. Побудувати стрілочну діаграму на підставі індивідуального завдання в додатку А.1.

4. Побудувати діаграму передування на підставі індивідуального завдання в додатку А.1.
5. Проаналізувати отримані результати.

Контрольні питання

1. Які правила побудови мережевої моделі проекту у формі стрілочної діаграми («Роботи на дугах»)?
2. Які правила побудови мережевої моделі проекту у формі діаграми передування («Роботи в вузлах»)?

Лабораторна робота № 8

Розрахунок тривалості проекту методом критичного шляху CPM та методом PERT

Мета роботи

Розрахувати тривалість проекту за допомогою двох методів: методом критичного шляху CPM та методом PERT.

Основні теоретичні відомості

Розглянемо приклад розрахунку параметрів мережевої моделі на прикладі проекту, наведеного в попередній роботі. Мережева модель проекту наведена на рис. 7.1 та 7.2. Проект складається з шести робіт. Тривалість робіт наведена в табл. 8.1.

Таблиця 8.1 – Характеристика робіт проекту

Робота	Тривалість роботи
A	3
B	2
C	6
D	4
E	2
F	1
G	3

Розглянемо приклад **розрахунку тривалості проекту та його критичного шляху**. Хоча метод розрахунку для стрілочних діаграм і діаграм передування є однаковим, інформація на мережевих діаграмах цих двох типів подається по різному. Розрахунок критичного шляху виконується двома проходами по діаграмі: вперед та назад. При прямому проході мережевою діаграмою визначаються найбільш ранні строки виконання для вузлів. При виконанні зворотного проходу визначаються пізні строки виконання для вузлів. Після проходів обчислюються резерви часу і як результат визначається тривалість всього проекту та критичний шлях проекту.

При використанні секторного методу на стрілочних діаграмах проекту подія розбивається на чотири сектори, в яких зазначається номер події (і), найбільш *ранній строк звершення події* (Earliest event

time, EET), найбільш пізній строк звершення події (Latest event time, LET) та резерв часу для події (R_i). На рис. 8.1 наведено приклад подання вузла.

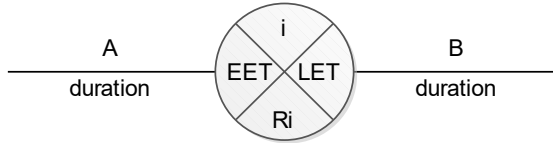


Рисунок 8.1 – Секторне подання вузла на стрілочній діаграмі

Розглянемо розрахунки для стрілочної діаграми, що використовується в якості прикладу (рис. 8.2).

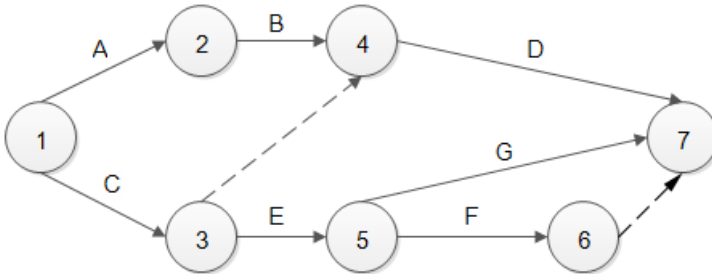


Рисунок 8.2 – Стрілочна діаграма проекту

При прямому проході розраховується ранній строк звершення подій. Ранній строк звершення початкової події дорівнює нулю ($EET_{\text{first}} = 0$). Для кожної наступної події ранній строк звершення дорівнює ранньому строку звершення пов'язаної попередньої події плюс тривалість роботи, що зв'язує їх. Якщо подія має декількох попередників, обирається найбільше з розрахованих значень. В результаті прямого проході отримаємо тривалість проекту, що дорівнює строку звершення останньої події.

При зворотному проході обчислюються пізні строки звершення події. Для останньої події ранній строк стає пізнім строком звершення ($LET_{\text{last}} = EET_{\text{last}}$). Пізні строки інших подій отримуються шляхом віднімання від пізнього строку наступної події тривалості роботи, що їх зв'язує. Якщо подія має декількох наступників, обирається найменше з розрахованих значень.

Після двох проходів для вузлів розраховується часовий резерв

подій, який дорівнює $R = LET - EET$.

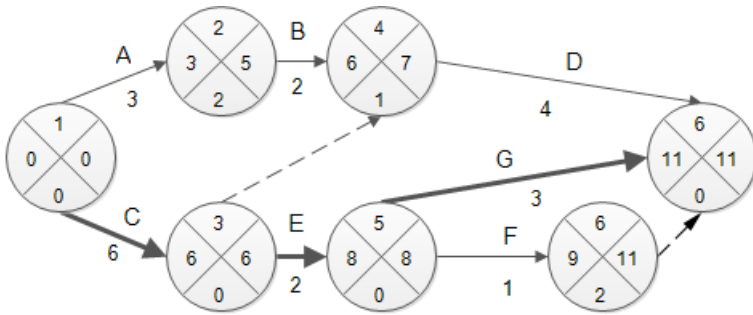


Рисунок 8.3 – Розрахунок критичного шляху для стрілочної діаграми

Найдовший шлях від початкового вузла до останнього через всі вузли з нульовим резервом, називається **критичним шляхом**. Довжина критичного шляху дорівнює тривалості проекту. Іншими словами можна сказати, що критичний шлях – це послідовність робіт проекту, від якої залежить строк завершення проекту. Знайдений критичний шлях проекту на стрілочній діаграмі позначається жирними стрілками.

Для діаграм передування вузол роботи розбивається на декілька частин, в яких вказується назва роботи, її тривалість, *ранній початок роботи* (ES), *раннє завершення* (EF), *пізній початок* (LS), *пізнє завершення* (LF), *резерв часу*. На рис. 8.4 наведено приклад позначення вузла з роботою.



Рисунок 8.4 – Подання вузла на діаграмі передування

Ранній початок ES (Early Start) – найбільш ранній з можливих термінів початку роботи. Раннє завершення EF (Early Finish) – найбільш ранній з можливих термінів завершення роботи. Пізній початок LS (Late Start) – найбільш пізній з можливих термінів початку роботи. Пізнє завершення LF (Late Finish) – найбільш пізній з можливих термінів завершення роботи.

При прямому проході розраховуються ранні строки для робіт. Для перших робіт, які не залежать від інших, ранній початок дорівнює нулю ($ES_{\text{first}} = 0$). Раннє завершення роботи дорівнює ранньому початку плюс тривалість роботи ($EF = ES + T$). Для кожної наступної роботи ранній початок дорівнює максимальному з ранніх завершень всіх її попередників. В результаті прямого проходу отримаємо тривалість проекту, що дорівнює максимальному з ранніх строків завершення останніх робіт, що не мають наступників.

При зворотному проході обчислюються пізні строки завершення і початку робіт. Для останніх робіт пізні строки завершення дорівнюють тривалості проекту, тобто максимальному із ранніх строків завершення останніх робіт $LF_{\text{last}} = \max(EF_{\text{last}})$. Пізній старт роботи дорівнює пізньому завершенню відняти тривалість роботи ($LS = LF - T$). Для попередніх робіт пізнє завершення роботи дорівнює мінімальному з пізніх початків всіх наступників.

Після двох проходів для вузлів розраховується загальний часовий резерв для робіт, який дорівнює $TF = LF - EF$.

Загальний (повний) резерв часу (Total float) показує проміжок часу, на який можна затримати початок роботи або збільшити її тривалість без зміни терміну завершення проекту.

Окремий (вільний) резерв часу (Free Float) – проміжок часу, на який можна затримати початок роботи, або збільшити її тривалість без зміни раннього початку наступних робіт.

Розглянемо розрахунки для діаграми передування, що використовується в якості прикладу (рис. 8.5).

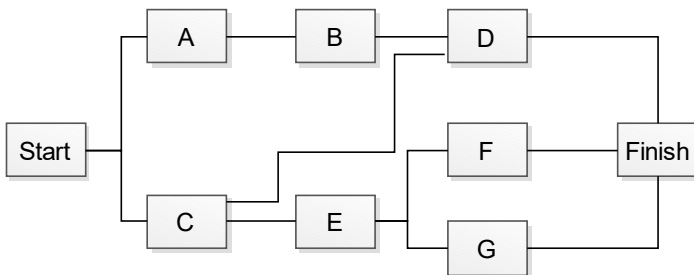


Рисунок 8.5 – Діаграма передування проекту

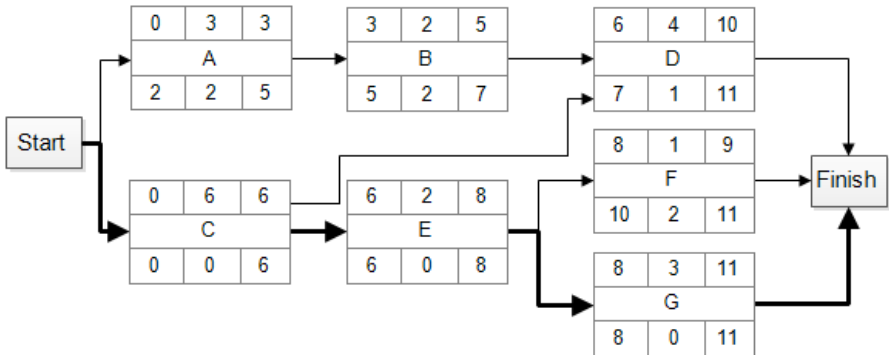


Рисунок 8.6 – Розрахунок критичного шляху для діаграми передування

Критичний шлях проходить по роботах C, E і G, що не мають резервів часу. Тривалість проекту становить 11 днів. При цьому робота A не має окремого (вільного) резерву часу, її затримка призведе до зриву термінів початку наступної роботи B.

Окремий резерв роботи не може бути більше загального. Він показує той резерв, який є в розпорядженні менеджера, щоб не зірвати початок робіт наступників.

У мережевому графіку може бути кілька критичних шляхів, але всі вони повинні починатися в вихідному подію графіка і без розривів завершуватися в кінцевому подію.

Метод аналізу і оцінки програм PERT (Program Evaluation and Review Technique) має переваги перед методом критичного шляху в ситуаціях, коли досягнення цілей проекту пов'язано з фактором невизначеності.

Для оцінки діапазону тривалості робіт PERT застосовує три оцінки:

- *найбільш ймовірна* (M) – тривалість роботи визначається з урахуванням попереднього виділення ресурсів, їх продуктивності, реалістичної оцінки їх доступності для виконання даної роботи, залежності від інших учасників і затримок;
- *оптимістична* (O) – тривалість роботи ґрунтується на аналізі найбільш сприятливого сценарію розвитку роботи;
- *песимістична* (P) – тривалість роботи ґрунтується на аналізі найбільш несприятливого сценарію розвитку роботи.

Розглянемо застосування методу PERT на наступному прикладі. Нехай тривалість робіт наведеного вище проекту наведено в табл.8.2.

Таблиця 8.2 – Оцінка тривалості робіт проекту

Робота	Оптимістична оцінка тривалості, О	Найбільш імовірна тривалість, М	Песимістична оцінка, Р
A	2	3	4
B	2	2	3
C	4	6	9
D	3	4	5
E	2	2	3
F	1	1	3
G	2	3	4

Очікувана тривалість роботи дорівнює середньому зваженому трьох оцінок:

$$m = (P + 4M + O) / 6.$$

Стандартне відхилення тривалості роботи дорівнює

$$\sigma = (P - O) / 6.$$

Розрахунок очікуваного часу завершення робіт, дисперсії і середньоквадратичного відхилення приведені в табл. 8.3, а модель проекту на рис. 8.5.

Таблиця 8.3 – Оцінка параметрів проекту

Робота	Очікувана тривалість, М	Стандартне відхилення, σ	Дисперсія, D
A	3	0.33	0.11
B	2.2	0.17	0.03
C	6.2	0.83	0.69
D	3.8	0.33	0.11
E	2.2	0.17	0.03
F	1.3	0.33	0.11
G	3	0.33	0.11

Після розрахунків очікуваної тривалості необхідно побудувати мережеву діаграму проекту та визначити критичний шлях проекту. Нехай після розрахунків визначено, що критичний шлях проходить по роботах С, Е і G.

Для розрахунку тривалості шляху додається очікувана тривалість виконання всіх робіт, що лежать на шляху. Для визначення стандартного відхилення тривалості робіт на шляху необхідно скористатися формулою

$$\sigma = \sqrt{D_1 + \dots + D_n}.$$

Тоді в наведеному прикладі тривалість критичного шляху $T_k = MC + ME + MG = 6.2 + 2.2 + 3 = 11.4$ днів. Дисперсія критичного шляху $D_k = DC + DE + DG = 0.69 + 0.03 + 0.11 = 0.83$. Стандартне відхилення $\sigma = \sqrt{D_k} = 0.91$.

Стандартне відхилення можна використовувати для ілюстрації ступеня невизначеності виконання проекту за час T_k (рис. 2.7).

В межах трьох стандартних відхилень від T_k тривалість виконання проекту може змінюватися від 8,67 до 14,13 днів. В межах одного стандартного відхилення тривалість виконання проекту може мінятися від 10,49 до 12,31 днів з ймовірністю в 68%. Ймовірність завершення проекту між 8,67 до 14,13 днями дорівнює 99,74%.

