

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Управління проєктами Лабораторний практикум

Навчальний посібник

Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського
як навчальний посібник для здобувачів ступеня бакалавра
за спеціальністю 126 Інформаційні системи та технології

Укладач: Б. Я. Корнієнко

Електронне мережне навчальне видання

Київ
КПІ ім. Ігоря Сікорського
2023

Рецензент *Ладієва Л. Р.*, канд. техн. наук, доцент,
доцент кафедри технічних та програмних засобів автоматизації,
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Відповідальний редактор *Ролік О.І.*, докт. техн. наук, професор

*Гриф надано Методичною радою КПІ ім. Ігоря Сікорського
(протокол № 7 від 27 04 2023 р.)
за поданням Вченої ради Факультету інформатики та обчислювальної техніки
(протокол № 11 від 24 04 2023 р.)*

Навчальний посібник розрахований на одержання студентами практичних навичок з управління проектами і складається з 5 лабораторних робіт, що охоплюють питання автоматизації та інформатизації прикладних процесів та управління проектами зі створення програмних продуктів відповідно до існуючих в ІТ-індустрії технологічних процесів. В навчальному посібнику приділено увагу основним підходами до стратегічного планування і стратегічного аналізу проектів, управління інтеграцією проектів, управління часом проектів та управління ризиками проектів.

Містить теоретичні положення, порядок виконання лабораторних робіт, завдання та список літератури з дисципліни «Управління проектами».

Для студентів всіх форм навчання, які навчаються за спеціальністю 126 «Інформаційні системи та технології» факультету інформатики та обчислювальної техніки КПІ ім. Ігоря Сікорського.

Реєстр. № НП 22/23-658. Обсяг 2,6 авт. арк.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
проспект Перемоги, 37, м. Київ, 03056
<https://kpi.ua>

Свідоцтво про внесення до Державного реєстру видавців, виготовлювачів і розповсюджувачів видавничої продукції ДК № 5354 від 25.05.2017 р.

© КПІ ім. Ігоря Сікорського, 2023

ЗМІСТ

ВСТУП.....	4
Лабораторна робота 1. Методології управління ІТ-проєктами	5
Лабораторна робота 2. Аналіз ризиків	23
Лабораторна робота 3. Порівняльний аналіз інформаційних систем	31
Лабораторна робота 4. Планування проєкту	40
Лабораторна робота 5. Виявлення потреб проєкту	54
ДОДАТОК	60
СПИСОК ЛІТЕРАТУРИ.....	63

ВСТУП

Використання принципів проектного управління дозволяє ефективно розв'язувати задачі розвитку організації та підвищує надійність успішного досягнення поставлених цілей. Разом з тим, успіх проектного управління залежить від вмотивованості співробітників організації, від їх розуміння та підтримки філософії проектного підходу, а також від ефективності побудови внутрішньокорпоративних комунікацій.

Управління ІТ-проектами є самостійною синтетичною дисципліною в межах якої вивчаються функціональний, динамічний та предметний аспекти управлінської діяльності у сфері розробки інформаційних технологій.

Метою викладання навчальної дисципліни «Управління проектами» є теоретична та практична підготовка здобувачів освітнього ступеня бакалавра у напрямку виконання проектних робіт щодо автоматизації та інформатизації прикладних процесів та управління проектами зі створення програмних продуктів відповідно до існуючих в ІТ-індустрії технологічних процесів.

Основне завдання дисципліни «Управління проектами» є забезпечити розуміння і засвоєння здобувачами освітнього ступеня бакалавра стандартів, процесів і процедур управління ІТ-проектами, принципів командної роботи та отримання практичних навичок використання програмних систем проектного управління.

Метою лабораторних робіт є набуття здобувачами освітнього ступеня бакалавра практичних навичок стратегічного планування і стратегічного аналізу ІТ-проектів, управління інтеграцією ІТ-проектів, управління змістовим наповненням ІТ-проектів, управління часом ІТ-проектів та управління ризиками ІТ-проектів.

Лабораторна робота 1

Методології управління ІТ-проектами

Мета роботи: знайомство з методологіями управління ІТ-проектами.

Порядок виконання роботи

1. Вивчити теоретичні відомості.
2. Разом з викладачем вибрати варіант завдання.
3. Виконати завдання до лабораторної роботи згідно свого варіанту.
4. Скласти та оформити звіт.

Теоретичні положення

В даний час управління проектами має свою методологію і ґрунтується на певних стандартах. В основі таких методів лежать методики мережевого планування, які з'явилися наприкінці 1950-х рр. США. У цьому методики управління проектами широко поширилися у країнах із ринковою економікою, а й у країнах із т. зв. "планової" економікою. Вони почали використовуватися в будівництві, що послужило основою їхнього поширення в інших галузях та появі методів проектного управління.

Прикладом тут може бути об'єднання зусиль фірми «Дюпон» та фірми «Ремінгтон Ренд» для складання плану графіка комплексних робіт з модернізації заводів «Дюпон» за допомогою обчислювальної машини Univac у 1956 р. Результатом стало створення раціонального та простого методу опису проекту на ЕОМ – методу критичного шляху (CPM – Critical Path Method).

Практично паралельно та незалежно було створено метод аналізу та оцінки програм PERT (Program Evaluation and Review Technique) у військово-морських сил США. Цей метод розроблявся корпорацією «Локхід» консалтинговою фірмою Буз, Аллен енд Гамільтон для ракетної системи Поларіс. Проект включав 3800 основних підрядників та складався з 60 000 операцій. Завдяки використанню даного методу, проект став успішним і завершився на два роки раніше терміну. Після наочного успіху метод PERT став використовуватись у збройних силах США для планування проектів.

Завдання: знайдіть приклади інших великих успішних проектів минулого століття, які використовують для управління.

З одного боку, застосування інформаційних технологій для планування та управління проектами давало на той час значну перевагу, а з іншого боку, перші комп'ютери були дорогими і залишалися доступними лише великим компаніям. Усе змінилося з появою персональних комп'ютерів, тепер комп'ютер став робочим інструментом керівників ширшого кола.

Система управління проектами постійно розвивалася і стала самостійною галуззю професійної діяльності. У результаті було створено уніфіковані методології, інструментарії, механізми, стандарти, доступні й у проектів у ІТ-сфері. Наприклад, на сьогоднішній день існує єдина Міжнародна асоціація управління проектами – IPMA (International Project Management Association) із центром у м. Цюріх (Швейцарія).

З найпоширеніших можна відзначити процесну модель, що використовується в документах методологічних засад управління проектами - Project Management Body of Knowledge (PMBOK) Американського інституту управління проектами (PMI). Цей документ визнається міжнародним стандартом де-факто. Крім того, стандарт ISO 10006:1997 надав ряду найважливіших положень PMBOK статусу стандарту де-юре.

В 2014 р. вийшло п'яте видання PMBOK, що містить вказівки на 589 сторінок. Усі положення представлені на сайті <http://www.pmi.org/default.aspx>.

Дані Міжнародної асоціації управління проектами (IPMA) свідчать, що використання сучасних методологій управління проектами заощаджує близько 20–30% часу та близько 15–20% коштів на здійснення проектів.

Усі методології (ще їх називають моделями, методиками) розробки програмного забезпечення класифікують за «вагою», тобто за кількістю формалізованих процесів та детальністю їхньої регламентації. Отже, що більше процесів документовано, що більш детально описана методологія, то більше вписуватиметься її «вага».

1. Важковагові методології:

- Capability Maturity Model for Software (SW-CMM) визначає п'ять рівнів зрілості проекту;
- Rational Unified Process (RUP) – ітеративна модель розробки;
- Microsoft Solutions Framework (MSF) – база знань компанії Microsoft з розробки програм;
- Personal Software Process – модель визначає вимоги до компетенцій розробника. Team Software Process – модель орієнтує на самоврядні команди від 3 до 20 розробників.

2. Легкі або Agile- методології:

- eXtreme Programming або XP – екстремальне програмування, що пропонує 12 інженерних практик;
- Crystal Clear – сімейство методологій, що визначають необхідний ступінь формалізації процесу розробки залежно від кількості учасників та критичності завдань;
- Feature Driven Development (FDD) - функціонально-орієнтована технологія;
- OpenUP – ітеративно-інкрементальний метод розробки програм, що позиціонується як легкий та гнучкий варіант великовагової методології RUP;
- Scrum – управління розробкою інформаційних систем із високим ступенем невизначеності;
- Kanban - методологія "ощадливого виробництва".

Результати дослідження Agile Survey щодо популярності гнучких методологій представлені на рис. 1 [11].

Легкі методології Agile з'явилися порівняно недавно. У лютому 2001 р. 17 фахівців (консультантів та практиків) провели семінар, на якому сформулювали основні засади гнучкої розробки програмного забезпечення –

Agile Manifesto – маніфест гнучкої розробки. Він перекладений багатьма мовами світу і доступний на сайті <http://agilemanifesto.org/iso/>.

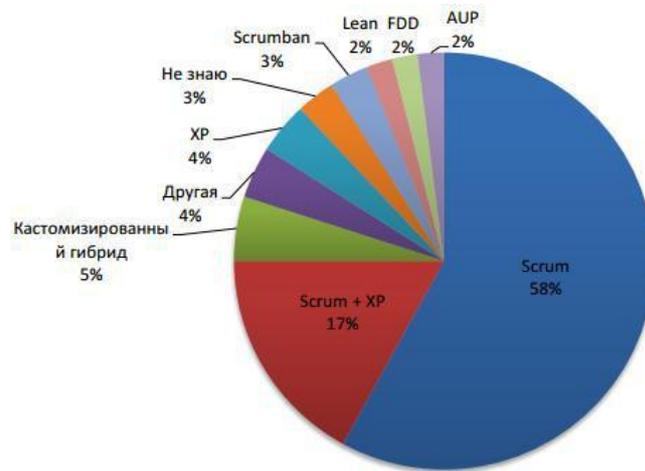


Рис. 1. Популярність гнучких методологій

Ідея всіх гнучких (легковагових) методологій полягає в тому, що процес, що застосовується в розробці, повинен бути адаптивним, орієнтованим на людей та їх взаємодію. По суті, це швидше набір практик, ніж методологія.

На цьому ж семінарі було запропоновано нову назву таких методологій – гнучка розробка Agile Software Development.