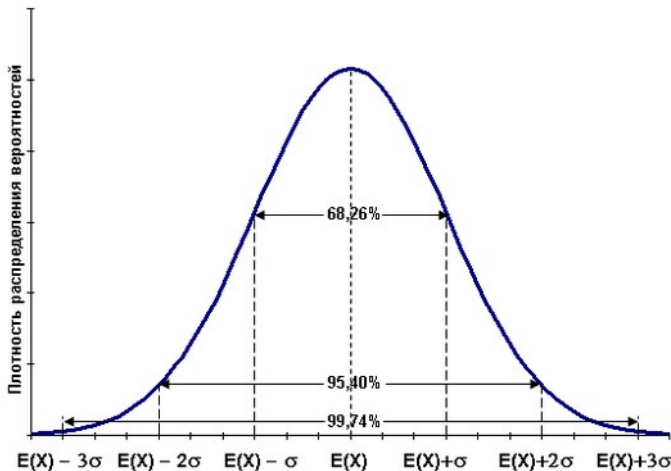


Рисунок 8.7 – Ступінь невизначеності тривалості проекту

Підрахувавши середню тривалість робіт проекту і середні відхилення робіт, можна за допомогою статистичних таблиць розрахувати ймовірність виконання проекту (або частини проекту) до конкретного часу. Для цього необхідно розрахувати наступну величину

$$Z = [T_s - T_o] / \sigma, \quad (8.1)$$

де T_s – планова тривалість робіт, для яких проводяться розрахунки;

T_o – очікувана тривалість робіт;

σ – стандартне відхилення робіт;

Z – кількість стандартних відхилень від середнього, яка за статистичними таблицями може бути переведена на ймовірність $P(z < Z)$.

В таблиці 8.4 наведені деякі значення величини Z та відповідні ймовірності.

Таблиця 8.4 – Таблиця відповідності величини Z та ймовірності

Величина Z	Ймовірність	Величина Z	Ймовірність
-2,0	0,02	+2,0	0,98
-1,5	0,07	+1,5	0,93
-1,0	0,16	+1,0	0,84
-0,7	0,24	+0,7	0,76
-0,5	0,31	+0,5	0,69
-0,3	0,38	+0,3	0,62
-0,1	0,36	0,1	0,54

В програмі Excel шукана ймовірність може бути отримана за допомогою функції НОРМ.СТ.РАСП(«значення Z »;1).

Нехай необхідно визначити ймовірність завершення проекту за 10 днів. Тоді $Z = (10 - 11,4) / 0,91 = -1,53$. Ймовірність, що відповідає значенню $Z = -1,53$ приблизно дорівнює 0,07. Це значить, що ймовірність завершити роботи над проектом на 10 день або раніше дорівнює 7 %.

Можна вирішити зворотну задачу і визначити, який граничний кінцевий термін відповідає заданому рівню ймовірності завершення проекту. Припустимо, що необхідно визначити, який граничний кінцевий термін відповідає 95% рівню ймовірності завершення

проекту.

Знаходимо значення Z , що відповідає ймовірності 0,95%.

В програмі Excel для знаходження Z можна скористатися функцією =НОРМ.СТ.ОБР(«ймовірність»).

В результаті знайдене значення $Z = 1,645$, що відповідає ймовірності 0,95%. Підставивши знайдене Z в рівняння (3.1) і вирішивши його, отримаємо строк виконання в 12,9 днів.

Такі самі розрахунки можна провести для визначення ймовірності завершення некритичних шляхів проекту. Щоб визначити ймовірність того, що некритичний шлях затримає проект, необхідно задати в якості планової тривалості T_s очікувану довжину критичного шляху (тобто очікувану тривалість проекту). Розрахованому значенню Z відповідає ймовірність завершення шляху в заданий строк (P_s). Тоді ймовірність затримання проекту $P_z = 1 - P_s$.

Завдання на лабораторну роботу

1. Мережевий графік проекту взяти з роботи №7.
2. За індивідуальним завданням в додатку А.2 провести розрахунки методом PERT, розрахувавши для робіт очікувані тривалості, дисперсії та стандартні відхилення.
3. Знайти критичний шлях проекту, використовуючи знайдені очікувані тривалості робіт. Розрахунок критичного шляху провести на стрілочній діаграмі та на діаграмі передування. Визначити очікувану тривалість проекту.
4. Провести необхідні розрахунки:
 - а) визначити, якій тривалості відповідають 85, 90, 99 відсоткові ймовірності завершення проекту;
 - б) визначити ймовірність завершення проекту на два дні швидше та на два дні довше за розраховану тривалість;
 - в) обрати некритичний шлях в проекті та визначити з якою ймовірністю він може затримати проект.

Контрольні питання

1. Що таке критичне критичний шлях проекту?
2. Що таке ранній початок роботи, раннє завершення, пізній початок і пізнє завершення?
3. Що таке резерв часу? Які бувають резерви часу?

Лабораторна робота № 9 Створення проекту та календарне планування робіт

Мета роботи

Одержання практичних навичок створення проекту і планування робіт із застосуванням програмного забезпечення по керуванню проектами.

Основні теоретичні відомості

ProjectLibre 2.5 [6] – це програмний пакет загального призначення для керування проектами. ProjectLibre дозволяє швидко спланувати проект, наочно представити його структуру, виявити "вузькі місця", випробувати різні сценарії й варіанти, одержати набір проектної документації, що відповідає загальноприйнятим світовим стандартам. При цьому ProjectLibre охоплює як управлінські функції («хто, коли й що повинен робити»), так і вартісні характеристики проекту.

На рис. 9.1 наведена екранна форма, яка ілюструє основні елементи інтерфейсу системи ProjectLibre.

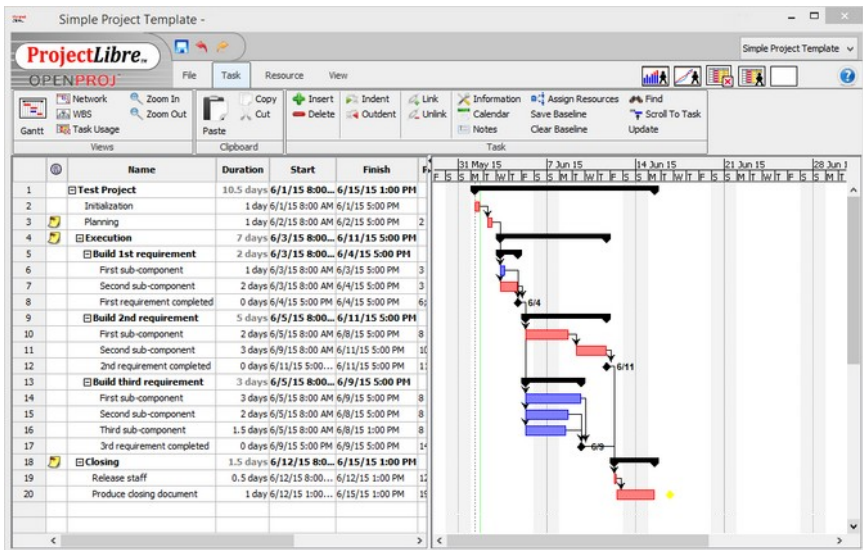


Рисунок 9.1 – Інтерфейс системи

Створення нового проекту

Для створення нового проекту скористайтеся меню *[File » New]* (рис. 9.2).

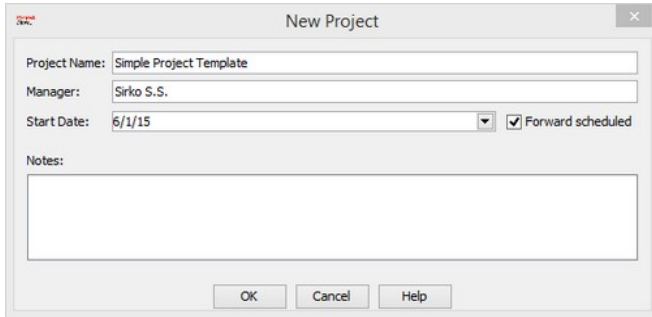


Рисунок 9.2 – Вікно створення нового проекту

У діалоговому вікні, що відкрилося, додайте наступні дані:

- назву проекту,
- менеджера (ПІБ студента),
- дату початку (або дату закінчення) проекту,
- тип розкладу (при плануванні вперед потрібно вказати дату початку, а при плануванні назад – дату закінчення проекту).

Більш детальну інформацію про проект можна додати за допомогою меню *[File » Project » Information]* (рис. 9.3).

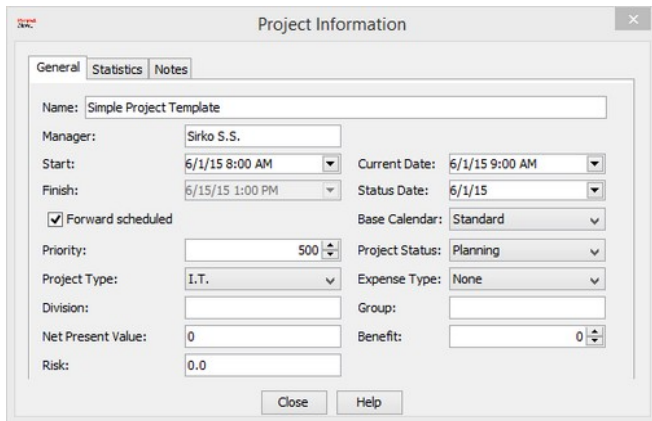


Рисунок 9.3 – Вікно інформації про проект

У цьому ж діалоговому вікні на вкладці *Статистика* можна ознайомитися з поточною статистикою по відкритому проекту: вартість та тривалість проекту, поточний відсоток виконання тощо.

Календар проекту

Календар (Calendar) – це інструмент формування розкладу проекту, що дозволяє встановлювати розподіл робочого й неробочого часу завдань і ресурсів.

За допомогою меню [*File » Project » Calendar*] можна налаштувати робочі календарі для проекту (рис.9.4-9.5): задати час початку роботи, кількість робочих годин на день, тиждень та місяць, а також розклад змін при змінній роботі.

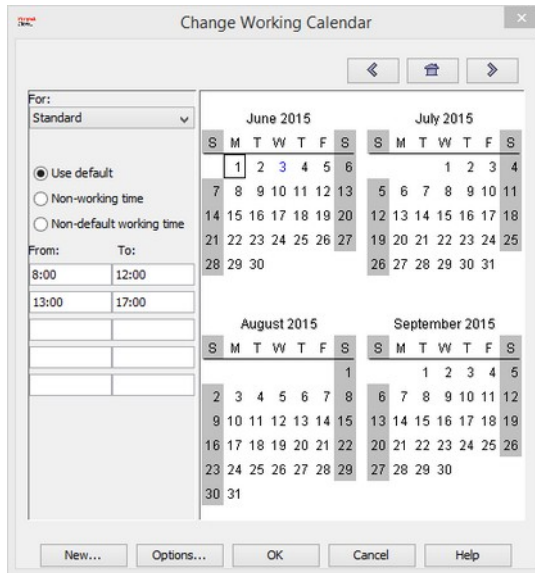


Рисунок 9.4 – Робочий календар проекту

ProjectLibre надає три усталених календарі: стандартний календар (п'ятиденка), 24-годинний календар і календар з нічними змінами. Зазвичай в якості базового використовується стандартний календар.

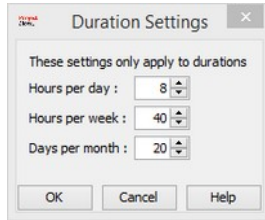


Рисунок 9.5 – Налаштування тривалості робочого часу

Види

Види дозволяють розглядати проект із різних точок зору. В ProjectLibre існує дві основні **категорії видів** (вкладка *[View]* або пункти меню *[Task]* і *[Resource]*):

- вид завдань;
- вид ресурсів.

В одному й тому самому вікні ProjectLibre можна виводити комбінації видів за допомогою пунктів у меню *[View » Sub-views]*.

Усталеним видом проекту є *Подання Гантта* (рис.9.1).

Завдання проекту

Планування завдань проекту тією чи іншою мірою здійснюється в період усього строку виконання проекту. На самому початку життєвого циклу проекту зазвичай розробляється неофіційний попередній план – грубе уявлення про те, що буде необхідно виконати у випадку реалізації проекту.

Рішення про вибір проекту значною мірою ґрунтується на оцінках попереднього плану. Формальне й детальне планування проекту починається після ухвалення рішення про його реалізацію. Визначаються ключові точки проекту, завдання і їх взаємозалежність.

Середовище ProjectLibre включає набір засобів для розробки плану проекту: таблицю для введення завдань і діаграму Гантта *[Task » Views » Gantt]*, графічне подання ієрархічної структури робіт *[Task » Views » WBS]*, мережний графік *[Task » Views » Network]*.

Діаграма Гантта – це графічне подання всіх завдань, які складають проект. Діаграма Гантта складається зі смуг, орієнтованих уздовж осі часу. Кожна смуга представляє окреме завдання (роботу) проекту, її кінці – моменти початку і завершення роботи, її довжина –

тривалість роботи. Крім того на діаграмі можуть бути відзначені сукупні завдання, відсотки завершення, покажчики залежності робіт, віхи проекту, мітка теперішнього моменту часу тощо.

Завдання (Task) у плані проекту передбачає якусь діяльність, яка необхідна для досягнення конкретних результатів (кінцевих продуктів нижнього рівня, deliverables). На виконання завдань необхідний час. Момент завершення завдання означає факт одержання кінцевого продукту, тобто результату виконання завдань.

Виділяють три основні *типи* завдань.

Зведені або складені завдання (Summary task), що містять декілька підлеглих завдань.

Підлеглий завдання (Subtask) – невеликі завдання, що є частиною якихось більших завдань.

Віхи (Milestone) – це завдання, у яких встановлена нульова тривалість і які позначають проміжні цілі проекту. Віха – ця подія або дата в ході виконання проекту. Вона використовується для відображення стану завершеності тих або інших завдань. У контексті проекту менеджери використовують віхи для того, щоб позначити важливі проміжні результати, які повинні бути досягнуті в процесі реалізації проекту. Послідовність віх проекту називається *планом по віхах*. Важливою відмінністю віхи від завдання є те, що вона не має тривалості.

Організація завдань проекту включає наступні дії.

Визначення завдань – розбивка очікуваних результатів проекту на більш дрібні завдання, які легше піддаються обліку й керуванню.

Визначення послідовності завдань – установа залежності між завданнями й визначення обмежень у властивостях завдань.

Оцінка тривалості завдань і проекту в цілому – оцінка часу, необхідного для завершення всіх завдань проекту.

Для організації завдань проекту в середовищі ProjectLibre необхідно перейти у подання Ганта.

Перший крок – створення начерку робіт (зведених завдань), які необхідно виконати в проекті.

Для створення завдання в таблиці завдань виберіть комірку *Name* і вкажіть назву завдання. Встановіть тривалість завдання або оцінку тривалості завдання (позначається знаком питання).

При подвійному клацанні мишею на завданні або виборі меню [Task » Information] відкриється діалогове вікно *Інформація про*

завдання (рис. 9.6). Тут можна додати більш докладний опис завдання, а також указати коментарі та посилання на відповідні документи.

Для створення *віхи*, укажіть назву й встановіть тривалість завдання в значення 0. Віхи в ProjectLibre на діаграмі Гантта представлені у вигляді ромбів, а не смужок.

Перелік стовпців таблиці *Завдання* можна змінювати за необхідності, натиснувши праву кнопку мишки на заголовку й вибравши в контекстному меню пункт *Вставити стовпець* або *Сховати стовпець*.

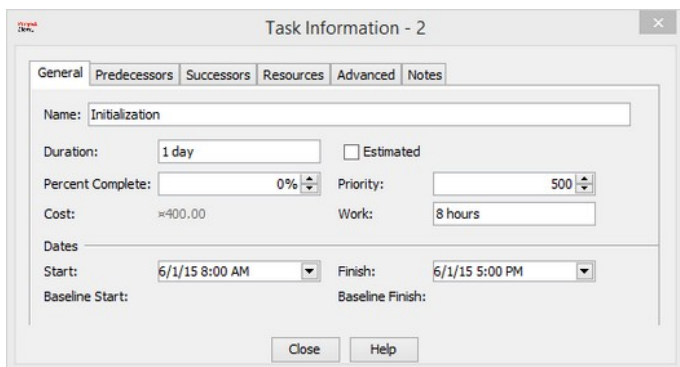


Рисунок 9.6 – Вікно інформації про завдання

Для того щоб **розділити завдання на частині** (Split task) так, щоб її частини можна було виконувати в різний час, або якщо буде потреба перервати завдання на час (відпустка або хвороба виконавця), натисніть праву кнопку миші на завданні в діаграмі Гантта й у спливаючому контекстному меню виберіть пункт *[Split]* (рис.9.7).

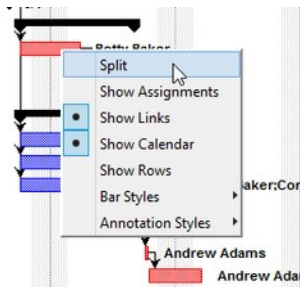


Рисунок 9.7 – Поділ завдання

За замовчуванням в ProjectLibre використовується шкала тижнів і днів для відображення діаграми Гантта. Якщо необхідно побачити укрупнену картину проекту, ви можете перемкнутися на шкалу місяців, років й кварталів і так далі. Для цього скористайтеся кнопками зміни масштабу виводу *[Zoom In]* і *[Zoom Out]* у режимі перегляду завдань.

Структурування завдань

Для структурної декомпозиції завдань проекту після внесення зведених завдань у таблицю необхідно додати підлеглі завдання й включити їх у зведені (рис. 9.1).

Інструмент *структурування* доступний у вигляді пунктів меню *Відступ* і *Виступ* або відповідно *[Task » Indent]* і *[Task » Outdent]*.

Зведене завдання вирівняне вліво, його назва виділена жирним шрифтом, і перед назвою розміщається кнопка *[-]*, що означає *Згорнути*. Відповідні підлеглі завдання розташовані з відступом вправо щодо зведеного завдання.

Зверніть увагу, що будь-яка дія над зведеним завданням – видалення, переміщення або копіювання – застосовується також до всіх вкладених завдань.

Зв'язки між завданнями, затримки й перекриття зв'язків

Зв'язки передування (логічні залежності, Link) між завданнями відображають природу залежностей між ними. Більшість залежностей у проектах відносяться до типу «Закінчення-Початок», при якому наступне завдання може бути виконане тільки після завершення попереднього завдання.

Зв'язки передування утворюють мережну структуру проекту. Сукупність взаємозв'язків між завданнями також називаються логічною структурою проекту, яка визначає послідовність виконання завдань.

Для переходу до календарного планування необхідно визначити зв'язки передування між завданнями проекту. Дати початку або завершення кожного завдання у середовищі ProjectLibre не встановлюються вручну, а визначаються автоматично після встановлення зв'язків між завданнями.

Завдання, яке впливає на інше завдання, називається

Попередником (Predecessor), а завдання, яке залежить від іншого завдання, називається **Послідовником** (Successor). Наприклад, завдання «Програмування» є попередником для завдання «Тестування».

Завдання можуть бути зв'язані чотирма типами зв'язків, наведеними в таблиці 9.1.

Таблиця 9.1 – Типи зв'язків між завданнями

Зв'язок	Тип	Опис	Подання на діаграмі Гантта
Закінчення-Початок (Finish-Start)	FS	Завдання В почнеться відразу чи через N днів після завершення завдання А. Якщо завдання А прострочене, то старт завдання В відкладається до тих пір, поки завдання А не буде закрито	
Початок-Початок (Start-Start)	SS	Завдання В почнеться відразу чи через N днів після старту завдання А. Якщо дата старту завдання А зміниться, то і старт завдання В зміститься відповідно	
Закінчення-Закінчення (Finish-Finish)	FF	Завдання А повинно завершитися перед тим, як завдання В зможе завершитися	
Початок-Закінчення (Start-Finish)	SF	Завдання А повинно початися перед тим, як завдання В зможе завершитися (рідко використовується)	

Іноді буває необхідно вказати **запізнювання** або **випередження** (Lag and Lead, затримка) в зв'язку між завданнями. Затримки можуть бути зазначені в одиницях часу або у відсотках від тривалості

завдання-попередника. При цьому позитивна затримка означатиме запізнювання, а негативна – випередження.

Приміром, між завданнями «Програмування» і «Тестування» повинен пройти один тиждень, щоб з'явився код для тестування. У випадку, коли затримка задається у відсотках, якщо попереднє завдання триває 4 дні, затримка в 25% складатиме 1 день.

Існує декілька способів установки й редагування зв'язків в Projectlibre.

Оберіть завдання, які ви прагнете зв'язати. Завдання з меншим номером ID буде трактуватися як попередник. Потім для установки зв'язку натисніть у меню *[Task » Link]*.

Інший спосіб – установка й редагування зв'язків за допомогою мишки – виконується шляхом перетаскування одного завдання на інше в діаграмі Гантта

За замовчуванням ProjectLibre встановлює зв'язок типу Закінчення-Початок (FS). Для зміни зв'язку необхідно двічі клацнути по стрілці зв'язку на діаграмі Гантта, а потім у вікні *Залежність завдань* (Task Dependancy) обрати необхідний тип зв'язку й указати затримку (рис. 9.8).

Для швидкої установки попередників завдання можна скористатися полем *Predecessors* таблиці **завдань**. У ньому через крапку з комою необхідно вказати ID завдань-попередників. Тип зв'язку встановлюється за замовчуванням FS. У тих випадках, коли необхідний зв'язок іншого типу, крім номера завдання необхідно вказати англійську аббревіатуру типу зв'язку (рис. 9.9).

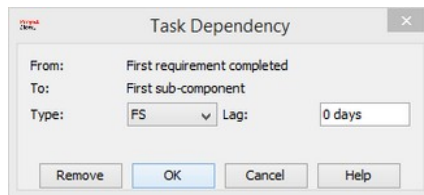


Рисунок 9.8 – Установка типу залежності між завданнями

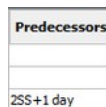


Рисунок 9.9 – Попередники завдань

Також можна редагувати зв'язки в діалоговому вікні *Інформація про завдання*, як показано на рис. 9.10.

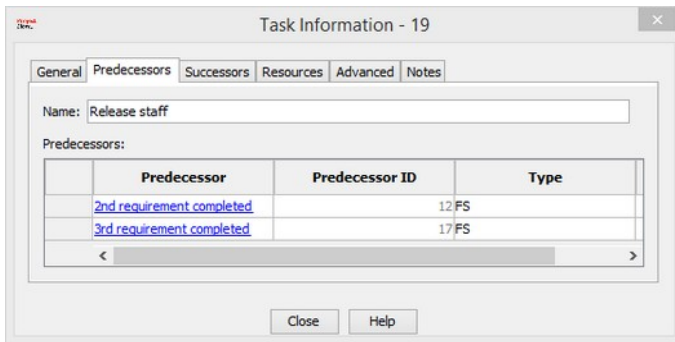


Рисунок 9.10 – Установка попередників для завдання

Щоб видалити зв'язок, виберіть два завдання й скористайтесь меню *[Task » Unlink]*.

Не намагайтеся видалити зв'язок шляхом видалення комірки в стовпці попередника натисканням клавіші , тому що при цьому віддалиться вся завдання.

Обмеження

Деякі завдання можуть бути прив'язані до певної дати. Розглянемо застосування прив'язки завдань до певних дат при складанні плану проекту. Прив'язка завдань до дат у середовищі ProjectLibre виконується за допомогою параметра **Обмеження** (Constraint). Існує 8 типів обмежень (рис. 9.11). Обмеження бувають двох категорій – *гнучке* обмеження й *негнучке* обмеження.

Установка обмеження проводиться в такий спосіб (рис. 9.11).

1. Відкрийте діалогове вікно *Інформація про завдання*.
2. Перейдіть на вкладку *Додатково* й виберіть *Тип обмеження* зі списку.
3. Укажіть значення *Дати обмеження*.

Негнучкі обмеження суттєво обмежують можливість планування та можуть викликати конфлікти між наступним й попереднім завданнями, що може викликати необхідність видалити такі обмеження.

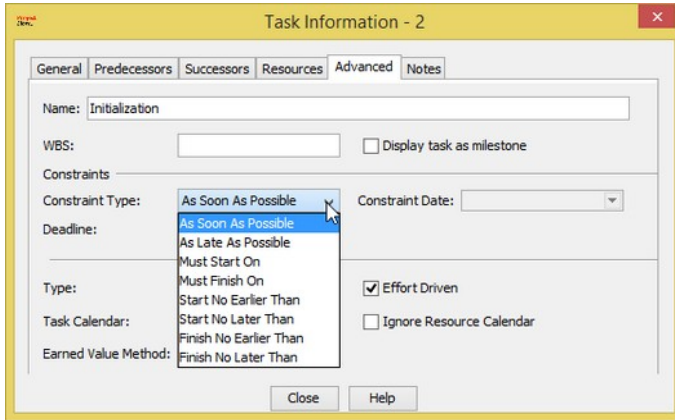


Рисунок 9.11 – Установка обмеження для завдання

Установка контрольного строку

Встановіть дату **контрольного строку** (Deadline), щоб не ставити негнучке обмеження. Установка контрольного строку ніяк не впливає на розрахунок графіка проекту. Контрольний строк позначається на діаграмі Гантта жовтим ромбом. Якщо строк пройшов, а завдання не було виконане, ProjectLibre укаже це в першому стовпці таблиці завдань *Індикатор* (хрестик у червоному кружку, у списку завдань при виводі діаграми Гантта).

Критичний шлях проекту

Критичне завдання (Critical task) – це завдання, що не має резерву часу. Будь-який зсув дати завершення такого завдання на більш пізній строк приведе або до зсуву наступного завдання, або до запізнювання проекту в цілому.

Критичний шлях (Critical path) – це послідовність взаємозалежних критичних завдань, що з'єднує початкову й кінцеву дати проекту. Іншими словами це найбільш довгий маршрут у мережній діаграмі проекту. Критичний шлях визначає мінімально можливу тривалість проекту при заданій структурі. Будь-які затримки на критичному шляху затримують закінчення проекту.

В ProjectLibre на діаграмі Гантта критичні завдання відображаються червоним кольором, а не критичні – синім кольором.

Для некритичних завдань також можна відобразити **часовий резерв** (Total Slack) в контекстному меню на діаграмі Гантта.