При виборі моделі управління проектом можна орієнтуватися на таку таблицю (Таблиця 1).

А. Коуберн, один із авторів «Маніфесту», провів аналіз ІТ-проектів за останні 20 років [4], виконаних на підставі різних моделей управління: від полегшених, гнучких до важких (СММ-5). Даний аналіз показав відсутність кореляції між успіхом чи провалом проектів та обраними моделями розробки, що застосовуються у проектах.

А. Коуберн зробив також висновок про те, що:

1. У кожного проекту має бути своя модель процесу розробки.
2. Кожна модель має свій час.

Таблиця 1 - Вибір методології управління проектом

Вага моделі	Переваги	Недоліки
Важковагові	<p>Процеси розраховані на середню кваліфікацію виконавців; Велика спеціалізація виконавців;</p> <p>Низькі вимоги до стабільності команди;</p> <p>Відсутні обмеження по обсягу та складності виконуваних проектів.</p>	<p>Вимагають суттєвої управлінської надбудови;</p> <p>Тривалі стадії аналізу та проектування;</p> <p>Формалізовані комунікації.</p>
Легкі (Гнучкі)	<p>Зниження непродуктивних витрат, пов'язаних з управлінням проектом, ризиками, змінами, конфігураціями; Спрощення стадій аналізу та проектування, основний акцент на розробку функціональності, поєднання ролей; Неформальні комунікації.</p>	<p>Ефективність сильно залежить від індивідуальних здібностей, вимагають більш кваліфікованої, універсальної і стабільної команди;</p> <p>Об'єм і складність виконуваних проектів обмежені.</p>

Для розробників це означає, що немає єдиного правильного процесу розробки.

Уніфікований процес Rational (Rational Unified Process, RUP) [4] є досить складною, детально пропрацьованою ітеративною моделлю життєвого циклу ПЗ. RUP є програмним продуктом в якості доповнення до мови моделювання UML, розроблений компанією Rational Software, яка нині входить до складу ІВМ.

Основні принципи RUP

Історично RUP є розвитком моделі процесу розробки, прийнятої в компанії Ericsson в 70-х-80-х роках ХХ століття. Ця модель була створена Джекобсоном (Ivar Jacobson), який у 1987 заснував власну компанію Objectory АВ саме для розвитку технологічного процесу розробки ПЗ як окремого продукту, який можна було б переносити в інші організації. Після вливання Objectory в Rational в 1995 розробки Джекобсона були інтегровані з роботами Ройса (Walker Royce, сина автора «класичної» каскадної моделі), Крухтена (Philippe Kruchten) і Буча (Grady Booch), а також з тим, що розвивалося паралельно універсальною мовою моделювання (Unified Modeling Language, UML) [17].

Основними принципами RUP є:

1. Ітераційний і інкрементний (нарощуваний) підхід до створення ПЗ.
2. Планування і управління проектом на основі функціональних вимог до системи – варіантів використання.
3. Побудова системи на базі архітектури ПЗ.

Перший принцип є визначальний. Відповідно до нього розробка системи виконується у вигляді декількох короткострокових міні-проектів фіксованої тривалості (від 2 до 6 тижнів), що називаються ітераціями. Кожна ітерація включає свої власні етапи аналізу вимог, проектування, реалізації, тестування, інтеграції і завершується створенням працюючої системи.

Фази проекту

З точки зору етапів (фаз) проектування, то вони, загалом, аналогічні «водоспаду». Але RUP виділяє в життєвому циклі 4 основних фази, у рамках кожної з яких можливе проведення декількох ітерацій.

1. Фаза початку проекту (Inception). Основна мета цієї фази – досягти компромісу між усіма зацікавленими особами відносно завдань проекту і ресурсів, що виділяються на нього. На цій фазі визначаються основні цілі проекту, керівник проекту і бюджет проекту, основні засоби його виконання – технології, інструменти, ключові виконавці, а також відбувається, можливо,

апробація вибраних технологій з метою підтвердження можливості досягти цілей з їх допомогою. На цю фазу може йти близько 10% часу і 5% трудомісткості одного циклу.

Результатами початкової стадії є:

- загальний опис системи: основні вимоги до проекту, його характеристики і обмеження;

- початковий бізнес-план;

- план проекту, що відбиває стадії і ітерації;

- один або декілька прототипів.

2. Фаза проектування (Elaboration). Основна мета цієї фази – на базі основних, найбільш суттєвих вимог розробити стабільну базову архітектуру продукту, яка дозволяє вирішувати поставлені перед системою завдання і надалі використовується як основа розробки системи. На цю фазу може йти близько 30% часу і 20% трудомісткості одного циклу.

3. Фаза побудови (Construction). Основна мета цієї фази – детальне прояснення вимог і розробка системи, що задовольняє їм, на основі спроектованої раніше архітектури. В результаті повинна вийти система, що реалізовує усі виділені варіанти використання. На цю фазу йде близько 50% часу і 65% трудомісткості одного циклу.

Результатами стадії розробки є:

- модель варіантів використання (завершена принаймні на 80%), що визначає функціональні вимоги до системи;

- перелік додаткових вимог, включаючи вимоги нефункціонального характеру;

- опис базової архітектури майбутньої системи, працюючий прототип; □ план розробки усього проекту, що відбиває ітерації і критерії оцінки для кожної ітерації.

4. Фаза впровадження (Transition). Мета цієї фази – зробити систему повністю доступною кінцевим користувачам. На цій стадії відбувається розгортання системи в її робочому середовищі, бета-тестування,

підгонка дрібних деталей під потреби користувачів. На цю фазу може йти близько 10% часу і 10% трудомісткості одного циклу.

Ключові ідеї RUP

Робочі продукти (артефакти), що виробляються в ході проекту, можуть бути представлені у вигляді баз даних і таблиць з інформацією різного типу, різних видів документів, початкового коду і об'єктних модулів, а також моделей, що складаються з окремих елементів.

Найбільш важливі з точки зору RUP артефакти проекту – це моделі, що описують різні аспекти майбутньої системи. Більшість моделей є наборами діаграм UML.

Перерахуємо основну техніку, використовувану в RUP:

1. Вироблення концепції проекту на його початку для чіткої постановки завдань.
2. Управління за планом.
3. Зниження ризиків і відстежування їх наслідків, як можна більше ранній початок робіт по подоланню ризиків.
4. Ретельне економічне обґрунтування усіх дій – робиться тільки те, що треба замовникові.
5. Як можна більше раннє формування базової архітектури.
6. Використання компонентної архітектури.
7. Прототипирование, інкрементна розробка і тестування.
8. Регулярні оцінки поточного стану.
9. Управління змінами, постійний відробіток змін ззовні проекту.
10. Націленість на створення продукту, працездатного в реальному оточенні.
11. Націленість на якість.
12. Адаптація процесу під потреби проекту.

Як видно, фази в цілому відповідають моделі водоспаду, з тим виключенням, що фаза супроводу не розглядається як окрема. В цілому, при розробці проекту RUP базується на чотирьох ключових ідеях.

1. Ключовою ідеєю процесу є його ітеративність – кожна фаза, починаючи з розвитку, включає декілька ітерацій, на кожній з яких виконується свій шматочок аналізу, проектування, реалізації і тестування готового продукту (вірніше за його частину).

2. Увесь хід робіт спрямовується підсумковими цілями проекту (ітераціями), вираженими у вигляді **прецедентів використання** або **варіантів використання (use cases)** – сценаріїв взаємодії результуючої програмної системи з користувачами або іншими системами. Вимоги аналізуються також як і проект на кожній ітерації і в цілому дуже ретельно, але не до кінця – діє правило 70-80% – розробник повинен уявляти собі вимоги саме настільки, перш ніж починати кодування.

3. На етапі розвитку приймаються ключові рішення, що стосуються **архітектури** системи в цілому, її властивостей, функцій, використовуваних технологій і так далі. У кінці цієї фази реалізується 10-30% системи, але усі основні рішення вже прийняті (до 70-80%) і на етапі конструювання у рамках цих рішень система доводиться до кінця. Архітектура встановлює набір компонентів, з яких буде побудовано ПО, відповідальність кожного з компонентів (тобто вирішувані ним підзадачі у рамках загальних завдань системи), чітко визначає інтерфейси, через які вони можуть взаємодіяти, а також способи взаємодії компонентів один з одним.

4. Основою процесу розробки є **плановані і керовані ітерації**, об'єм яких визначається на основі архітектури (функціональність, що реалізовується у рамках ітерації, і набір компонентів).

RUP як продукт входить до складу інтегрованого комплексу інструментальних засобів **Rational Suite**, який існує в наступних варіантах:

- Rational Suite AnalystStudio – призначений для визначення і управління повним набором вимог до системи, що розробляється;

- Rational Suite DevelopmentStudio – призначений для проектування і реалізації ПЗ;

- Rational Suite TestStudio – є набір продуктів, призначених для автоматичного тестування додатків;

- Rational Suite Enterprise – забезпечує підтримку повного життєвого циклу

ПЗ і призначений як для менеджерів проекту, так і окремих розробників, що виконують декілька функціональних ролей в команді розробників.

До складу Rational Suite, окрім самої технології RUP як продукту, входять наступні компоненти:

- Rational Rose – засіб візуального моделювання (аналізу і проектування), що використовує мову UML;

- Rational XDE – засіб аналізу і проектування, інтегрований з платформами MS Visual Studio .NET і IBM WebSphere Studio Application Developer;

- Rational Requisite Pro – засіб управління вимогами, призначене для організації спільної роботи групи розробників;

- Rational Rapid Developer – засіб швидкої розробки додатків на платформі Java 2 Enterprise Edition;

- Rational SoDA – засіб автоматичної генерації проектної документації;

- Rational Quantify – засіб кількісного визначення вузьких місць, що впливають на загальну ефективність роботи програми;

- Rational TestManager – засіб планування функціонального і навантаження тестування;

- Rational TestFactory – засіб тестування надійності;

- Rational Quality Architect – засіб генерації коду для тестування.

Засоби автоматичної генерації коду, використовуючи інформацію, що міститься в діаграмах класів і компонентів, формують файли описів класів. Створюваний таким чином скелет програми може бути уточнений шляхом

прямого програмування на відповідній мові (основні мови, підтримувані Rational Rose, – C++ і Java).