Підготовка звітів по проекту

В ProjectLibre не так багато варіантів для формування звітів, якщо порівнювати з комерційними продуктами. Однак подання інформації у звітах проводиться з достатньою якістю.

Дата звіту про стан (або контрольна дата, Status date) – це дата, на яку слід одержати інформацію про стан проекту. За замовчуванням датою звіту про стан вважається поточна дата. Користувач може вибирати контрольну дату довільно, але з однією умовою: вона повинна бути розташована на осі часу «лівіше» поточної дати.

Фільтрація дозволяє в будь-який момент накладати певні умови на спосіб виводу інформації (у тому числі й для звітів). Ви можете сховати інформацію, яку ви не прагнете бачити або виділити важливу інформацію, змінити порядок сортування даних, об'єднати їх у групи (наприклад, по виконанню, критичності, по залучених ресурсах). Для цього використовується пункт меню [*View » Filters*].

Для формування звітів по проекту виберіть пункт меню [*View » Other views » Report*].

Тип звіту – це тип або категорія звіту, що визначає, якого роду інформація про проект повинна бути поміщена в друкований документ. В ProjectLibre передбачено 4 категорії звітів:

- загальні відомості про проект (Project details);
- інформація про ресурси (Resource information);
- інформація про завдання (Task information);
- призначення (Who Does What).

Завдання на лабораторну роботу

1. Використовуючи рекомендовану літературу й дані методичні вказівки, ознайомтеся з основними принципами роботи із програмами для керування проектами в цілому й зокрема із системою ProjectLibre.

2. Ознайомтеся із проектним завданням у додатку Б.

3. Створіть новий проект. Виберіть спосіб планування від дати початку проекту. Уведіть дату початку проекту, скорегувавши строки проектного завдання так, щоб початок проекту збігся з

початком семестру.

4. Налаштуйте календар проекту, у якому святкові дні відзначте як неробочі.

5. Створіть план проекту відповідно до проектного завдання у додатку А: у таблицю завдань додайте завдання, обмеження, тривалості завдань, визначте логічні зв'язки між завданнями.

6. Придумайте й додайте у проект своє невелике завдання.

7. Додайте до плану необхідні затримки й випередження.

8. Порівняйте отримані та установлені у завданні строки виконання етапів робіт.

9. Встановіть контрольний строк завершення останньої завдання проекту трьома днями пізніше планового строку її завершення.

10. Визначте тривалість проекту.

11. Підготуйте наступні звіти: Загальні відомості про проект (Project Details), Завдання проекту (Task Information=>Default), Діаграма Гантта, Мережний графік, Ієрархічна структура робіт проекту, Критичні й некритичні завдання й часові резерви завдань (Task Information=>Schedule CPM).

Контрольні питання

4. Яке призначення систем керування проектами?

5. Як визначається тривалість виконання проекту?

6. Що таке критичне завдання й критичний шлях проекту? Що таке часовий резерв завдання?

7. Які типи логічних залежностей між завданнями існують і як вони впливають на розрахунки календарного плану проекту?

8. Як використовувати запізнювання й випередження при формуванні залежностей між завданнями?

9. Для чого необхідні календарі?

10. Що таке віхи й навіщо вони необхідні?

11. Для чого необхідно структурувати завдання проекту? Що таке ієрархічна структура робіт проекту?

12. Що таке обмеження завдань і які вони бувають?

13. Що таке крайній строк виконання завдання і як його можна використовувати?

14. Яким чином можна врахувати перерви у виконанні завдань?

Лабораторна робота № 10

Планування ресурсів, створення призначень і вартісний аналіз проекту

Мета роботи

Вивчення особливостей планування призначень для співробітників і матеріальних ресурсів, а також принципів розподілу завантаження ресурсів під час виконання завдань, проведення вартісного аналізу проекту.

Основні теоретичні відомості

Ресурси проекту

Ресурси (Resource) – це компоненти, які забезпечують виконання проектних робіт. Вони включають виконавців, матеріали, устаткування, енергію тощо.

Ресурси бувають *двох типів* – робота й матеріали.

Ресурси типу **Робота** (Work) виконують завдання шляхом витрати часу на завдання. Звичайно це люди або устаткування.

Ресурси типу **Матеріал** (Material) – це запаси видаткових матеріалів і компонентів, які потрібні для виконання проекту.

Доступний для виконання роботи набір ресурсів називають *пулом ресурсів*. Після визначення необхідної кількості ресурсів, необхідно встановити час і доступність кожного ресурсу. У випадку трудових ресурсів, кількість часу, яку вони можуть працювати, буде вимірятися в годинах, днях місяцях або роках, а кількості матеріальних ресурсів повинні бути зазначені в одиницях виміру.

Доступність (Availability) – це період часу, протягом якого ресурс може бути використаний при виконанні конкретного завдання.

Доступність ресурсу визначається трьома факторами:

- календарями робочого часу проекту й даного ресурсу;
- датами початку й завершення використання ресурсу в проекті;
- сумарним обсягом призначень ресурсу на завдання проекту (максимальне використання).

Створення пулу ресурсів проводиться у *Виді ресурсів*, який дозволяє керувати списком ресурсів та витратами на ресурси.

Заповнення списку ресурсів

Уведення ресурсів проводиться через таблицю *Виду ресурсів* (рис.10.1).

1. Для доступу до таблиці ресурсів скористайтеся меню [*Resource » Resources*] або [*View » Resource views » Resources*].

2. Виберіть комірку в колонці *Назва* й укажіть назву ресурсу.

3. Укажіть тип ресурсу – *Робота* або *Матеріал*. Для матеріалів укажіть *одиниці виміру*.

	Name	Type	Material Label	Max. Units	Standard Rate	Overtime Rate	Cost Per Use	Accrue At
1	Andrew Adams	Work		100%	¥50.00/hour	¥50.00/hour		¥0.00 Prorated
2	Betty Baker	Work		100%	¥12.00/hour	¥18.00/hour		¥0.00 Prorated
3	Cory Clark	Work		100%	¥10.00/hour	¥15.00/hour		¥0.00 Prorated
4	coffee	Material	cup		¥5.00/hour			¥0.00 Prorated
5	sugar	Material	kg		¥0.00			¥20.00 Prorated

Рисунок 10.1 – Вид списку ресурсів

У стовпці **Максимальне використання** (Maximum units) показане максимальне завантаження, яке можливе для ресурсу при виконанні будь-яких завдань у певний період часу. За замовчуванням використовується формат відсотків. Наприклад, якщо зазначено 100%, це означає, що цей ресурс буде працювати всі вісім годин у звичайний робочий день. Якщо зазначено 50%, то цей ресурс буде працювати по чотири години на день. У випадку пулу ресурсів, що складається з 5 працівників, їх максимальна кількість одиниць буде 500%.

Ресурси типу *Матеріал* не містять величини *Максимальне використання*. Одиниці таких ресурсів містять вартість споживання, яка може бути фіксованою або змінною.

При клацанні в таблиці ресурсів по назві ресурсу буде виведене вікно, що містить інформацію про ресурс (рис.10.2).

На вкладці *General* можна вказати індивідуальний календар ресурсу або змінити робочий час ресурсу.

Для редагування календаря ресурсу натисніть кнопку із зображенням календаря праворуч від поля вибору календаря зі списку й внесіть необхідні зміни.

The image shows a 'Resource Information' dialog box with the following fields and values:

Field	Value
Name	Andrew Adams
Initials	AA
E-mail Address	andrew@noemail.com
Group	
RBS	
Type	Work
Base Calendar	Standard
Material Label	

Buttons: Close, Help

Рисунок 10.2 – Вікно з інформацією про ресурс

У поле *Group* можна вказати професійну приналежність трудового ресурсу або вид матеріального ресурсу (наприклад, назву відділу або назву групи матеріалів).

Методи нарахування

Плануючи вартість проекту, необхідно не тільки визначити його бюджет, але й визначити, як цей бюджет буде витрачатися протягом усього проекту. Витрата бюджету залежить від порядку оплати робіт.

Оплатувати роботу можна різними способами: може використовуватися передплата, оплата за фактом завершення, а іноді, і пропорційна оплата в міру виконання робіт, причому звичайно в проєкті можуть використовуватися кілька способів оплати.

Метод нарахування витрат (Accrual method) – це метод обліку разових або фіксованих витрат, що входять у вартість роботи. В ProjectLibre передбачено три методи нарахування витрат:

- на дату початку завдання (start);
- на дату завершення роботи (end);
- метод рівномірного розподілу на період виконання завдання (prorated).

Слід зауважити, що порядок оплати робіт не впливає на загальну суму витрат по проєкту.

Визначення витрат на ресурси

В ProjectLibre є два типи витрат, що визначають вартість ресурсів проєкту: **погодинна ставка** (стандартна та понаднормова ставка) і **витрати за використання**.

Таблиця норм витрат (Cost rate table) – це набір значень витрат на ресурс, що включає стандартний і понаднормовий тариф оплати ресурсу, вартість разового використання ресурсу, а також інтервали часу, протягом яких діє той або інший тариф.

Вартість ресурсу визначається у відповідних стовпцях таблиці у виді *Ресурси* або у вікні *Інформації про ресурс* на вкладці *Costs*. Ви можете вказати до 5 різних тарифних ставок для обраного ресурсу (A-E). У кожній з них ви можете вказати наступні ставки:

- стандартна ставка;
- ставка за переробку;
- ставка за кожне використання ресурсу.

Щоб додати витрати на ресурс виконайте наступне.

1. Відкрийте діалогове вікно *Інформація про ресурс* і виберіть вкладку *[Costs]* (рис.10.3).

The screenshot shows the 'Resource Information' dialog box with the 'Costs' tab selected. The 'Name' field contains 'Andrew Adams'. Below it, the 'Cost rate tables' section contains a table with 4 columns: 'EffectiveDate', 'Standard Rate', 'Overtime Rate', and 'Cost Per Use'. The table has 5 rows, with the first row containing data: '1/1/70 12:00 AM', '>50.00/hour', '>50.00/hour', and '>0.00'. Below the table, there is a dropdown menu for 'Accrue At' set to 'Prorated'. At the bottom of the dialog are 'Close' and 'Help' buttons.

EffectiveDate	Standard Rate	Overtime Rate	Cost Per Use
1/1/70 12:00 AM	>50.00/hour	>50.00/hour	>0.00

Рисунок 10.3 – Вікно інформації про ресурс, вкладка «Вартість»

2. Дата початку дії тарифу (Effetive date) може збігатися з датою початку проекту. Якщо це не так, укажіть дату або виберіть її з календаря.

3. Укажіть *стандартну ставку* (Standart rate). За замовчуванням вказується значення в грн/година, але ви можете при необхідності вказати ставку в /рік, або в /місяць.

4. Укажіть *ставку за переробку* (Overtime rate). Її необхідно вказувати, навіть якщо вона дорівнює стандартній ставці.

5. Укажіть *додаткову ставку за використання ресурсу* (вартість доставки й таке інше), яка не залежить від того, скільки часу

задіяний ресурс у завдання.

6. Якщо ви знаєте, що ставки можуть мінятися під час виконання проекту, клацніть на другому рядку, укажіть ефективну дату дії й внесіть нові стандартну ставку, ставку за переробку й ставку за використання. Ви можете вказати або абсолютні величини, або збільшення або зменшення в % від ставок, перерахованих у першому рядку.

7. Укажіть метод нарахування шляхом вибору зі списку, що випадає [*Accrue at*].

При створенні призначення дані про ставку ресурсу за замовчуванням беруться з таблиці А норм витрат. Змінити вартість призначення можна, указавши іншу таблицю норм витрат для потрібного ресурсу. Таблиця норм витрат вказується у вікні *інформації про завдання* на вкладці *Resources* колонка *Cost Rate Table*.

Призначення ресурсів на завдання

Наступний крок після визначення ресурсів – **призначення ресурсів на завдання**. **Призначення** (Assignment) – це елемент розкладу проекту, що відбиває взаємозв'язок між завданням і ресурсом, призначеним для її виконання. Змістовна інтерпретація цього терміну залежить від «кута зору»: з погляду розподілу ресурсів – це величина працезатрат ресурсу для виконання конкретної завдання; з погляду параметрів завдання – це її трудомісткість (обсяг робіт), що доводиться на даний ресурс.

Для того щоб призначити ресурс на завдання, необхідно виконати наступні дії.

1. На діаграмі Гантта виберіть завдання, на яке потрібно призначити ресурс, і виберіть меню [*Task » Assign Resources*]. Або відкрийте вікно *інформації про завдання*. Перейдіть на вкладку *Resources* і натисніть кнопку [*Assign Resources*].

2. У вікні, що відкрилося (рис. 8.4) виберіть ресурс, який ви прагнете призначити на завдання.

3. Для робочих ресурсів укажіть відсоток участі в стовпці *Units*. **Одиниці** (обсяг призначень, units) для робочих ресурсів – це кількість виконавців (або часток ресурсу), призначених на завдання.

4. При призначенні матеріальних ресурсів можна вказати як фіксований обсяг виділюваних на завдання ресурсів (наприклад, 2

коробки), так і змінний обсяг (наприклад, 1 коробка в день). Для матеріальних ресурсів виберіть поле *Одиниці*, натисніть <F2> і вкажіть кількість призначуваних одиниць або дані у форматі: число одиниць/позначення тривалості (3/d або 3/д для позначення витрати трьох одиниць матеріалу в день). Якщо ви залишите поле порожнім, буде встановлене значення за замовчуванням 1. ProjectLibre додасть *одиниці виміру* до зазначеного вами десяткового значення.

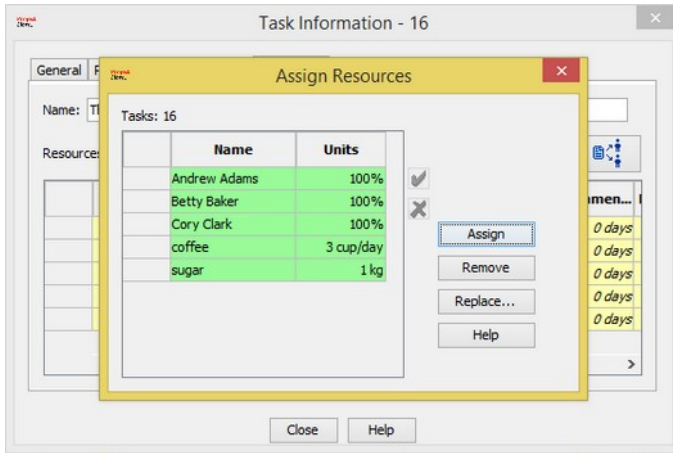


Рисунок 10.4 – Призначення ресурсів на завдання

5. Натисніть кнопку *[Assign]* для підтвердження.

Щоб вилучити призначення з завдання, виберіть завдання, потім назву ресурсу, який потрібно вилучити, і натисніть клавішу .

Після призначення ресурсів на завдання ProjectLibre визначає працезатрати кожного з ресурсів для виконання цього завдання й планує розподіл цих працезатрат на кожний із днів протягом усієї тривалості завдання.

В ProjectLibre **тривалість** (duration) відповідає часу, через який буде отриманий результат завдання або іншими словами кількість часу між початком і закінченням завдання, а **працезатрати** або **робота** (work) відповідають часу, витраченому відповідальним виконавцем для одержання результату.

При призначенні ресурсів на завдання ProjectLibre перемножує тривалість завдання (D) на зазначені відсотки або одиниці виміру ресурсів (U) для визначення робочих годин (W), які будуть витрачені

на завдання ($D \cdot U = W$).

При першому призначенні ресурсів на завдання, працезатрати обчислюються і фіксуються.

Наприклад, для розробки деякого модуля програми встановлена тривалість в 5 днів і призначено 3 ресурси із завантаженням 100%. Припускаючи, що в дні за замовчуванням 8 робочих годин, кожний із цих ресурсів повинен буде працювати по 8 годин усі 5 днів. Повні працезатрати на завдання становлять 120 людино-годин.

При повторному призначенні ресурсів на завдання буде проведений перерахунок тривалості й роботи залежно від *Типу завдання*.

Типи завдань

ProjectLibre визначає **три типи завдань** (Task type), які дозволяють перерахувати значення змінних, коли на завдання робиться нове призначення ресурсів:

- фіксовані одиниці / фіксований обсяг ресурсів (Fixed Units);
- фіксовані працезатрати (Fixed Work);
- фіксована тривалість (Fixed Duration).

Тип завдання визначає, як редагування однієї із властивостей завдання – тривалості, працезатрат або призначень – буде впливати на дві інші властивості. Обраний тип завдання визначає, значення якої із властивостей буде зафіксовано. Тривалість – це змінна завдання, а кількість одиниць і працезатрати – це змінні призначення.

Розглянемо кожний з типів завдань.

Тип завдання **Fixed Duration** означає, що зміна працезатрат або числа призначених на виконання завдань співробітників не змінить її тривалість

Тип **Fixed Work** (Фіксовані працезатрати) установлюється для завдань, виконання яких вимагає строго певних працезатрат. Чим більше співробітників призначено на завдання, тем менший обсяг роботи припадає на кожного з них і тем швидше завдання буде виконане.

Тип **Fixed Units** (Фіксований обсяг ресурсів) використовується в плані проекту для завдань, тривалість і обсяг роботи для яких повністю залежать від виділених на них ресурсів. Тривалість завдань із фіксованим обсягом ресурсів дорівнює працезатратам діленим на

число призначених ресурсів.

У випадку якщо завдання не відноситься до типу *Фіксовані працезатрати*, додатково до позначення типу завдання можна використовувати прапорець **Фіксованого обсягу робіт** (effort-driven). Якщо включити дану ознаку, то працезатрати будуть зафіксовані одночасно з однією із двох інших ознак завдання: тривалістю або обсягом ресурсів.

Керівник проекту повинен проаналізувати кожне завдання проекту і вибрати для нього відповідний тип.

Нехай існує завдання тривалістю 5 днів. На нього призначено один трудовий ресурс обсягом 100%. Таким чином, працезатрати на завдання становлять 40 год. Розглянемо приклад в табл. 10.1, як тип завдання впливає на зміни.

Таблиця 10.1 – Приклад завдань трьох типів

Тип завдання	Варіант змін	Результат
фіксована тривалість	Тривалість – 10 днів	Працезатрати – 80 годин
	Працезатрати – 48 годин	Обсяг призначення ресурсу – 120%
	Обсяг призначення ресурсу – 50%	Працезатрати – 20 годин
	Додаємо аналогічний ресурс	Працезатрати – 80 годин
фіксовані працезатрати	Тривалість – 10 днів	Обсяг призначення ресурсу – 50%
	Працезатрати – 48 годин	Тривалість – 6 днів
	Обсяг призначення ресурсу – 50%	Тривалість – 10 днів
	Додаємо аналогічний ресурс	Тривалість – 2,5 дня
фіксований обсяг ресурсів	Тривалість – 10 днів	Працезатрати – 80 годин
	Працезатрати – 48 годин	Тривалість – 6 днів
	Обсяг призначення ресурсу – 50%	Тривалість – 10 днів
	Додаємо аналогічний ресурс	Працезатрати – 80 годин

За замовчуванням в ProjectLibre тип завдання встановлений в значення «фіксований обсяг ресурсів» (Fixed Units) і відзначений прапорець «фіксований обсяг робіт». Тривалість завдання при цьому залежить лише від кількості призначених на неї ресурсів. Якщо ви зміните кількість ресурсів після початкового призначення, ProjectLibre змінить тривалість, але залишить вихідний обсяг робіт.

Щоб змінити тип завдання необхідно виконати наступне: відкрийте вікно *інформації про завдання*, виберіть вкладку *Advanced* і в списку вибору *Type* виберіть необхідний тип завдання (рис. 10.5).

Ви можете вивести інформацію про призначення ресурсів на завдання за допомогою пункту меню [*Views* » *Task Usage*].

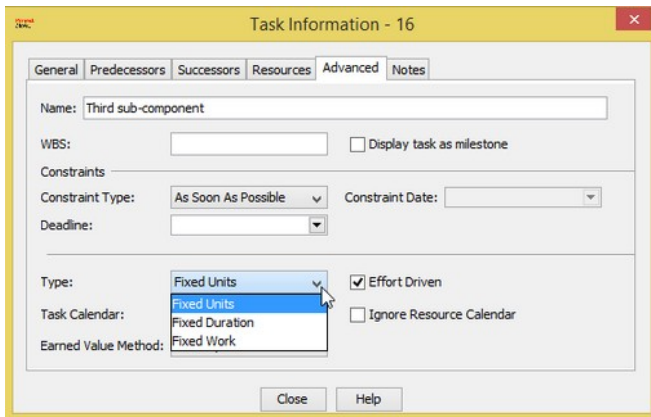


Рисунок 10.5 – Зміна типу завдання

Визначення вартості проекту

Наступний етап – це формування витрат проекту. Витрати проекту складаються з вартості використання ресурсів (як матеріальних, так і трудових) і фіксованих витрат на завдання.

Фіксовані витрати на завдання (Fixed cost) – це витрати, не пов'язані з використанням проектних ресурсів.

Для впровадження фіксованих витрат використовується поле *фіксованих витрати* в таблиці завдань (стовпець фіксованих витрат можна вставити натиснувши правою кнопкою миші на заголовках стовпців у таблиці й вибравши *Insert Column* => *Fixed Cost*) (рис. 8.6).

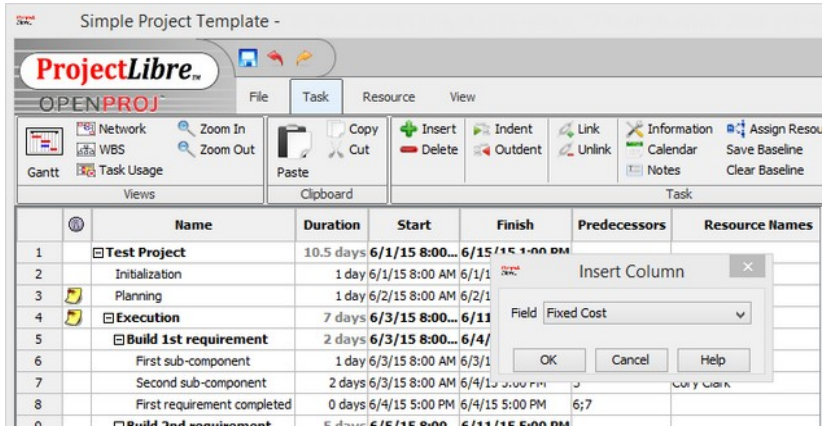


Рисунок 10.6 – Додавання стовпця Фіксованих витрат

Контури

За замовчуванням при створенні призначення працездатності розподіляються рівномірно по всій тривалості завдання.

Контур (*Work Contour*) або профіль завантаження в ProjectLibre – це форма розподілу робіт усередині призначення. Звичайно, якщо ресурсу призначають 100% використання в 40-годинному завданні, завдання буде розпочато негайно з роботою по 8 годин на день впродовж 5 днів. Це – *плоский контур* за замовчуванням, де робота кожного ресурсу рівномірно розподіляється по всій тривалості завдання.

Іноді вам може знадобитися змінити розподіл роботи над завданням. ProjectLibre містить 8 визначених контурів (рис.8.7).

Розглянемо особливості кожного контуру:

- *плоский* – контур за замовчуванням з рівномірним розподілом роботи;
- *завантаження наприкінці* – пік активності виникає наприкінці завдання;
- *завантаження на початку* – пік активності виникає на початку завдання;
- *подвійний пік* – у завдання існують два піки активності;
- *ранній пік* – пік активності на початковому етапі але з рампою до піка;
- *пізній пік* – пік активності наприкінці проекту, але з рампою;

- «дзвін» – одиночний пік активності в середині проекту;
- «черепаха» – плоске завантаження без виражених піків, з рампами на початку й наприкінці.

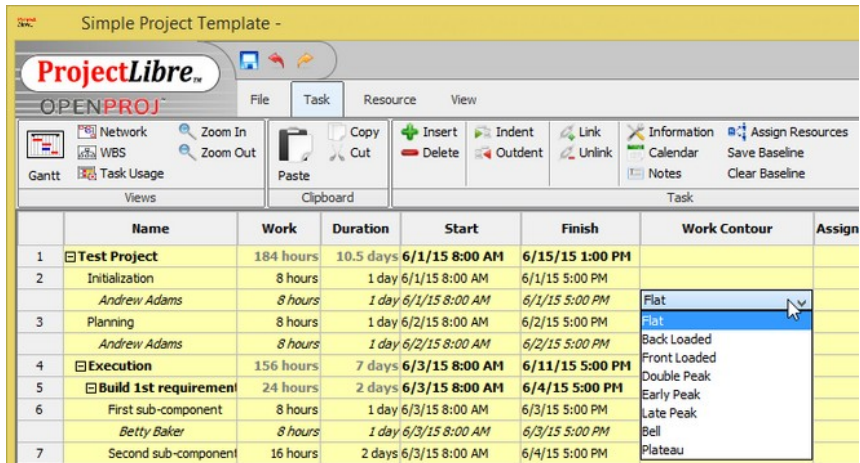


Рисунок 10.7 – Вибір профілю завантаження ресурсу

Змінити робочий контур для призначення можна у поданні *Використання завдань* (Task Usage).

Базовий план проекту

Базовий план проекту (Baseline plan) – це розклад, вартість і зміст проекту, які використовуються в якості еталонних при керуванні ходом виконання проекту. Базовий план використовується для виконання аналізу поточної продуктивності в порівнянні із затвердженим планом.

По закінченню планування й до початку виконання зберігається «знімок» проекту як базового плану ([File »Project » Save Baseline]).

Крім того в проекті може бути кілька базових планів (до 10), які звичайно зберігаються в найбільш важливі ключові моменти проекту.

При збереженому базовому плані на діаграмі Гантта під кожним завданням відображаються тіні, що відповідають базовому розкладу.

Графіки

В ProjectLibre є можливість спостерігати за вартістю й роботою

проекту за допомогою графіків.

Для того щоб переглянути графіки, необхідно наступне.

1. Вибрати пункт меню [View » Sub-views » Charts].
2. У таблиці завдань вибрати необхідні завдання (чи всі завдання проекту).
3. У панелі, що з'явилася знизу головного вікна, вибрати необхідні ресурси.
4. Налаштувати необхідні параметри для відображення: вибрати перемикач робота/вартість і зі списку вибрати відображувані величини.

Наприклад, розглянемо Графік витрати бюджету проекту (рис.10.8).