Уніфікований процес RUP хоча і є досить складною ітеративною моделлю розробки життєвого циклу ПЗ, все ж може з успіхом застосовуватися, якщо на деяких етапах (фазах) допускати деякі «вільності».

Серед проблем власне кодування, слід виділити небажання деяких розробників (а також деяких замовників) використати компоненти сторонніх виробників. Щоб не винаходити велосипед, при написанні коду також можуть використовуватися готові програмні конструкції (готові рішення), типові рішення, шаблони, так звані «патерни» (design patterns – зразки проектування або просто patterns).

Проте які б поліпшення ми не вносили, техніку RUP властиві серйозні обмеження і недоліки:

1. Він уніфікований, тобто підходить для різномірних процесів, проектів, розробок, що робить його трохи заплутаним і неконкретним (мало конкретних чітких рекомендацій).
2. Він важкуватий для невеликих проектів і колективів розробників, особливо коли бюджет і терміни проектів невеликі.

Він вимагає глибокого осмислення вимог, як і традиційний водоспад (хоча і залишає на потім 30%). У реальних ситуаціях і 70% відразу отримати важко!

Останнім часом все більшу популярність стали набирати так звані «гнучкі» («живі») методи розробки ПЗ (Agile Software Development). Серед них найпоширенішим являється *Екстремальне програмування (eXtreme Programming, XP)* – полегшений (рухливий) процес (чи методологія), головний автор якого – Кент Бек (1999) [5].

XP-процес орієнтований на групи малого і середнього розміру, що будують програмне забезпечення в умовах невизначених або швидко таких, що змінюються вимог.

Основні принципи «живої» розробки ПЗ зафіксовані в маніфесті «живої» розробки, що з'явився в 2000 році:

1. Люди, що беруть участь в проекті, і їх спілкування важливіші, ніж процеси і інструменти.
2. Працююча програма важливіша, ніж вичерпна документація.
3. Співпраця із замовником важливіша, ніж обговорення деталей контракту.
4. Відробіток змін важливіший, ніж наслідування планів.

«Живі» методи з'явилися як протест проти надмірної бюрократизації розробки ПЗ, великої кількості побічних що не є необхідними для отримання кінцевого результату документів, які доводиться оформляти при проведенні проекту відповідно до більшості «важких» процесів. Велика частина таких робіт і документів не має прямого відношення до розробки ПЗ і забезпечення його якості, а призначена для дотримання формальних пунктів контрактів на розробку, отримання і підтвердження сертифікатів на відповідність різним стандартам.

Основна ідея XP-процесу – усунути високу вартість зміни, характерну для додатків з використанням об'єктів, **патернів** (рішення типових проблем в певному контексті або готові програмні конструкції) і реляційних баз даних. Тому XP-процес має бути високо динамічним процесом. XP-група має справу зі змінами вимог на всьому протязі ітераційного циклу розробки, причому цикл складається з дуже коротких ітерацій.

За твердженням авторів XP, ця методика є не стільки наслідуванням якихось загальних схем дій, скільки застосування комбінації відповідної техніки (практик). Кожна техніка важлива, і без її використання розробка вважається такою, що йде не по XP.

1. Гра в планування (Planning game) – швидке визначення зони дії наступної реалізації шляхом об'єднання ділових пріоритетів і технічних оцінок. Замовник формує зону дії, пріоритетність і терміни з точки зору бізнесу, а розробники оцінюють і простежують просування (прогрес).

2. Часта зміна версій (Small releases) – швидкий запуск у виробництво простої системи, тобто найперша працююча версія повинна з'явитися якнайшвидше, і тут же повинна почати використовуватися. Для реалізації нових версій вводяться ще жорсткіші обмеження на тривалість однієї ітерації.

3. Метафора (Metaphor) – уся розробка проводиться на основі простої, загальнодоступної історії про те, як працює уся система. Метафора в досить простому і зрозумілому команді виді повинна описувати основний механізм роботи системи. Це поняття нагадує архітектуру, але повинне набагато простіше, усього у вигляді однієї-двох фраз описувати основну суть прийнятих технічних рішень.

4. Просте проектування (Simple design) – проектування виконується настільки просто, наскільки це можливо в даний момент. Не потрібно додавати функції заздалегідь – тільки після явного прохання про це. Уся зайва складність віддаляється, як тільки виявляється.

5. Тестування (Testing) – безперервне написання тестів для модулів, які повинні виконуватися бездоганно. Розробники спочатку пишуть тести, потім намагаються реалізувати свої модулі так, щоб тести спрацьовували.

6. Реорганізація (рефакторинг, Refactoring) – наведення ладу в коді, переробка окремих файлів, видалення максимальної кількості незрозумілих фрагментів коду, об'єднання класів на основі їх схожої функціональності, корекція коментарів, осмислене перейменування об'єктів. Система реструктурується у зв'язку з додаванням нової функціональності, але її поведінка не змінюється; мета – усунути дублювання, спростити систему.

7. Парне програмування (Pair programming) – увесь код пишеться двома програмістами, працюючими на одному комп'ютері. Об'єднання в пари довільно і міняється від завдання до завдання. Той, в чиїх руках клавіатура, намагається найкращим способом вирішити поточне завдання. Другий програміст аналізує роботу першого і дає поради, обмірковує наслідки тих або інших рішень, нові тести, менш прямі, але гнучкіші рішення.

8. Колективне володіння кодом (Collective ownership) – будь-який розробник може покращувати будь-який код системи у будь-який час. Ніхто не повинен виділяти свою власну область відповідальності, уся команда в цілому відповідає за увесь код.

9. Безперервна інтеграція (Continuous integration) – система інтегрується і будується багато разів в день, у міру завершення кожного завдання. Безперервне регресійне тестування, тобто повторення попередніх тестів, гарантує, що зміни вимог не приведуть до регресу функціональності.

10. 40-годинний тиждень (40 – hour week) – як правило, працюють не більше 40 годин в тиждень. Не можна подвоювати робочий тиждень за рахунок наднормових робіт.

11. Локальний замовник (On – site customer) – в групі увесь час повинен знаходитися представник замовника, дійсно готовий відповідати на питання розробників. Його обов'язком є досить оперативні відповіді на питання будь-якого типу, що стосуються функцій системи, її інтерфейсу, необхідної продуктивності, правильної роботи системи в складних ситуаціях.

12. Стандарти (стили) кодування (Coding standards) – повинні витримуватися правила, що забезпечують однакове представлення програмного коду в усіх частинах програмної системи. Код розглядається як найважливіший засіб спілкування усередині команди. Ясність коду – один з основних пріоритетів.

13. Відкритий робочий простір (Open workspace). Команда розміщується в одному, досить просторому приміщенні, для спрощення комунікації і можливості проведення колективних обговорень при плануванні і ухваленні важливих технічних рішень.

14. Зміна правил з потреби (just rules). Кожен член команди повинен прийняти перераховані правила, але при виникненні необхідності команда може поміняти їх, якщо усі її члени прийшли до згоди з приводу цієї зміни.

Недоліками цього підходу деякі фахівці вважають нездійсненність в такому стилі досить великих і складних проектів, неможливість планувати терміни і трудомісткість проекту на досить довгу перспективу і чітко передбачити результати тривалого проекту в термінах співвідношення якості результату і витрат часу і ресурсів.

Scrum методологія

Перевагами XP є велика гнучкість і простота розуміння, проте в повному об'ємі XP не була використана навіть її авторами. Крім того, відомі і успішно впроваджуються окремі практики XP, як, наприклад, парне програмування, колективне володіння кодом, і, звичайно ж, рефакторинг кода. Ідея простого, ненадмірного дизайну проекту також зробила значний вплив на світ розробників ПО.

Більше практичним «гнучким» методом розробки є Scrum [35].

Історія

У 1986 японські фахівці Hirota Takeuchi і Ikujiro Nonaka опублікували повідомлення про новий підхід до розробки нових сервісів і продуктів (не обов'язково програмних). Основу підходу складала згуртована робота невеликої універсальної команди, яка розробляє проект на усіх фазах. Наводилася аналогія з регбі, де уся команда рухається до воріт супротивника як єдине ціле, передаючи (пасуючи) м'яч своїм гравцям як вперед, так і назад. На початку 90-х років цей підхід став застосовуватися в програмній індустрії і набув назви **Scrum** (термін з регбі, що означає, – сутичка), в 1995 році Jeff Sutherland і Ken Schwaber представили опис цього підходу на OOPSLA '95 – одній з найавторитетніших конференцій в області програмування. Відтоді метод активно використовується в індустрії і багаторазово описаний в літературі.

Загальний опис

Метод Scrum дозволяє гнучко розробляти проекти невеликими командами (7 чоловік плюс/мінус 2) в ситуації вимог, що змінюються. При цьому процес розробки ітеративен і надає велику свободу команді.

Спочатку створюються вимоги до усього продукту. Потім з них вибираються найактуальніші і створюється план на наступну ітерацію. Впродовж ітерації плани не міняються (цим досягається відносна стабільність розробки), а сама ітерація триває 2-4 тижні. Вона закінчується створенням працездатної версії продукту (робочий продукт), яку можна пред'явити замовникові, запустити і продемонструвати, хай і з мінімальними функціональними можливостями. Після цього результати обговорюються і вимоги до продукту коригуються.

Ролі

У Scrum є всього три види ролей.

Власник продукту (Product Owner) – це менеджер проекту, який представляє в проекті інтереси замовника. У його обов'язки входить розробка початкових вимог до продукту (Product Backlog), своєчасна їх зміна вимог, призначення пріоритетів, дат постачання і ін. Важливо, що він абсолютно не бере участь у виконанні самої ітерації.

Scrum-мастера (Scrum Master) забезпечує максимальну працездатність і продуктивну роботу команди – як виконання Scrum-процесса, так і рішення господарських і адміністративних завдань. Зокрема, його завданням є обгороджування команди від усіх дій ззовні під час ітерації.

Scrum-команда (Scrum Team) – група, що складається з п'яти-дев'яти самостійних, ініціативних програмістів. Першим завданням команди є постановка для ітерації реально досяжних і пріоритетних для проекту в цілому завдань (на основі Project Backlog і при активній участі власника продукту і Scrum-мастера). Другим завданням є виконання цього завдання щоб то не було, у відведені терміни і із заявленою якістю.

Практики

У Scrum визначені наступні практики.