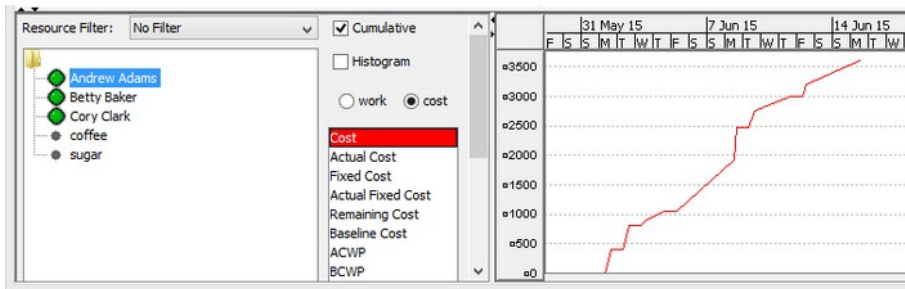


Рисунок 10.8 – Графік витрати бюджету

Точно так само можна переглядати графіки, що стосуються робіт проекту, що залишилися для виконання. Для цього необхідно змінити перемикач у положення «робота».

Завдання на лабораторну роботу

1. Відкрийте проект, створений у лабораторній роботі №9.
2. Створіть список трудових ресурсів, які будуть використовуватися в проекті згідно із проектним завданням. Придумайте і додайте в проект два матеріальні ресурси.
3. Уведіть стандартні ставки для трудових ресурсів проекту відповідно до проектного завдання. Понаднормові ставки встановіть в 1,5-2 рази більше стандартних.
4. Призначте ресурси на завдання проекту. При виконанні призначення використовуйте матрицю завдань і відповідальності

проекту (див. табл. Б.5). Вмовимося, що основна відповідальність означає 100% призначення ресурсу на завдання, а допоміжна – від 1 до 99% (залежить від завдання). Зверніть увагу, що тривалість виконання завдань за проектним завданням в даній роботі є фіксованою. При цьому прапорець Фіксованого обсягу робіт вмикати не потрібно. Встановити тип завдання необхідно до призначень ресурсів.

5. Для завдання, створеного у лабораторній роботі №9 п.6, введіть фіксовану вартість.

6. Визначите час участі ресурсів у проекті.

7. Визначите вартість етапів робіт і проекту в цілому.

8. Збережіть базовий план проекту №1.

9. Порівняйте отриману вартість проекту із затвердженим бюджетом. Розбіжність із затвердженим бюджетом у даній роботі є можливою.

10. Підготуйте наступні звіти: Загальні відомості про проект (Project Details), Використання завдань (Task Usage), Список ресурсів (Resource information=>Default), Використання ресурсів (Who does what=>Basic report), Рух коштів по проекту (Task Information=>Cost, Baseline), Графік витрати бюджету.

11. Створіть копію отриманого плану проекту й проведіть серію експериментів для вивчення властивостей призначень:

- 1) визначите персональний календар для одного ресурсу проекту;
- 2) нехай один з виконавців не працює над проектом по п'ятницях в одному з місяців (наприклад, у жовтні), а інший в ці дні працює половину робочого дня;
- 3) розподіліте завантаження ресурсів під час виконання завдань за допомогою профілів, визначте різні профілі завантаження для деяких завдань;
- 4) проведіть дослідження по вивченню впливу властивостей завдань різних типів на параметри призначення;
- 5) реалізуйте перерву у виконанні однієї із завдань;
- 6) внесіть зміни ставки співробітника з деякої встановленої дати до завершення проекту, визначте різні норми витрат для ресурсу, призначеного на кілька різних завдань.

12. Включіть у звіт результати трьох експериментів і проаналізуйте отримані результати.

13. Підготуйте звіт за лабораторної роботи.

Контрольні питання

1. Що таке ресурси і яких типів вони бувають?
6. Що таке доступність ресурсу?
7. Для чого необхідні календарі ресурсів?
8. Що таке призначення?
9. Поясніть три типи завдань: фіксовані одиниці, фіксовані працезатрати та фіксована тривалість.
10. Як визначаються витрати на завдання?
11. Що таке фіксовані витрати на завдання?
12. Що таке метод нарахування при розрахунках витрат?
13. Що таке контур або профіль завантаження?
14. Як визначається вартість проекту?
15. Що таке бюджет проекту й що таке витрата бюджету?
16. Що таке базовий план проекту?

Лабораторна робота № 11 **Аналіз завантаження ресурсів і оптимізація параметрів проекту**

Мета роботи

Отримання практичних навичок проведення аналізу та оптимізації завантаження ресурсів та оптимізації параметрів проекту.

Основні теоретичні відомості

Перевантажені ресурси

Перевантаження ресурсів – це перевищення в процесі планування максимально можливого для цього ресурсу обсягу призначення, передбаченого в певний проміжок часу.

Поява в проекті перевантаженого ресурсу може бути викликана наступними причинами:

- призначення ресурсу на завдання в кількості, що перевищує максимально припустимий обсяг призначень;
- одночасне призначення ресурсу на дві або більш завдання, у результаті чого сумарний обсяг призначень перевищує максимально припустимий;
- призначення ресурсу на завдання, що виконується в період недоступності ресурсу (у його неробочий час).

Перевищення завантаження ресурсу можна знайти, відкривши додаткову панель із виводом *Гістограми завантаження ресурсів*. Графік завантаженості ресурсу представляє собою стовпчасту діаграму, у якій на горизонтальній осі відображаються дати, а на вертикальній осі вказується обсяг призначень ресурсу (рис. 11.1). Чорною лінією позначається рівень доступності ресурсу, тобто припустимий обсяг призначень. Рівень завантаження ресурсу, що перевищує встановлене для нього максимальне значення, виступає на діаграмі за границю чорної лінії.

Шляхів усунення перевищень завантаження ресурсів

Вирівнювання ресурсів – це процес реорганізації плану проекту з метою ліквідації перевантаженості його ресурсів.

При виявленні перевищення завантаження ви можете:

- змінити розподіл працезатрат через профіль завантаження;

- затримати виконання завдання або вставити в завдання перерву;
- змінити обсяг призначень ресурсу на завдання;
- реорганізувати мережний графік робіт;
- замінити перевантажений ресурс іншим вільним;
- урахувати наднормативні працезатрати ресурсів як понаднормові (не більш 2 – 4 годин на добу).

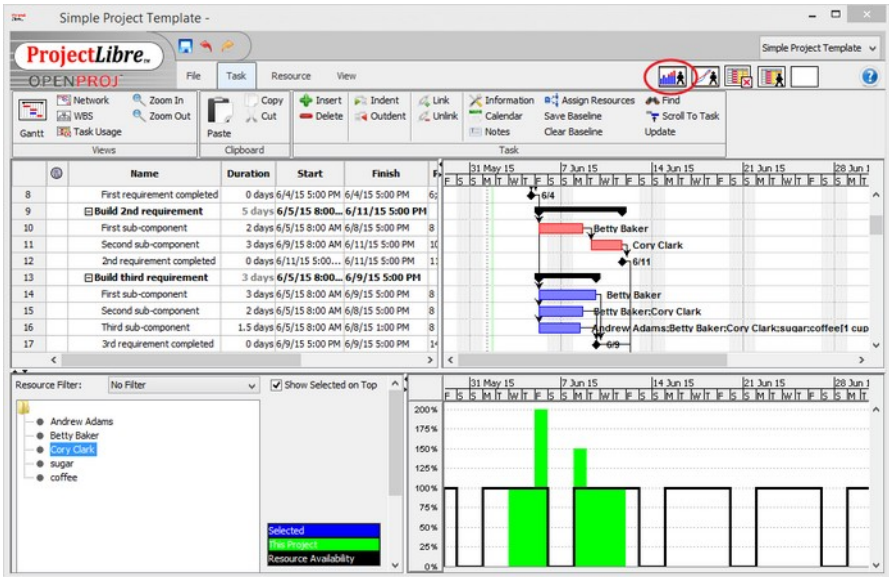


Рисунок 11.1 – Подання перевантаженого ресурсу

У деяких випадках може виявитися досить корисним змінити робочий час ресурсу за допомогою календаря ресурсу або уточнити періоду часу, протягом якого даний ресурс буде задіяний для виконання конкретного завдання. Справа в тому, що за замовчуванням ProjectLibre вважає, що призначений на завдання ресурс бере участь у його виконанні протягом всього часу завдання, від першої до останньої хвилини. Щоб указати період часу, протягом якого ресурс реально повинен брати участь у виконанні завдання, можна скористатися полем *Assignment Delay* в призначенні Ресурсу.

Вибір способу вирівнювання перевантаженого ресурсу прямо залежить від причини перевантаження.

Оптимізація параметрів проекту

Після того, як сформовано план проекту й обчислено його часові параметри й вартість, при наявності часових або грошових резервів можна спробувати поліпшити його показники:

- закінчити проект раніше спочатку наміченого строку при збереженні сформованого бюджету;
- підвищити якість робіт (залученням кращих ресурсів) за рахунок збільшення бюджету, при збереженні строків виконання;
- підвищити якість робіт за рахунок збільшення часу, що приділяється на них, при збереженні бюджету;
- зберегти розраховані строки виконання робіт, забезпечивши економію матеріальних засобів.

Однак значно більш складні проблеми виникають у тих випадках, коли розроблювач проекту одержує план, параметри якого не вміщаються в часові й/або бюджетні рамки.

Для вирішення позначених проблем використовують методи оптимізації строків і вартості проектних робіт. Класичними методами аналізу й оптимізації плану робіт є методи PERT (Program Evalution and Rewiew Technique, техніка оцінки й перегляду програм) і метод критичного шляху CPM (Critical Path Method).

Оптимізація строків проекту

При оптимізації часових параметрів проекту необхідно визначити критичний шлях і спробувати скоротити його. Для цього можна застосувати наступні способи:

- перекриття завдань;
- планування переробки й/або зміна календарів ресурсів;
- поділ тривалих завдань на більш дрібні;
- пошук і виправлення помилок планування.

Як самий радикальний метод зміни строку завершення проекту можна застосувати зміну дати початку проекту.

Оптимізація бюджету

При оптимізації бюджету доцільно дотримуватися наступного порядку дій.

1. Виявити завдання, що викликають перевищення бюджету.

2. Визначити наявність резервів часу для завдань, які не укладаються в бюджет.

3. Зберегти резервну копію плану,

4. Скорегувати состав і розподіл ресурсів з метою зниження витрат.

5. Оцінити результати оптимізації плану.

Засоби, які здатні допомогти менеджерів виявити недоліки проекту на етапі планування: фільтри для відбору завдань із перевищенням витрат; звіти про витрати.

Завдання на лабораторну роботу

1. Відкрийте проект, створений у лабораторній роботі №10.

2. Отримайте список ресурсів з перевищенням завантаження. Якщо перевантажені ресурси не виявлені, створіть варіант (копію) плану, у якому присутні як мінімум два перевантажені ресурси з урахуванням наступних обставин:

- ресурсу призначений на завдання в кількості, що перевищує його максимально припустимий обсяг призначень;
- одночасне призначення ресурсу на два або більше завдання;
- призначення ресурсу на завдання, що виконується в період його недоступності.

3. Виконайте аналіз плану для кожного перевантаженого ресурсу та сформулюйте способи вирівнювання.

4. Усуньте перевантаження ресурсів.

5. Підготуйте звіт щодо проведеної оптимізації.

6. Збережіть новий базовий план проекту.

7. Створіть копію отриманого плану проекту й проведіть серію експериментів:

1) перенесіть дату початку проекту на тиждень уперед; внесіть зміни в план проекту таким чином, щоб його тривалість зменшилася й проект уклався в строк;

2) визначте критичний шлях проекту; вивчіть специфіку завдань критичного шляху; змініть план так, щоб зменшити число завдань на критичному шляху;

3) скоротіть тривалість критичного шляху проекту за рахунок перегляду і зміни типів залежностей між завданнями;

4) скоротіть тривалість критичного шляху проекту за

рахунок планування робіт у понаднормовий час;

5) скоротіть тривалість критичного шляху проекту за рахунок призначення додаткових ресурсів.

Додайте до звіту результати трьох експериментів і проаналізуйте отримані результати.

8. Проведіть огляд витрат в отриманому плані проекту. Визначте завдання з перевищенням бюджету.

9. Оптимізуйте план у випадку розбіжності вартості етапів і робіт із затвердженими витратами, наведеними в проектному завданні.

10. Сформуйте звіт про рух коштів по проекту.

11. Створіть копію отриманого плану проекту й проведіть серію експериментів:

1) внесіть зміни, у результаті яких отриманий бюджет проекту перевищує можливості замовника; тепер вам необхідно зменшити бюджет на 10%; для цього ваше керівництво дозволяє використовувати більш низькі таблиці норм витрат у ресурсів;

2) визначите, на який з матеріальних ресурсів іде найбільше коштів, і яка повинна бути вартість ресурсу щоб знизити проектні витрати на 5%; у випадку, якщо вартість ресурсів становить менше 5% проектних витрат, збільшити вартість ресурсів для проведення аналізу.

Додайте до звіту результати одного експерименту і проаналізуйте отримані результати.

12. Підготуйте звіт за лабораторної роботи.