Sprint Planning Meeting. Проводиться на початку кожного Sprint. Спочатку Product Owner, Scrum-мастер, команда, а також представники замовника і інші зацікавлені особи визначають, які вимоги з Project Backlog найбільш пріоритетні і їх слід реалізовувати у рамках цього Sprint. Формується

Sprint Backlog. Далі Scrum-мастер і Scrum-команда визначають те, як саме буде досягнута певна вище мета з Sprint Backlog. Для кожного елементу Sprint Backlog визначається список завдань і оцінюється їх трудомісткість.

Daily Scrum Meeting – п'ятнадцятихвилинна щоденна нарада, метою якої є досягти розуміння того, що сталося з часу попередньої наради, скоректувати робочий план згідно реаліям сьогоднішнього дня і позначити шляхи рішення існуючих проблем. Кожен учасник Scrum-команди відповідає на три питання: що я зробив з часу попередньої зустрічі, мої проблеми, що я робитиму до наступної зустрічі?

У цій нараді (15-20 хвилин) може брати участь будь-яка зацікавлена особа, але тільки учасники Scrum-команди мають право приймати рішення. На них лежить відповідальність за їх власні слова, і, якщо хтось з боку втручається і приймає рішення за них, тим самим він знімає відповідальність за результат з учасників команди.

Sprint Review Meeting. Проводиться у кінці кожного Sprint. Спочатку Scrum-команда демонструє Product Owner зроблену впродовж Sprint роботу, а той у свою чергу веде цю частину мітингу і може запросити до участі усіх зацікавлених представників замовника. Product Owner визначає, які вимоги з Sprint Backlog були виконані, і обговорює з командою і замовниками, як краще розставити пріоритети в Sprint Backlog для наступної ітерації. У другій частині мітингу робиться аналіз минулого спринту, який веде Scrum-мастер. Scrumкоманда аналізує в останньому Sprint позитивні і негативні моменти спільної роботи, робить висновки і приймає важливі для подальшої роботи рішення.

Хід роботи

Завдання 1. За допомогою пошуку в Інтернеті знайдіть інформацію про сучасні методології управління ІТ-проектами. Подайте підстави для їх класифікації. Для кожної основи наведіть приклади методологій.

Завдання 2. З отриманого списку важковагових методологій управління ІТ-проектами виберіть одну. Проведіть дослідження методології. Результат подайте в таблиці (таблиця 2).

Завдання 3. З отриманого списку легких методологій управління ІТ-проектами виберіть одну. Проведіть дослідження методології. Результат подайте в таблиці (таблиця 2).

Завдання 4. Виберіть будь-яку із проаналізованих методологій. Створіть про неї презентацію на 10-15 слайдів. Виступіть у групі, будьте готові відповісти на запитання.

Таблиця 2 - Особливості методики

Характеристика	Опис
Повна назва методології	
Автори	
Історія виникнення	
Країна появи	
Основні засади, підходи	
Програмні засоби реалізації методології	
Чи використовується в даний час	
Приклади успішних проєктів, реалізованих за допомогою даної методології	

Контрольні питання

1. Що таке методологія управління ІТ-проєктом?
2. Які види методологій ви знаєте?
3. У чому особливості великовагових та легких методологій управління?
4. Наведіть приклади методологій для розробки ІТ-проєктів.

Після закінчення заняття студент має:

1. Знати поняття методології управління ІТ-проектами, їхні види.
2. Наводити приклади різних методологій.
3. Перелічувати переваги великовагових та легких методологій.
4. Здійснювати вибір методологій управління під час роботи над ІТ-проектом.

Лабораторна робота 2

Аналіз ризиків

Мета роботи: здійснити аналіз ризиків методами «Матриця компромісів» та «Таблиця аналізу ризиків».

Теоретичні положення

Ризик органічно пов'язаний з прийняттям рішень. Рішення приймаються в умовах визначеності (результат рішення відомий), ризику (існує певна ймовірність того, що подія відбудеться і може бути проведена деяка оцінка), невизначеності (ймовірність і наслідки події передбачити неможливо). Процеси прийняття рішень щодо управління проектами відбувається в умовах невизначеності, тобто під впливом факторів: неповним знанням ситуації, наявності випадковості, наявності форс-мажорних обставин. Таким чином, реалізація проекту відбувається в умовах невизначеності і ризиків. Ці дві категорії взаємопов'язані.

Невизначеність – це неповнота чи неточність інформації щодо умов реалізації проекту, в тому числі пов'язаних із ними витратах і результатах. Джерелами невизначеності слугують: - стохастичний характер процесів, які відбуваються в господарській діяльності і суспільстві; - брак інформації, необхідної для обґрунтування і прийняття проектних рішень; - вплив суб'єктивних чинників на вироблення рішень (рівень кваліфікації виконавців, їх психологічний стан, свідоме приховання інформації тощо). За ступенем ймовірності настання події розрізняють повну невизначеність, часткову невизначеність і повну визначеність. Всі аспекти життя та підприємництва пов'язані з ризиками. В цілому ризик відображає дефіцит наших знань про майбутні події. При цьому сприятливі події ми називаємо можливостями, а несприятливі – загрозами. Ризик – це можливість чи загроза відхилення результатів конкретних дій від очікуваних.

Проектні ризики - сукупність ризиків, що загрожують реалізації інвестиційного проекту чи можуть знизити його ефективність (комерційну, економічну, бюджетну, соціальну, екологічну тощо); сукупність обставин за

яких ймовірність завершення поставлених цілей проекту зменшується або виключається; сукупність ризиків, які зумовлюють загрозу економічній ефективності проекту, що виражається в негативному впливі різних чинників на грошові потоки. Ризик має три основні атрибути: 1) випадок, що містить ризик; 2) ймовірність; 3) наслідок (дія ризику).

Є декілька видів випадків, які містять ризик для проекту: 1. Випадки, які можуть статися. 2. Випадки, які матимуть великі наслідки, якщо вони відбудуться. 3. Випадки поза вашим контролем. 4. Випадки, про які вам відомо дуже мало.

Існує два види ризику, пов'язаного з підготовкою і реалізацією проекту: систематичний і несистематичний. Систематичний ризик належить до зовнішніх щодо проекту чинників, приміром, стан економіки в цілому, і перебуває поза загальним контролем над виконанням проекту. Прикладами систематичного ризику є також політична нестабільність, умови оподаткування, тобто чинники, пов'язані з діями держави. Інші види систематичного ризику відбивають вплив чинників конкурентного середовища, як-от загальний ринковий попит, рівень конкуренції, ціни на сировину і робочу силу в галузі. Означені чинники мають розглядатися, бо проект замалий для того, щоб впливати на зміну цих чинників. Несистематичним є ризик, що безпосередньо стосується проекту. Рівень рентабельності виробництва, період початку будівництва і сам процес будівництва, вартість основного капіталу і продуктивність — усе це є видами несистематичного ризику. Інші види несистематичного ризику включають у себе зовнішні чинники, які можна контролювати або впливати на них у межах проекту. Це — заробітна плата персоналу проекту, ціни збуту продукції проекту, ціни постачальників на сировину і навіть урядові податки, як-от митний та акцизний збори, інші види податків.

Для глибшого розуміння компромісів програмного проекту часто буває дуже корисно зобразити замовнику залежність основних характеристик проекту у вигляді трикутника компромісів.

Затверджена рівновага із замовником досягається в тому випадку, якщо виконавець з урахуванням запитуваних параметрів назвав і зафіксував час (строки) та ресурси (кошторис). Слід враховувати, що будь-яка зміна однієї зі сторін трикутника обов'язково тягне зміну на двох, що залишилися.

Управління проектами передбачає не тільки констатацію факту наявності невизначеності й ризиків та аналіз збитку. Ризиками проекту можна й потрібно управляти. Управління ризиками – сукупність методів аналізу й нейтралізації факторів ризиків. Управління ризиками є підсистемою управління проектами. Управління ризиками – це процеси, пов'язані з ідентифікацією, аналізом ризиків і прийняттям рішень, які включають максимізацію позитивних і мінімізацію негативних наслідків настання ризикових подій; включає в себе процеси, пов'язані з плануванням управління ризиками, їх ідентифікацією та аналізом, реагування на ризики, а також контролю та управління ризиками в рамках проекту. Ціль управління проектними ризиками – підвищення ймовірності позитивних для цілей проекту подій і зниження ймовірності несприятливих подій.

Згідно з РМВОК виділяють наступні процеси управління проектними ризиками: 1. Планування управління ризиками – вибір підходів і планування діяльності щодо управління ризиками проекту 2. Ідентифікація ризиків – визначення ризиків, які здатні вплинути на проект, і документування характеристик цих ризиків. 3. Якісний аналіз ризиків - процес розстановки пріоритетів щодо ризиків для їх подальшого аналізу або дій, що виконується шляхом оцінки і зіставлення їх впливу і ймовірності виникнення. 4. Кількісний аналіз ризиків - процес чисельного аналізу впливу ідентифікованих ризиків на цілі проекту в цілому. 5. Планування реагування на ризики – розробка можливих варіантів і дій, які сприяють підвищенню сприятливих можливостей і зниженню загроз для досягнення цілей проекту 6. Моніторинг і контроль ризиків - моніторинг ризиків, визначення ризиків, що залишилися, виконання плану управління ризиками проекту й оцінка ефективності дій з мінімізації ризиків.

Планування управління ризиками - процес прийняття рішень по застосуванню й плануванню управління ризиками для конкретного проекту. Цей процес може включати в себе рішення по організації, кадровому забезпеченню процедур управління ризиками проекту, вибору кращої методології, джерел даних для ідентифікації ризику, часового інтервалу для аналізу ситуації. Важливо спланувати управління ризиками, адекватне як рівню й типу ризику, так і важливості проекту для організації. План управління ризиками є складовою частиною плану управління проектом. План управління ризиками є елементом плану управління проектом і включає: – методологію (ідентифікує і описує підходи, інструменти і джерела даних, які використовуються для роботи з ризиками); – ролі і обов'язки (визначає, хто і яку роботу виконує в ході управління ризиками проекту); – бюджет і строки (встановлює бюджет для управління ризиками, а також частоту процесів управління ризиками); – інструменти (описує, які контрольні заходи якісного і кількісного аналізу ризиків треба застосувати); – звітність і простеження (визначає формат плану реагування на ризики і звіту, способи документування результатів дій з управління ризиками, доведення інформації про них зацікавленим сторонам. Для оцінювання ризиків прийнята шкала виміру ризиків. Для розрахунку матриці ризиків слід визначити шкалу оцінки ймовірності ризику. Для цього слід обрати інтервал ймовірності і присвоїти йому числове значення. Існують трирівневі та семирівневі (більш деталізовані) розподіли ймовірності проектного ризику. Шкала оцінки наслідків ризику може різнитися в залежності від цілі, якої ризик потенційно може торкнутися, типу і розміру проекту, прийнятої в організації стратегії, її фінансового стану, а також від чутливості організації до конкретного виду наслідків. Відносна шкала наслідків містить лише описові позначення, наприклад «дуже низький», «низький», «середній», «високий» і «дуже високий», які розташовані в порядку зростання максимальної сили впливу ризику згідно з визначенням організації.

Для ефективного досягнення компромісів протягом усього життєвого циклу програмного проекту на початкових етапах слід виявити та зафіксувати пріоритет факторів (ресурси, час, можливості). Якщо один із факторів фіксується як незмінний, то впливати на нього протягом проекту практично неможливо. Другий фактор узгоджується за таким принципом: він матиме певний пріоритет у разі необхідності досягнення компромісів. Останній фактор просто приймається відповідно до перших двох. Матриця компромісів наведена в таблиці 3.

Таблиця 3 – Матриця компромісів

	Фіксується (Зафіксовано)	Узгоджується (Визначено)	Приймається (Коректовано)
Ресурси			
Час (графік)			
Можливості (набір функцій програми)			

Традиційна модель управління проектами передбачає чітке формулювання вимог на початковому етапі проекту, розробку виходячи з технічного завдання. Підхід компромісів ґрунтується на принципі проектних умов, що змінюються. Розробникам необхідно виявляти гнучкість і в будь-який момент часу бути готовим до змін та ризиків. Методологія Microsoft Solution Framework пропонує виокремлювати можливі ризики та аналізувати їх, щоб ними можна було керувати.

Ризиком називається проблема, яка ще виникла. У свою чергу проблемою називають ризик, який вже матеріалізувався. Причиною виникнення будь-яких ризиків є невизначеність у програмному проекті. Отже, потрібно прагнути виявлення ризиків. Іноді це здійснюється методом мозкового штурму, опитування експертів, методу Дельфі та ін. або на досвід попередніх проектів, досвід інших розробників.

Ризики ділять на два види:

1) відомі - ризики, які визначені, оцінені, для них можливе планування та аналіз;

2) невідомі – вони можуть бути заздалегідь ідентифіковані чи спрогнозовані.

Б. Боем наводить список 10 найпоширеніших ризиків програмного проекту [1]:

- 1) дефіцит спеціалістів;
- 2) нереалістичні терміни та бюджет;
- 3) реалізація невідповідної функціональності;
- 4) розробка неправильного інтерфейсу користувача;
- 5) «золоте сервірування», перфекціонізм, непотрібна оптимізація та відточування деталей;
- 6) безперервний потік змін;
- 7) брак інформації про зовнішні компоненти, що визначають оточення системи або залучені в інтеграцію;
- 8) недоліки у роботах, що виконуються зовнішніми (стосовно проекту) ресурсами;
- 9) недостатня продуктивність одержуваної системи;
- 10) «розрив» у кваліфікації спеціалістів різних галузей знань.

Сенс того, щоб описувати ризики та проводити їх аналіз зводиться до якомога більш ранньому виявленню цих ризиків, проведенню комплексу запобіжних заходів. Все це дозволить уникнути складних проблем вже на етапі реалізації проекту.

Усі ризики оцінюються у таблиці (таблиця 4).

Таблиця 4 - Аналіз ризиків

№ п/п	Ризик	Наслідки настання ризику	Заходи по запобігання	Заходи щодо мінімізації

Поля таблиці означають таке.

1. Під ризиком розуміється подія, яку слід однозначно і конкретно сформулювати, наприклад, погодні умови.

2. У третьому стовпчику оцінюються негативні наслідки матеріалізації цієї події.

3. Запобіжні заходи мають на увазі можливі дії команди, щоб уникнути матеріалізації ризику.

4. Заходи щодо мінімізації – це дії команди для скорочення втрат на випадок, якщо подія таки сталася.

Хід роботи

Робота у групі

Завдання 1. Створити матрицю компромісів до проекту «Кваліфікаційна робота» (таблиця 4).

Завдання 2. Розділитись на дві групи та методом мозкового штурму визначити список ризиків для проекту «Кваліфікаційна робота» (таблиця 5).

Таблиця 5 – Список ризиків

№ п/п	Ризик

Після обговорення об'єднати результати груп та методом експертних оцінок провести оцінку отриманого списку ризиків (таблиця 6).

Таблиця 6 – Експертна оцінка

Ризик	Експерт 1	Експерт ...	Експерт N	Рейтинг ризику

Завдання 3. З отриманого списку ризиків (таблиця 5) вибрати п'ять із найвищим рейтингом. Скласти таблицю «Аналіз ризиків проекту «Кваліфікаційна робота» (таблиця 4).

Завдання 4. Індивідуальна робота

Для обраного варіанта інформаційної системи (Додаток):

- заповнити матрицю компромісів (Таблиця 3);
- написати перелік ризиків (Таблиця 5);
- повести аналіз ризиків (Таблиця 4).

Контрольні питання

1. Що таке трикутник компромісів?
2. Коли і навіщо використовується трикутник компромісів?
3. До чого призводить зміни однієї із сторін трикутника компромісів?
4. Наведіть приклад заповненої матриці компромісів, наприклад для проекту челябінського метро.
5. Що таке ризик?
6. Назвіть типи ризиків.
7. Наведіть найпоширеніші ризики для програмного забезпечення проекту.
8. Навіщо керувати ризиками?
9. Як управляти ризиками?
10. Поясніть поля таблиці аналізу ризиків.

Після закінчення заняття студент має:

1. Знати що таке ризики.
2. Мати уявлення про види ризиків.
3. Знати методи аналізу ризиків.
4. Заповнювати матрицю компромісів для конкретного програмного забезпечення проекту.
5. Пояснювати методологію аналізу ризиків, зазначати її призначення.
6. Наводити приклади ризиків.
7. Здійснювати аналіз ризиків за допомогою таблиці аналізу ризиків.

Лабораторна робота 3

Порівняльний аналіз інформаційних систем

Мета роботи: провести аналіз аналогів - інформаційних систем з однієї предметної області - для виявлення вимог до програмного продукту, що розробляється.

Теоретичні положення

Ініціація – процес формального визнання необхідності виконання проекту. Основними причинами ініціації проектів виступають:

- вимоги ринку, наприклад: авторизація проекту реконструкції аеропорту великого науково-промислового й культурного центру як заявка на включення мегаполіса в міжнародну мережу авіамаршрутів;

- потреби бізнесу, наприклад: авторизація проекту будівництва елітного житлового комплексу девелопером (забудовником) з метою виходу на ринок житла premier-класу;

- попит споживачів, наприклад: авторизація проекту будівництва потужної технологічної лінії з випуску високоякісної бутильованої води у відповідь на зростаючі потреби в ній населення;

- технологічний розвиток, наприклад: авторизація проекту впровадження роздільного збору городянами твердих побутових відходів у результаті переорієнтації політики муніципальної влади в сфері обігу з відходами;

- юридичні вимоги, наприклад: авторизація проекту розробки міським управлінням транспорту й зв'язку нових інструкцій для проїзду в муніципальному транспорті з метою підвищення безпеки пересування в межах міста.

Ініціація проекту проходить через такі основні стадії:

- визначення проблеми, яку необхідно вирішити; – позначення вимірного очікуваного результату проекту;

- аналіз досяжності цілей проекту;

- ухвалення рішення про старт/скасування проекту;

- визначення пріоритетності проекту;

- призначення менеджера проекту;
- фіксація точки старту проекту.

Перераховані стадії можуть реалізовуватися одночасно. Підприємства часто недооцінюють стадію ініціації, приступаючи відразу в найкращому разі до планування, в гіршому – безпосередньо до реалізації. Однак значення ініціації важко переоцінити – саме на цій стадії відбувається обґрунтування проекту й аналіз досяжності його цілей. Недостатня увага до цих кроків найчастіше приводить до розпорошення зусиль підприємства на хаотичні ініціативи без видимого результату.

Обґрунтування проекту являє собою документ, що формально підтверджує обґрунтованість проекту, й містить опис:

- потреб бізнесу, на задоволення яких орієнтується проект;
- опис продукту, об'єкта, послуги.

Обґрунтування повинно складатися менеджером, зовнішнім щодо проекту, але на такому рівні ієрархії організації, який був би достатнім для задоволення потреб проекту. Обґрунтування дає менеджеру проекту можливість розподіляти ресурси підприємства по роботах проекту. У проектах, виконуваних за контрактом, сам контракт служить обґрунтуванням проекту. Однією з форм обґрунтування майбутнього проекту є бізнес-план. У бізнес-плані обґрунтовуються всі майбутні аспекти діяльності нового проекту, аналізуються можливі проблеми, які можуть виникнути. Актуальність бізнес-плану залежить від особливостей інноваційних проектів.

Бізнес-план — це докладний, чітко структурований і детально підготовлений документ, що описує цілі і задачі, які необхідно вирішити підприємству, способи досягнення поставлених цілей і техніко-економічні показники підприємства і/або проекту в результаті їх досягнення. У ньому міститься оцінка теперішнього моменту, сильних і слабких сторін проекту, аналіз ринку і інформація про споживачів продукції або послуг.

Бізнес-план: – дає можливість визначити життєздатність проекту в умовах конкуренції; – містить орієнтир розвитку проекту (підприємства, організації); –

служить важливим інструментом отримання фінансової підтримки від зовнішніх інвесторів.

Бізнес-план призначений, перш за все, для трьох категорій учасників проекту:

1) менеджерів — творців бізнес-плану, розробка якого, окрім вищезгаданих результатів, дозволяє отримати очевидні переваги від самого процесу планування;

2) власників, зацікавлених в складанні бізнес-плану з погляду перспектив розвитку фірми;

3) інвесторів — зазвичай банків, для яких бізнес-план є обов'язковим документом, підтверджуючим комерційну привабливість проекту.