Контрольні питання

1. Перелічіть причини появи перевантаження ресурсів.
2. Яким чином усувається перевантаженість ресурсів?
3. Які дії необхідно виконати для оптимізації бюджету проекту?
4. Які дії необхідно виконати для оптимізації строків проекту?
5. З якою метою виконують виявлення завдань із перевищенням вартості?
6. Яким чином здійснюється підвищення рівня доступності ресурсу в проекті?
7. Яким чином можна скоротити критичний шлях проекту?
8. З якою метою проводиться зміна календарів проекту?

Лабораторна робота № 12

Контроль за ходом реалізації проекту методом освоєного обсягу

Мета роботи

Розробити опорний план проекту і провести аналіз і прогнозування ходу виконання проекту методом освоєного обсягу.

Основні теоретичні відомості

Метод освоєного обсягу заснований на визначенні відношення фактичних витрат до обсягу робіт, які повинні бути виконані на певну дату. При цьому враховується інформація по вартості, плановому і фактичному графіку робіт і дається узагальнена оцінка про стан робіт на поточний момент часу. Виявлені тенденції використовуються для прогнозу майбутньої вартості робіт при завершенні і визначення факторів, що впливають на графік виконання робіт.

Поточна дата – це дата, на яку виконується розрахунок звітів.

При аналізі освоєного обсягу використовуються три показники для визначення розбіжності в графіку робіт і вартості: планові витрати, фактичні витрати та освоєний обсяг.

Освоєний обсяг (Earned value, EV) або *базова вартість виконаних робіт* (Budgeted Cost of Work Performed, BCWP) – це планова вартість фактично виконаних робіт або кількість ресурсу, запланована на фактично виконаний обсяг робіт до поточної дати. Освоєний обсяг не залежить від фактично зроблених витрат по роботі.

Плановий обсяг (Planned value, PV) або *базова вартість запланованих робіт* (Budgeted Cost of Work Scheduled, BCWS) – це бюджетна вартість робіт, запланованих відповідно до розкладу до поточної дати, або кількість ресурсу, передбачуваного для використання до поточної дати.

Фактичні витрати (Actual cost, AC) або *фактична вартість виконаних робіт* (Actual Cost of Work Performed, ACWP) – це фактична вартість виконаних робіт на поточну дату або кількість ресурсу, фактично витраченого на виконання робіт до поточної дати.

Метод освоєного обсягу дозволяє визначити реальне відхилення проекту за витратами та за графіком виконання робіт.

Відхилення за витратами (CV, Cost Variance) (перевитрата коштів) – це величина, отримана з різниці освоєного обсягу і

фактичної вартості виконаних робіт:

$$CV = EV - AC.$$

Його значення:

- $CV > 0$ показує економію бюджету;
- $CV = 0$ показує, що робота вкладається в бюджет;
- $CV < 0$ показує перевищення бюджету.

Для роботи, що знаходиться в процесі виконання, необхідно визначити процентну оцінку завершеності (з точки зору витрат).

Відставання від графіка (SV, Schedule Variance) визначається різницею між освоєним обсягом і плановим обсягом:

$$SV = EV - PV.$$

Вимірює, наскільки в грошовім вираженні виконання проекту випереджає або відстає від плану:

- $SV > 0$ показує, що виконано більше робіт, чим заплановано;
- $SV = 0$ показує, що робота іде за графіком;
- $SV < 0$ показує, що виконано менше робіт, чим заплановано.

Відхилення за витратами та відставання від графіка розраховуються для кожної роботи та підсумовуються для оцінки всього проекту.

Індекс виконання бюджету (CPI, Cost Performance Index) показує відношення освоєного обсягу до фактичних витрат:

$$CPI = EV / AC.$$

CPI показує, наскільки добре проект укладається в бюджет:

- $CPI > 1$, вартість нижче, чим було заплановано;
- $CPI < 1$, вартість вище, чим було заплановано.
- $CPI = 0$ показує, що роботу ще не було розпочато.

Індекс виконання розкладу (SPI, Schedule Performance Index) показує відношення освоєного обсягу до планового обсягу:

$$SPI = EV / PV.$$

SPI показує, наскільки добре проект укладається в розклад:

- $SPI > 1$ показує, що проект випереджає розклад;
- $SPI < 1$ показує, що проект відстає від розкладу.
- $SPI = 0$ показує, що роботу ще не було розпочато.

Кінцева вартість проекту прогнозується на підставі інформації про витрати проекту на поточний момент часу.

Планова бюджетна вартість (BAC, Budget At Completion) – це обсяг коштів, запланованих на проект. Вона дорівнює сумі бюджетних витрат всіх запланованих робіт.

Оцінка кінцевої вартості проекту (EAC, Estimate At Completion) дорівнює сумі фактичних витрат на поточну дату та оцінки робіт, що залишилися (ETC, Estimate to completion):

$$EAC = AC + ETC.$$

Оцінка робіт, що залишилися, може бути виконана декількома способами.

- це може бути нова оцінка вартості проекту, що залишилася;
- планова вартість робіт, що залишилась, при умові, що попередні відхилення були тимчасовими і далі проект буде йти за планом:

$$ETC = BAC - EV;$$

- вартість проекту, що залишилася, скоригована з урахуванням поточної продуктивності:

- з урахуванням поточної витрати бюджету

$$ETC = (BAC - EV) / CPI;$$

- з урахуванням поточної витрати бюджету і виконання графіку робіт

$$ETC = (BAC - EV) / (CPI \times SPI).$$

На основі прогнозованої та планової вартості визначають показник **прогнозного відхилення вартості проекту** (VAC, Variance at Completion):

$$VAC = BAC - EAC.$$

Слід зазначити, що часто значимість показників самих по собі нижча, ніж їхня зміна з часом.

Розглянемо приклад розрахунків на прикладі проекту, вихідні дані до якого наведені в табл. 12.1.

Таблиця 12.1 – Вихідна інформація про проект

Робота	Попередня робота	Тривалість, дні	Вартість в день, тис. грн.	Вартість, тис. грн.
A	-	3	3	9
B	A	5	5	25
C	A	3	12	36
D	A	4	9	36
E	A	2	9	18
F	C	4	6	24
G	D, E	3	4	12
H	B, F, G	2	11	22

Побудуємо мережевий графік проекту та розрахуємо критичний шлях. На рис. 12.1 наведено діаграму передування проекту з розрахунками. За розрахунком, тривалість проекту складає 12 днів.

Планова вартість робіт проекту (BAC) складає 182 тис. грн.

Будемо вважати, що вартість робіт списується в проекті пропорційно до відсотка їх завершення.

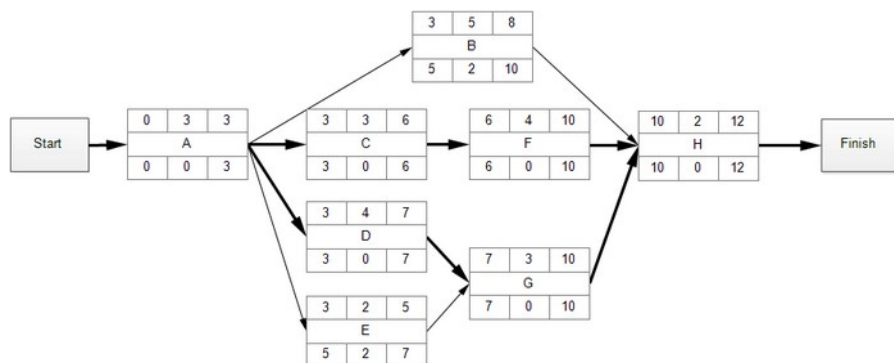


Рисунок 12.1 – Діаграма передування проекту з розрахованим критичним шляхом

Побудуємо опорний або базовий план проекту, який буде використовуватися в якості основи при розрахунках (рис. 4.2). На рисунку для кожної роботи вказано тривалість (T), строк раннього старту (ES) та пізнього фінішу (LF), часовий резерв (TF) та планова вартість (PV). Графік робіт показано у вигляді діаграми Ганта. Для кожної роботи вказано моменти списання вартості.

						Періоди часу (дні)															
Робота	T	ES	LF	TF	BA C	0	1	2	3	4	5	6	7	8	9	10	11	12			
A	3	0	3	0	9	3	3	3													
B	5	3	10	2	25					5	5	5	5	5							
C	3	3	6	0	36					12	12	12									
D	4	3	7	0	36					9	9	9	9								
E	2	3	7	2	18					9	9										
F	4	6	10	0	24								6	6	6	6					
G	3	7	10	0	12									4	4	4					
H	2	10	12	0	22												11	11			
Бюджет по днях						3	3	3	35	35	26	20	15	10	10	11	11				
Кумулятивний бюджет						3	6	9	44	79	105	125	140	150	160	171	182				

Рисунок 12.2 – Опорний план проекту

Знизу графіку Ганта для кожного періоду часу показана планова вартість робіт за вказаний період та кумулятивна планова вартість робіт. З цих даних можна побудувати графік витрати бюджету проекту (рис. 12.3).

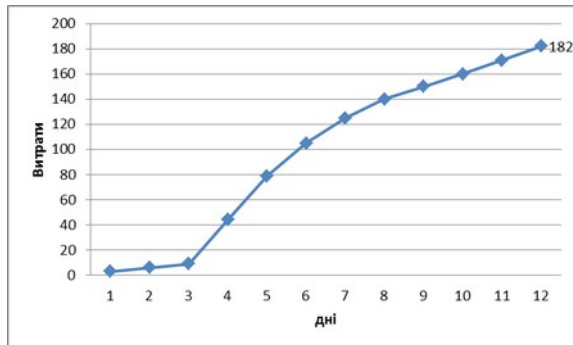


Рисунок 12.3 – Графік витрати бюджету проекту

Звіт про статус – це моментальний знімок проекту в конкретний момент часу (на день звіту). У звіті про статус використовуються параметри освоєного обсягу, фактичної вартості робіт та строків виконання роботи.

Роботи можуть перебувати в одному з трьох станів на день звіту: ще не починалися, вже закінчені та знаходяться в процесі виконання тобто частково завершені.

Нехай нам необхідно сформулювати звіт про статус проекту, параметри якого наведені в табл. 12.1, на сьомий день його виконання. На рис. 12.2 статусну дату показано зеленим кольором. Припустимо наступні параметри виконання проекту на цю дату.

Робота А вже завершена в заплановані строки за 3 дні, фактична вартість її виконання 9 тис. грн.

Робота В розпочалася на день пізніше запланованого строку, вона знаходиться в процесі виконання і виконана на 60 %. Фактична вартість її виконання 15 тис. грн.

Робота С почалася в строк, але її виконання зайняло 4 дні. Фактична вартість її виконання 48 тис. грн.

Робота D вже завершена в заплановані строки за 4 дні, фактична вартість її виконання 36 тис. грн.

Робота Е вже завершена в заплановані строки за 2 дні, фактична вартість її виконання 18 тис. грн.

Роботу F ще не розпочато, адже попередня перед нею робота С затрималась.

Робота G ще не розпочато.

Робота H ще не розпочато.

Розглянемо поточну ситуацію. Зсув роботи В на один день жодним чином не впливає на інші роботи, адже ця робота не є критичною і має часовий резерв в два дні. Однак затримка роботи С на один день зсуває час завершення проекту на один день, адже ця робота лежить на критичному шляху. Це призводить до необхідності перепланування календарного графіку наступних робіт, тобто їхнього зсуву.

Звіт про статус проекту за методом освоєного обсягу наведено в табл. 12.2.

Для того, щоб обрахувати планового обсягу та освоєного обсягу для певної роботи скористаємося наступними формулами:

$$PV = (\text{відсоток завершення за планом}) * \text{BAC}$$

$$EV = (\text{відсоток завершення фактичний}) * \text{BAC}$$

Так на сьомий день за планом передбачено наступний відсоток

виконання робіт: А – 100%, В – 80%, С – 100%, D – 100%, E – 100%, F – 25%, G – 0%, H – 0%.

Фактичне виконання робіт наступне: А – 100%, В – 60%, С – 100%, D – 100%, E – 100%, F – 0%, G – 0%, H – 0%.

Таблиця 4.2 – Звіт про статус проекту

Робота	Start	T	BAC	PV	EV	AC	SV	SPI	CV	CPI
A	0	3	9	9	9	9	0	1	0	1
B	5	5	25	20	15	15	-5	0,75	0	1
C	3	4	36	36	36	48	0	1	-12	0,75
D	3	4	36	36	36	36	0	1	0	1
E	3	2	18	18	18	18	0	1	0	1
F	7	4	24	6	0	0	-6	0	0	0
G	7	3	12	0	0	0	0	0	0	0
H	11	2	22	0	0	0	0	0	0	0
Всього			182	125	114	126	-11	0,9	-12	0,9

Оцінка вартості робіт до завершення проекту:

$$ETC = (BAC - EV) / CPI = (182 - 114) / 0,9 = 75,6 \text{ тис.грн.}$$

Оцінка кінцевої вартості проекту

$$EAC = AC + ETC = 126 + 75,6 = 201,6 \text{ тис.грн.}$$

Прогнозоване відхилення вартості проекту

$$VAC = BAC - EAC = 182 - 201,6 = -19,6 \text{ тис.грн.}$$

В нашому прикладі оцінка до завершення робіт проекту ETC розраховується з урахуванням індексу освоєння витрат. В протилежність, можна було б при прогнозуванні вважати, що поточні проблеми були обумовлені одночасними проблемами і не будуть повторені, тобто подальше виконання піде за планом.