Таким чином, бізнес-план дає можливість зрозуміти загальний стан речей на даний момент; ясно уявляти той рівень, якого може досягти проект (організація); планувати про цес переходу від одного стану в інший. Бізнес планування — загальноприйнята форма ознайомлення потенціальних інвесторів, кредиторів й інших партнерів з проектом, в якому їм пропонується взяти участь. Склад бізнес-плану і ступінь його деталізації залежать від розмірів майбутнього проекту і сфери, до якої він відноситься; розміру ринку збуту; наявності конкурентів; перспектив розвитку материнської організації.

У вітчизняній літературі склад розділів бізнес-плану досить повно розроблений і звичайно включає наступні розділи, що інтерпретуються в залежності від специфіки інноваційного проекту, галузі, цільової спрямованості управлінського рішення:

- сутність проекту (можливості підприємства, резюме);
- підприємство;
- продукція;
- ринки збуту продукції;
- конкуренція (відомості про конкуруючих проектах);
- маркетинг (стратегія маркетингу);

- виробничий процес;
- організаційний план;
- оцінка ризиків і страхування;
- фінансовий план;
- стратегія фінансування.

Бізнес планування проектів може вирішувати наступні завдання:

- диверсифікація, перепрофілювання та реорганізація діючого виробництва;
- підготовка заявок діючих і новостворюваних підприємств з метою отримання кредитів на створення нових, реконструкцію та розширення виробництв;
- обґрунтування будівництва нових підприємств;
- обґрунтування пропозицій щодо приватизації державних і муніципальних підприємств;
- створення нових підприємств, визначення профілю майбутньої фірми та основних напрямів її комерційної діяльності;
- вихід на зовнішній ринок і залучення іноземних інвестицій;
- розробка пропозицій щодо державної підтримки підприємств;
- використання як внутрішнього документа, що представляє оцінку діяльності фірми, виявлення її сильних і слабких сторін, формування цілей її діяльності, обґрунтування способів і тактики функціонування проекту, прогнозування майбутніх фінансових результатів та інших цілей.

Розробка проекту являє собою особливим чином організовану науково-дослідну роботу прогнозно-аналітичного і техніко-економічного характеру, зв'язану з постановкою мети розробки проекту, розробкою його концепції, планування і оформлення проектно-кошторисної документації. На стадії розробки (передінвестиційній фазі) виконують такі види робіт:

- визначення інвестиційних можливостей і висунення бізнес-ідеї;
- аналіз альтернативних варіантів проекту й попередній вибір проекту;

- підготовка проекту - розробка попереднього техніко-економічного (ПТЕО) і техніко-економічного обґрунтування (ТЕО);
- функціональні дослідження з проекту;
- висновок з проекту й рішення про інвестування.

Таке виконання передінвестиційної фази за стадіями дозволяє робити поетапну перевірку бізнес-ідеї й оцінювати альтернативні варіанти рішень. Причиною появи будь-якого проекту є нерозв'язана проблема. Але це повинна бути така проблема, яку неможливо розв'язати завдяки реалізації звичайної функціональної діяльності. Розв'язання проблеми пов'язане з необхідністю застосування нових, інноваційних підходів та ідей. Всі проблеми, які є причиною появи проектів, можна розділити на дві групи. Проблеми першої групи пов'язані зі стратегічною діяльністю організації і стосуються необхідності реалізації розроблених і затверджених стратегій розвитку. В цьому випадку чітко відомі глобальні (стратегічні) та операційні цілі. Ці проблеми пов'язані з пошуком шляхів досягнення операційної мети. Проблеми другої групи пов'язані з оперативною діяльністю організації. Вони завжди виникають раптово (непередбачено). Немає єдиної форми документа, який би регламентував спосіб описання проблеми. Тому важливо її описати так, щоб було зрозуміло, в чому полягає суть проблеми і як вона пов'язана зі стратегією розвитку соціально-економічної системи. Такий документ є основою для розроблення іншого – «Бізнес-ідеї проекту». Його головне призначення – представити інноваційну ідею з позиції можливості розв'язання конкретної проблеми. Формалізовано представлена в документі «Бізнес-ідея проекту» інформація є основою прийняття стратегічного рішення з проекту про перехід до етапу розроблення концепції проекту командою управління проектом. Якщо інноваційна ідея проекту загалом зацікавила власника організації, виникає потреба отримати точнішу та глибшу інформацію про майбутній проект з позиції його цінності для стратегії розвитку організації, тобто потрібно розширити та доповнити розділи «Бізнес-ідеї проекту». Цю роботу зазвичай

доручають групі фахівців з боку замовника. Результатом глибшого дослідження та уточнення цієї інформації є документ «Концепція проекту».

Під проектуванням розуміють процес створення проекту, тобто прототипу, прообразу передбачуваного чи можливого об'єкта чи стану, і навіть комплекту документації щодо нього.

Від специфічного для машинобудування, будівництва та інших галузей науки і техніки поняття «проект» (англ. design) у значенні «проектна документація» слід відрізнити поняття «проект», що використовується в контексті менеджменту (англ. project, від лат. projectus – кинутий вперед, виступаючий) у значенні «деяке завдання з певними вихідними даними та необхідними результатами (цілями), що зумовлюють спосіб її вирішення», «програма», «комплекс робіт» тощо [10].

Проект (від лат. projectus - кинутий вперед, виступаючий, видатний вперед) - це унікальний процес, що складається з сукупності скоординованих та керованих видів діяльності з початковою та кінцевою датами, вжитий для досягнення мети, що відповідає конкретним вимогам, що включає обмеження за термінами, вартістю та ресурсів [3].

З точки зору системного підходу проект може розглядатися як процес переходу з вихідного стану в кінцевий результат за участю низки обмежень і механізмів.

Проект - деяке завдання з певними вихідними даними та необхідними результатами (цілями), що зумовлюють спосіб її вирішення. Проект включає задум (проблему), засоби його реалізації (вирішення проблеми) і одержувані в процесі реалізації результати [8].

У загальному сенсі слова проект – це все те, що змінює якимось чином наш світ, робить його кращим. Вирішальну роль для проекту мають три складові: ціль, бюджет і час.

В свою чергу, від будь-якого завдання проект відрізняється одноразовістю дій. Серійне виробництво не має обмеження у часі, проект

обмежений за термінами реалізації рівно настільки, скільки потрібно для досягнення результату.

Можна виділити такі критерії проекту:

- новизна;
- наявність чіткої цільової установки;
- початок та закінчення проекту чітко визначені у часі;
- комплексність поставленої мети; проект в основному включає пов'язані між собою і залежні один від одного завдання і підзадачі різного рівня;
- участь кількох спеціалістів для досягнення мети.

Ознаки «не проекту»:

- мета однозначно не визначена, неконкретна чи недосяжна;
- терміни не визначені, нереальні задля досягнення мети;
- результат неунікальний.

Класифікувати проекти можна так:

- за рівнем проекту: проект, програма, система;
- за масштабом проекту: малий, середній, мегапроект;
- за складністю: простий, організаційно-складний, технічно-складний, ресурсно-складний, комплексно-складний;
- за термінами реалізації: короткостроковий, середній, мегапроект. Існують інші підстави.

Як показує статистика компанії The Standish Group [9], у світі завжди є успішні проекти (досягнули мети без перевитрати бюджету термінів реалізації), спірні (досягнули мети з перевитратою бюджету та/або строків реалізації) та провальні (не досягли мети).

Зазвичай як причини невдач називаються такі:

- неповні вимоги;
- низький ступінь залучення замовника та кінцевих користувачів у процес розробки;
- недостатнє забезпечення ресурсами;

– недоліки планування.

Хід роботи

Завдання 1. Здійснити у мережі Інтернет пошук готових інформаційних систем, які вирішують завдання з предметної області, обрану вами відповідно до варіанта (Додаток). Подати результат у вигляді списку інформаційних систем (таблиця 7).

Таблиця 7 - Програмні продукти з предметної галузі

№ п/п	Назва продукту	Назва фірми	Вимоги до системі	Можливості	Вартість

Завдання 2. З наведеної вище таблиці вибрати три програмних продукту та провести їх порівняльний аналіз. Результат: Показники товарів, представлені у таблиці 8.

Таблиця 8 – Порівняння програмних продуктів

№ п/п	перелік характеристик	Назва продукту №1	Назва продукту №2	Назва продукту №3
		Представлено характеристика чи ні	Представлено характеристика чи ні	Представлено характеристика чи ні

Завдання 3. На підставі таблиць зробити висновок, яким має бути ваша інформаційна система, щоб враховувати всі переваги та недоліки готових програмних продуктів. Результат подати у вигляді списку відмінностей.

Завдання 4. Для вашої системи скласти список тих користувачів, які будуть мати справу з програмним продуктом, що розробляється.

Завдання 5. Для кожного користувача визначити список його можливостей у вашій інформаційній системі (опис має бути зроблений мовою, зрозумілою користувачеві!).

Завдання 6. Обговоріть результати роботи в групі. Призначте відповідальних за ведення документації.

Контрольні питання

1. Що таке проект, програмний проект, проектування?
2. Чим відрізняється завдання від проекту, наведіть приклади.
3. Назвіть підстави класифікації проектів.
4. Які критерії успішності проектів?
5. З якою метою проводиться аналіз аналогів програмного продукту, що розробляється.
6. Для чого складається список користувачів програмного продукту?
7. Хто такі зацікавлені особи проекту?

Після закінчення заняття студент має:

1. Знати поняття проекту, програмного проекту, проектування.
2. Мати уявлення про критерії проекту та його відмінність від завдання.
3. Знати про класифікацію проектів.
4. Мати уявлення про успішність проекту.
5. Здійснювати аналіз програмних продуктів з предметної галузі з метою виявлення вимог до програмного проекту, що розробляється.
6. Формулювати список зацікавлених осіб та майбутніх користувачів програмного проекту.

Лабораторна робота 4

Планування проєкту

Мета роботи: здійснити часове планування проєкту.

Теоретичні положення

Управління терміном проєкту (Project Time Management) – розділ проєктного менеджменту, що включає процеси, необхідні для забезпечення своєчасного виконання робіт проєкту. Процеси управління термінами проєкту:

- планування управління розкладом - процес, який встановлює політику, процедури та документацію з планування, розробки, управління, виконання і контролю за розкладом проєкту;
- визначення операцій - процес ідентифікації та документування конкретних дій, які необхідно виконати для створення поставляються результатів проєкту;
- визначення послідовності операцій - процес виявлення і документування залежностей між операціями проєкту;
- оцінка ресурсів операцій - процес оцінки типів і кількості матеріалів, людських ресурсів, обладнання або поставок, необхідних для виконання кожної операції;
- оцінка тривалості операцій - процес оцінки кількості робочих періодів, необхідних для завершення окремих операцій з урахуванням оцінки ресурсів;
- розробка розкладу - процес аналізу послідовностей операцій, їх тривалість, потреб в ресурсах і обмежень розкладу для створення моделі розкладу проєкту;
- контроль розкладу - процес моніторингу статусу операцій проєкту для актуалізації прогресу проєкту та управління змінами базового розкладу з метою відповідності з планом.

Планування управління розкладом - процес, який встановлює політику, процедури і документацію з планування, розробки, управління, виконання і контролю за розкладом проєкту. Ключова вигода даного процесу полягає в тому, що він надає керівництво і вказівки щодо управління розкладом проєкту протягом усього проєкту. Визначення операцій - процес визначення конкретних операцій, які необхідно виконати для отримання результатів проєкту. У процесі розробки Ієрархічної Структури Робіт (ICR) визначаються результати самого нижнього рівня - пакети робіт. Пакети робіт проєкту

зазвичай розкладаються на більш дрібні елементи під назвою «операції», які описують роботу, необхідну для виконання пакету робіт. Операції надають основу для оцінки, планування, виконання, моніторингу та контролю робіт по проекту. Мається на увазі, що визначення і планування операцій розкладу в даному процесі проводяться таким чином, який забезпечує досягнення цілей проекту. Визначення послідовності операцій - процес визначення та документування взаємозв'язків між операціями проекту. Визначення послідовності операцій здійснюється за допомогою логічних взаємозв'язків. Кожна операція та контрольне подія, крім перших і останніх, пов'язані принаймні з одного попередньої і однієї подальшої операцією. Іноді буває необхідно використовувати час випередження або затримки між операціями для підтримки реалістичного і досяжного розкладу проекту. Визначення послідовності може бути виконано за допомогою програм управління проектами або за допомогою автоматичних або ручних методів. Наступні процеси (оцінка тривалості операцій, розробка розкладу, контроль розкладу розкладом) детально розглянуті в наступних питаннях.

Організаційно-технологічні моделі планування проектів

Початковим кроком у плануванні проекту є структуризація, яка передбачає планування обсягів робіт. Проте етап структуризації не дає змоги відповісти на запитання: скільки часу потрібно, щоб виконати всі роботи за проектом, якими є календарні терміни виконання окремих робіт, субпроектів, як розподіляється у часі потреба у різних ресурсах упродовж виконання проекту? Тобто постає потреба планування ще однієї головної мети проекту — виконання його у часі. Для вирішення цього завдання у проектному менеджменті застосовується сіткове і календарне планування. Враховуючи, що для успішної роботи над проектом менеджеру треба швидко опрацьовувати значний масив інформації, життєво необхідними стають такі спеціальні інструменти, як сітковий і календарний графіки. Їхня роль посилюється ще й тим, що вони поєднують у собі параметри часу, вартості й ресурсів.

Використання цих інструментів у плануванні проекту дає низку переваг, до яких належать можливості:

- визначити і наочно представити повний обсяг робіт у вигляді графіка;
- встановити такі цілі проекту щодо часу виконання робіт, вартості й обсягів ресурсів, що їх реально можна досягнути;
- оцінити бюджет проекту; – за ходом здійснення проекту контролювати виконання робіт і передбачати подальший перебіг подій;
- ефективно розподілити відповідальність за проектні роботи між членами команди; – визначивши критичні роботи, переміщувати ресурси, зменшувати ризики і невизначеність.

Сіткове планування

Організаційно-технологічна модель проекту – це адекватне формалізоване відображення порядку (послідовності) виконання робіт у часі, зв'язків і залежностей між ними, встановлених згідно з вимогами (технології, організації тощо) та з урахуванням обмежень (насамперед – ресурсних). Зазвичай моделі подають у формі графічних об'єктів – діаграм. Найпоширенішими у практиці управління проектами є такі види організаційно-технологічних моделей: лінійні діаграми (графіки Ганта), циклограми, сіткові моделі. Лінійну діаграму зображають лінійним календарним графіком. Циклограмну модель використовують переважно в разі застосування потокових методів організації виконання робіт проекту. Циклограми свого часу найширше застосовували у будівництві під час проектування спорудження однотипних будівель і споруд потоковим методом. Значно простішими є сіткові моделі. Сіткове моделювання ґрунтується на теорії графів. Сіткова модель – множина поєднаних між собою елементів для опису технологічної залежності окремих робіт і етапів майбутніх проектів.

Сіткове планування полягає у створенні логічних діаграм послідовності виконання проектних робіт — сіткових графіків — і визначенні тривалості цих робіт та проекту в цілому з метою подальшого контролю; набір методів, який

призначений для управління розкладом проекту. Застосування сіткового планування допомагає відповісти на такі запитання:

1. Скільки часу потрібно на виконання усього проекту?
2. У який час мають розпочинатися та закінчуватися окремі роботи?
3. Які роботи є «критичними» і повинні виконуватися точно за графіком, аби не зірвати строки виконання проекту у цілому?
4. На який термін можна відкласти виконання «некритичних» робіт, щоб це не вплинуло на строки виконання проекту?

Методи сіткового планування — це методи, основна мета яких полягає в тому, щоб зменшити до мінімуму тривалість проекту. До основних методів сіткового планування відносяться:

- метод критичного шляху (CPM);
- метод оцінки і аналізу програм (PERT);
- метод графічної оцінки і перегляду планів (GERT);
- метод критичних ланцюгів (CCM).

Останні три методи застосовують, якщо необхідно врахувати ситуації ризику чи невизначеності (щодо номенклатури, послідовності, тривалості робіт), тобто – при розробленні стохастичних (ймовірнісних) моделей проекту. У методі PERT номенклатуру та послідовність робіт задають однозначно, а їх тривалість – у формі розподілу ймовірності, тобто враховують ризик зміни часу виконання кожної роботи, а відтак – і усього проекту. Метод GERT передбачає можливість моделювання сценаріїв проекту, які відрізняються як переліком робіт, так і їх послідовністю і тривалістю. Такий підхід імітує ситуацію невизначеності. Метод критичних ланцюгів дозволяє урахувати використання у проекті обмежених ресурсів і передбачає оптимізацію організації їх руху. Він є прикладом розвитку та удосконалення методів CPM і PERT. Метод критичного шляху (CPM) — це метод планування робіт в рамках проекту, включаючи управління цими роботами і складання графіку їхнього виконання. Ключовим моментом методу є поняття «критичного шляху». Метод критичного шляху обчислює детермінований розклад виконання проекту,

базуючись на єдиній оцінці тривалості кожної роботи. Обчислюються ранні і пізні дати початку і завершення операцій проекту, а значить, і резерви — проміжки часу, на які можна зрушити 150 виконання операцій без порушення обмежень і дати завершення проекту. Відповідно до цього методу для кожного виду робіт вказуються час і ресурси, необхідні для їхнього виконання, а також послідовність виконання окремих видів робіт. Потім будується граф (сітковий графік), що відображає черговість робіт і терміни їхнього виконання.

Далі на цьому графі шукається критичний шлях, тобто шлях, що вимагає максимальних витрат часу. Метод критичного шляху в управлінні проектами є прикладом практичного застосування положень загальної теорії обмежень: будь-яка система має певні обмеження («вузькі місця»), які й визначають кінцеві результати її функціонування. У рамках концепції СРМ таким обмеженням є тривалість проекту, окремих робіт чи пакетів (технологічних комплексів), а критичним процесом — управління часом. При цьому припускають, що жодних ресурсних обмежень немає, тобто вважають, що всі необхідні ресурси у достатній кількості доступні. Однак основною перевагою методу критичного шляху є можливість маніпулювання термінами виконання робіт, що не лежать на критичному шляху. Метод PERT — це аналітичний розрахунковий метод, що дозволяє спробувати Далі на цьому графі шукається критичний шлях, тобто шлях, що вимагає максимальних витрат часу. гнотувати найоптимістичніші, найпесимістичніші та найвірогідніші терміни виконання робіт (у ході аналізу будується середньозважена оцінка), виключає при цьому повторення одних і тих же робіт в один і той же час. Він не допускає опису робіт з невідомою кількістю ітерацій, але може враховувати невизначені величини для підрахунку вірогідності виконання як окремих завдань, так і всього проекту у відведені терміни. Для кожного зі сценаріїв задається своя оцінка тривалості виконання робіт.

У своїх базових формах методи PERT і СРМ призначені для визначення найбільш тривалого за часом шляху в ланцюзі робіт, який стає основою при плануванні та контролі за ходом виконання проекту. Для графічного

відображення цієї послідовності в обох методах застосовуються лінії зі стрілками і вузлами. Спочатку PERT і СРМ відрізнялися між собою тим, що в мережевому графіку PERT операція позначалася стрілкою, а в СРМ - вузлом (кружком). Існувало й ще одна відмінність: в PERT використовувалися три типи оцінки тривалості операцій (оптимістична, песимістична і найбільш ймовірна), а в СРМ - тільки найкраща. Ці відмінності пояснюються тим, що метод PERT розроблявся для роботи зі складними проектами, які характеризуються високим ступенем невизначеності, а СРМ - для складання графіків рутинних операцій, пов'язаних із заводським технічним обслуговуванням. За довгі роки існування цих двох методів відмінності між ними стерлися, оскільки користувачі СРМ почали також застосовувати три оцінки тривалості операцій, а в мережевих графіках PERT нерідко позначаються вузлами.