Завдання на лабораторну роботу

1. Мережевий графік проекту взяти з роботи №7.
2. Побудувати опорний плану проекту, використовуючи дані про тривалість та вартість робіт, наведених в табл. 12.3.

Таблиця 12.3 – Вихідні дані для методу освоєного обсягу

Робота	Тривалість, дні	Вартість в день, тис. грн
A	6	12
B	5	10

Робота	Тривалість, дні	Вартість в день, тис. грн
C	4	10
D	6	4
E	3	11
F	8	8
G	5	10
H	3	12
I	4	4
J	5	7
K	6	11
L	2	6
M	3	20
N	4	15
O	4	6

3. Побудувавши опорний план, за узгодженням з викладачем назначити параметри виконання проекту на випадкову дату.

4. Створити звіт за проектом методом освоєного обсягу.

5. Проаналізувати стан проекту, оцінити прогнозовану вартість проекту та зробити висновки.

Контрольні питання

1. Якими параметрами характеризується фактичне виконання робіт?

2. Що приводить до відхилення фактичних працезатрат від планових?

3. Що таке базова вартість запланованих робіт?

4. Що таке базова вартість виконаних робіт?

5. Що таке фактична вартість виконаних робіт?

6. Що таке індекс виконання бюджету або індекс відхилення вартості?

7. Як розрахувати індекс виконання розкладу або індекс відхилення від календарного плану?

8. Як розрахувати відхилення від графіка?

9. Як розрахувати відхилення по витратах?

10. Яку інформацію про хід проекту одержує менеджер

проекту, використовуючи метод освоєного обсягу?

11. Що таке планова бюджетна вартість, оцінка кінцевої вартості проекту, відхилення по завершенню та оцінка до завершення?

ЛИТЕРАТУРА

1. Веб-сайт системы Slack [Электронный ресурс]. – Режимы доступа: <https://slack.com/>.
2. Веб-сайт системы ToggI [Электронный ресурс]. – Режимы доступа: <https://toggl.com/>
3. Веб-сайт системы TopTracker [Электронный ресурс]. – Режимы доступа: <https://www.toptal.com/tracker/>
4. Веб-сайт системы Redmine 3.3 [Электронный ресурс]. – Режимы доступа: <http://www.redmine.org/>.
5. Веб-сайт системы Trello [Электронный ресурс]. – Режимы доступа: <https://trello.com/>.
6. Веб-сайт ProjectLibre 1.6 [Электронный ресурс]. – Режимы доступа: <http://www.projectlibre.org>.
7. Мазур И.И. Управление проектами: Справочное пособие / И.И. Мазур, В.Д. Шапиро. – М.: Высшая школа, 2001. – 875 с.
8. Куперштейн В.: Microsoft Project 2013 в управлении проектами / В. Куперштейн. – СПб.: BHV-СПб, 2014. – 432 с.
9. Уокер Ройс. Управление проектами по созданию программного обеспечения / Ройс Уокер. – Лори, 2002. – 426 с.
10. Арчибальд Рассел Д. Управление высокотехнологичными программами и проектами / Рассел Д. Арчибальд. – ДМК Пресс, 2010. – 464 с.
11. Грей Клиффорд Ф. Управление проектами: Практическое руководство / Клиффорд Ф. Грей, Эрик У. Ларсон. – М.: Дело и Сервис, 2003. – 528 с.
12. Кантор Марри. Управление программными проектами. Практическое руководство по разработке успешного программного обеспечения / Марри Кантор. – М.: Вильямс, 2002. – 176 с.
13. Баркалов С.А. Математические основы управления проектами: учебн. пособие / С.А. Баркалов В.И. Воропаев, Г.И. Секлетова и др. – М.: Высш. шк., 2005. – 423 с.

Додаток А
Варіанти індивідуальних завдань

А.1. Передування робіт проекту

Робота	Попер. робота	Робота	Попер. робота
Варіант 1			
A	-	H	C, E, F, G
B	A	I	H
C	B	J	H
D	B	K	I
E	D	L	J
F	D	M	K, L
G	D		
Варіант 2			
A	—	H	A, B
B	—	I	H
C	B	J	I
D	B	K	G
E	C	L	G
F	D	N	J, K, L
G	E, F	O	N
Варіант 3			
A	-	H	F
B	A	I	F
C	A	J	G, H, I
D	A	K	H, I
E	B	L	J
F	E, C, D	M	K
G	F	N	L, M
Варіант 4			
A	—	H	G
B	—	I	H, F
C	A	J	I
D	A, B	K	I
E	B	M	L, J
F	D, E	N	K, M
G	C	L	F
Варіант 5			

Робота	Попер. робота	Робота	Попер. робота
A	—	H	F
B	—	J	I, H
C	A	L	J
D	A, B	K	G
E	A, B	I	C
F	C, D	N	L, K
G	E	O	N
Варіант 6			
A	—	H	G
B	A	I	H, F
C	A	J	I
D	A	K	J
E	B, C	L	J
F	C, D	N	J
G	E	O	K, L, N
Варіант 7			
A	—	H	F
B	—	I	F
C	—	J	H, I
D	A, B	K	G
E	C	L	G
F	D	N	J, K, L
G	E	O	N
Варіант 8			
A	—	H	F, G
B	—	I	H
C	A, B	J	E
D	C	K	J, I
E	D	L	H
F	D	N	L
G	D	O	K, N
Варіант 9			
A	—	H	D
B	A	I	H, E, G
C	A	J	I
D	B	K	J
E	B	L	K
F	C	N	M, K

Робота	Попер. робота	Робота	Попер. робота
G	F	M	I
Варіант 10			
A	—	H	E
B	—	I	F, G, H
C	A	J	I
D	B	K	J
E	C, D	L	J
F	E	N	K, L
G	E	O	N

А.2. Тривалості робіт для методу PERT

Робота	Оптимістична оцінка тривалості, О	Найбільш імовірна тривалість, М	Песимістична оцінка, Р
Варіанти завдань 1-5 (№ - номер варіанту)			
A	1	4	7
B	2	4	8
C	1	4	7
D	№	9	18
E	4	8	12
F	2	4	6
G	4	6	8
H	№	6	12
I	№	9	17
J	3	4	5
K	4	6	8
L	3	4	5
M	2	4	6
N	№	11	16
O	№	7	15
Варіанти завдань 6-10 (№ - номер варіанту)			
A	2	4	8
B	2	4	6
C	1	4	№
D	2	4	№
E	4	6	№
F	3	4	5
G	6	6	№
H	4	8	12
I	6	11	16
J	7	9	17
K	3	4	5
L	1	4	7
M	4	8	12
N	5	№	14
O	7	9	11

Додаток Б

Проектне завдання

1. **Назва проекту:** Створення офіційного Web-Сайту ТОВ «Компанія»

2. **Головна мета проекту** – підвищення ефективності роботи компанії.

Мети проекту: підвищення рівня поінформованості потенційних клієнтів про переваги пропонованих послуг; забезпечення зв'язку клієнтів з компанією; залучення уваги клієнтів до найцікавіших пропозицій компанії.

3. **Планові строки**

початок робіт: 1 вересня 2019 р.

закінчення робіт: 26 грудня 2019 р.

(17 тижнів)

4. **Замовник:** ТОВ «Компанія»

5. **Виконавець:** ТОВ «Студент»

6. **Склад робіт**

Найменування етапів і робіт, їх орієнтовну тривалість представлено в табл. Б.1 (на підставі наявного досвіду виконавця).

Таблиця Б.1 – Склад і тривалість робіт проекту

№	Назва роботи	Тривалість, день
1	Проект	
2	Аналіз вимог	
3	Аналіз предметної області, інтерв'ювання клієнта	6
4	Розробка й затвердження концепту сайту	2
5	Планування проекту	3
6	Складання ТЗ	5
7	Затвердження ТЗ	2
8	Завершення аналізу вимог	0
9	Проектування	
10	Розробка дизайну за затвердженим концептом	7
11	Проектування архітектури сайту	3
12	Проектування БД	3
13	Завершення проектування	0

Продовження таблиці Б.1

№	Назва роботи	Тривалість, день
14	Програмування й верстка	
15	Верстка шаблонів сторінок	32
16	Розробка бази даних	5
17	Програмування	40
18	Заповнення сторінок і тестування на реальних даних	14
19	Доробка коду, виправлення помилок	7
20	Завершення програмування й верстки	0
21	Запуск і супровід	
22	Створення пакета передачі	2
23	Підготовка платформи	1
24	Заливання контенту	1
25	Перевірка якості	3
26	Здача сайту замовникові	1
27	Закриття проекту	1

Розглянемо короткий зміст етапів робіт у їхній технологічній послідовності для визначення зв'язків між завданнями.

Аналіз вимог. На даному етапі спочатку здійснюється аналіз предметної області: виявлення потреб майбутніх користувачів; визначення технічних вимог, що включають вимоги до структури сайту, змісту розділів і підрозділів; визначення вимог до дизайну; визначення вимог до структури й функцій програмних модулів; визначення вимог до архітектури, системи керування контентом сайту тощо.

За результатами аналізу складається концепт сайту, який обговорюється із Замовником і командою розробників проекту.

Після затвердження концепту розробники приступають до детального аналізу й формування вимог до розроблюваної системи.

Паралельно з розробкою ТЗ здійснюється планування проекту, що передбачає визначення складу й строків здійснення робіт, ресурсів і бюджету проекту.

Отримані результати приводяться в технічному завданні, яке затверджується Замовником.

Проектування. Дизайн сайту повинен відповідати встановленим цілям і бути функціональним.

Для створення ескізів дизайнер аналізує концепт і технічне завдання.

На даному етапі також проводиться проектування архітектури сайту й проектування бази даних згідно ТЗ.

Програмування й верстка. Верстка шаблонів сторінок здійснюється відповідно до керівництва зі стилю оформлення.

Паралельно з версткою починається розробка бази даних. Після розробки БД виконується реалізація спроектованої системи.

Заповнення сторінок контентом і тестування на реальних даних повинні початися за 10 днів до завершення програмування. Роботи з верстки до цього моменту вже повинні бути закінчені.

Доробка коду й виправлення помилок здійснюється відразу по завершенню програмування й наповнення.

Запуск і супровід. До передачі Замовникові й запуску сайту необхідно сформувати комплект документації («пакет передачі»), що включає всі вихідні файли, зображення, шаблони, інструкції та ін., необхідні особі, яка буде супроводжувати сайт після його впровадження.

Одночасно з формуванням пакета передачі проводиться підготовка платформи: розміщення сайту на хостинговому сервері, налаштування й підтримка DNS.

Потім здійснюється перенос сайту (заливання контенту) на сервер Замовника.

Після успішного розміщення на сервері сайт тестується ще раз для усунення помилок, пов'язаних з особливостями розміщення.

Після завершення всіх робіт у встановлений термін з необхідним рівнем якості Замовник підписує акт про впровадження, здійснює взаєморозрахунок з Виконавцем згідно з договором. Керівник здійснює закриття проекту.

7. Планові строки етапів робіт

Планові строки етапів робіт установлювалися на основі аналізу обсягу робіт і наявного досвіду створення аналогічних проектів Виконавцем і наведено в табл. Б.2.

Таблиця Б.2 – Строки виконання етапів робіт проекту

№	Найменування етапу робіт	Строки виконання	Тривалість
1	Аналіз вимог	1.09.14-21.09.14	15
2	Проектування	22.09.14- 30.09.14	7
3	Програмування й верстка	1.10.14- 17.12.14	56
4	Запуск і супровід	18.12.14-25.12.14	6
5	Закриття проекту	26.12.14	1
	Усього		85

8. Трудові ресурси проекту

Список трудових ресурсів, необхідних для виконання проекту, і їх вартість наведені в табл. Б.3. Матеріальні ресурси повинні бути сплановані студентами самостійно.

Таблиця Б.3 – Список трудових ресурсів

№	Назва ресурсу (роль)	Стандартна ставка грн/година
1	Керівник проекту	40
2	Аналітик	37
3	Дизайнер	30
4	Верстальщик	27
5	Програміст Senior	37
6	Програміст Middle	35
7	Програміст Junior	30
8	Тестувальник	25
9	Системний адміністратор	37

8. Орієнтовна вартість етапів робіт

Вартість етапів проекту наведена в табл. Б.4.

Таблиця Б.4 – Вартість етапів робіт

	Найменування етапу робіт	Вартість, тис. грн
1	Аналіз вимог	6
2	Проектування	4
3	Програмування й верстка	37
4	Запуск і супровід	3

9. Бюджет проекту: 50 тис. грн.

10. Матриця завдань і відповідальності

Матриця завдань і відповідальності встановлює зв'язок між роботами, визначеними в структурі проекту (табл. Б.1), і співробітниками (табл. Б.3), відображаючи призначення ресурсів на виконання завдань (табл. Б.5).

Кожний завдання можуть бути призначені декілька співробітників. Ступінь відповідальності співробітників позначена як «О» — основна відповідальність, «В » — допоміжна відповідальність.

Таблиця Б.5 – Матриця завдань і відповідальності

Номер роботи	Номер ресурсу								
	1	2	3	4	5	6	7	8	9
3	В	О							
4		В	О						
5	О								
6	В	О							
7	О	В							
10			О						
11		В			О				
12		В				О			
15				О					
16						О			
17					О	В	О		
18								О	
19							О	В	
22	В				О				
23									О
24					В				О
25						О	В		
26	О								
27	О								