Метод графічної оцінки і перегляду програм (Метод GERT). Метод графічної оцінки і перегляду програм (GERT) дозволяє проводити ймовірнісну обробку як сітьової логіки, так і оцінок тривалості робіт. GERT дає можливість врахувати ризик зміни складу робіт при настанні певних подій або за результатами виконання попередніх робіт: одні роботи можуть узагалі не виконуватися, інші — виконуватися частково, а треті виконуються кілька разів. Метод GERT дозволяє визначити очікувану тривалість робіт проекту на основі трьох ймовірнісних оцінок часу. Сіткова модель є ймовірнісною сіткою, що враховує можливість різного складу робіт проекту. Таким чином, можна врахувати не лише ризики (невизначеність) на рівні окремих робіт, а й на рівні проекту в цілому. Врахування ризиків, що впливають на тривалість робіт, здійснюється також, як і в методі PERT, тобто за результатами обчислення середньозваженої оцінки тривалості на базі трьох оцінок, виданих експертами.

В результаті моделювання по методу GERT з'явиться декілька графіків, що враховують ймовірність різної тривалості і невизначеності складу робіт проекту. Варто зауважити, що кожен із вказаних методів моделювання

ознаменував своєрідну революцію в методології та практиці управління проектами.

Календарне планування

Календарне планування – це складання та коригування розкладу, в якому роботи, виконані різними організаціями-учасниками проекту, погоджуються в часі між собою і з можливостями їхнього забезпечення різними видами ресурсів. При цьому повинне бути забезпечене дотримання заданих обмежень і оптимальний (за прийнятим критерієм) розподіл ресурсів.

Календарне планування проекту — це процес складання й коригування розкладу проекту, що полягає у визначенні календарних дат виконання всіх робіт.

Календарний план – план проекту, поданий у форматі реального часу (фактичних календарних дат). Календарне планування здійснюється на всіх етапах життєвого шляху проекту. Так, на етапі обрбування проекту розробляють укрупнений стратегічний план, на етапі підготовки формують базовий (цільовий) календарний план, а на етапі реалізації – детальні плани, які постійно коректують з урахуванням фактичного виконання завдань проекту. Процес календарного планування передбачає виконання таких кроків:

1. Ідентифікація проекту.
2. Структурування проекту.
3. Розроблення організаційно-технологічної моделі проекту.
4. Розроблення календарного плану виконання робіт проекту.
5. Розроблення календарного плану управління проектом.
6. Вартісна оцінка елементів проекту, визначення бюджету проекту.
7. Оптимізація планів за вибраним критерієм.

Календарний план (Schedule) як перелік тільки планових параметрів проектних робіт втрачає свій сенс без порівняння з фактичними термінами їх виконання, тому частіше ведуть мову про календарні графіки. Календарне планування ставить за мету координацію діяльності залучених до проекту виконавців для забезпечення його успішного завершення, створення умов задля

реагування на ринкові можливості та вчасного надходження доходів, що гарантує ефективність інвестицій. Цілі календарного плану: – забезпечити вчасне надходження фінансування; – координувати надходження ресурсів; – вчасно забезпечити потрібні ресурси; – передбачити у різні моменти рівень потрібних фінансових витрат і ресурсів та раціональний розподіл їх між проектами; – забезпечити вчасне виконання проекту. Календарний графік відбиває планові й фактичні дані про початок, кінець і тривалість кожного робочого елементу WBS. У ньому також відмічається можлива гнучкість у даті початку роботи без ускладнення виконання усього проекту (тобто запас часу по некритичних роботах). Для найскладнішого календарного графіку записується чотири версії для дат початку, кінця, тривалості та запасу: рання, пізня, запланована календарна, фактична. Мета календарного плану - координація діяльності залучених до проекту виконавців для забезпечення його успішного завершення, створення умов задля реагування на ринкові можливості та вчасного надходження доходів, що гарантує ефективність інвестицій.

Параметри календарного плану – це дати початку та закінчення кожної роботи, їх тривалість та необхідні ресурси. Тривалість роботи – головний параметр планування, залежить від сумарної трудомісткості (ТМ) та чисельності працюючих (ТМ:Чисел). Критична тривалість – мінімальна тривалість, протягом якої може бути виконаний весь комплекс робіт по проекту. Критичний шлях – шлях у сітковій моделі, тривалість якого дорівнює критичній. При календарному плануванні обов'язково повинно враховуватись дотримання заданих обмежень (тривалість робіт, ліміти ресурсів тощо) та оптимальний розподіл ресурсів. У ході реалізації проекту застосовуються різні типи календарних планів, які можна класифікувати за різними ознаками. Однією з ознак - за рівнем планування: – календарні плани проекту (розробляються до укладання контрактів); – функціональні календарні плани робіт (ФКПР). У свою чергу функціональні календарні плани робіт поділяються

1) за типами робіт: – ФКПР проектування; – ФКПР матеріально-технічного забезпечення; – ФКПР будівництва; – ФКПР введення в

експлуатацію і освоєння. – ФКПР також можуть бути складені: на окремі елементи, підсистеми, комплекси великого проекту, які в цьому випадку розглядаються як мініпроекти;

2) за глибиною планування: – перспективні графіки; – графіки початку й завершення робіт по проекту; – щомісячні, щотижневі, щоденні.

3) за формою подання: – логічні мережі; – графіки; – діаграми і т.д.

У цілому існують такі різновиди календарних планів: - календарний план за ранніми початками «жорстко ліворуч» – використовується для стимулювання виконавців проекту; - календарний план за пізніми закінченнями «жорстко праворуч» – використовується для подання проекту якнайкраще для споживачів; - календарний план «по середині» – створюється або для оптимізації споживаних ресурсів, або для показу замовнику найбільш імовірного результату. У повній системі календарного планування існує до 15 дат і моментів часу, що описують роботу. Процес складання календарного плану полягає у встановленні значень цих дат і моментів часу. На першому кроці оцінюється тривалість роботи, на другому – дати її початку й закінчення, де: планова тривалість = планове закінчення – плановий початок; плановий резерв часу = пізнє закінчення – планове закінчення. Види календарних графіків Існує два прийнятних шляхи подання календарного графіка: — табличний — з переліком робіт із зазначенням тривалості їх виконання; — діаграмний (балочні діаграми, або діаграми Ганта).

Як показує практика управління проектами, використання методик планування та контролю за ходом робіт не тільки дозволяє прискорити виконання проекту, а й значно зменшує витрати на реалізацію програми.

Після того, як визначено мету проекту, необхідно скласти список робіт для виконання. Менеджери проекту рекомендують скласти та дотримуватися плану проекту. Цей документ чітко вказує контрольні точки проекту та основні дії, необхідні для досягнення мети проекту. Крім того, він визначає дату кожної контрольної точки (завершення головних дій щодо досягнення цілей) та

відповідальних за кожну дію. План складається на початковому етапі, схвалюється командою проектувальників.

План може бути складений на прикладі подібних проектів, або це може бути просто перелік дій у формальному вигляді. Необхідно завжди записувати фактичні дати завершення кожного етапу та/або дії. Якщо дата відрізняється від плану, потрібно провести коригування.

При включенні необхідних пунктів до плану слід:

- застосовувати унікальні ідентифікатори (ID), які можна використовувати у разі, якщо необхідні оновлення;
- давати назви завданням;
- вказувати на початок виконання дії;
- вказувати дату завершення виконання;
- записувати фактичну дату завершення виконання дії;
- будь-яке завдання завершувати до того, як буде розпочато наступне;
- вказувати відповідального за виконання (господаря завдання);
- вказувати відсоток завершеності кожної дії.

Зазвичай призначається відповідальний для здійснення контролю за виконанням плану та внесення змін до нього.

Одним із способів контролю над часом, планування проекту може бути діаграма Ганта.

Діаграма Ганта(Gantt chart, стрічкова діаграма, графік Ганта) – тип стовпчастих діаграм, що стали популярними у проектному менеджменті. Використовується для подання плану проекту чи графіка робіт [7].

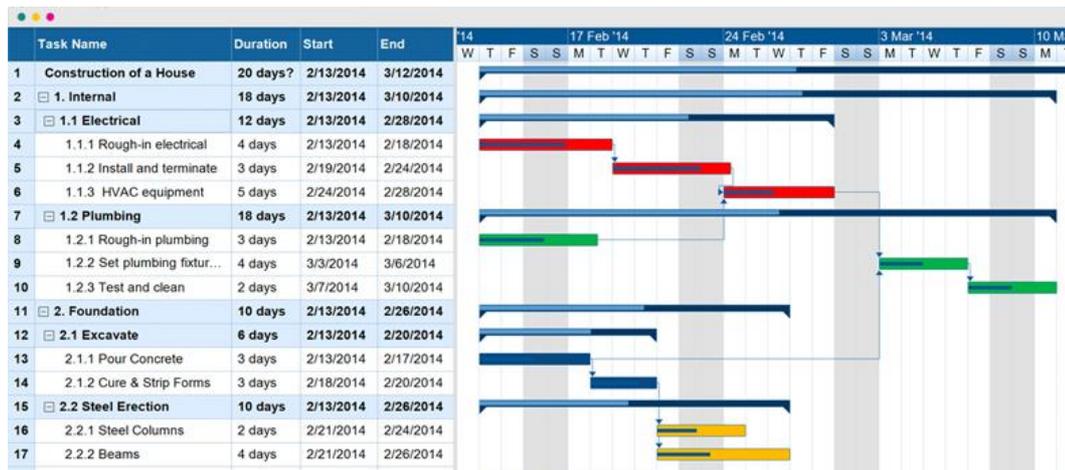


Рис. 2. Діаграма Ганта.

Г.Л. Гант (1861 – 1919) вивчав менеджмент з прикладу будівництва кораблів. Він запропонував свій варіант діаграми під час Першої світової війни, коли потрібно відстежувати хід будівництва великих трансконтинентальних океанських лайнерів. Така діаграма складалася з відрізків (завдань) та точок, т.з. "завершальних завдань", або "віх". І виступала засобом уявлення тривалості та послідовності завдань у проекті.

Ідея планування у тому, що його головним ресурсом є час. У зв'язку з цим основою прийняття управлінських рішень є порівняння запланованого стану робіт і фактичного. На діаграмах по горизонталі вказуються інтервали часу, по вертикалі – операції, роботи чи устаткування. Горизонтальні відрізки ілюструють тривалість виконання робіт. Вибравши по горизонталі поточний момент часу та отримавши оперативну інформацію про хід проекту, можна зіставити стан справ за фактом та стан справ за планом.

Діаграма Ганта дозволяє:

- візуально оцінити послідовність завдань, їх відносну тривалість та протяжність проекту в цілому;
- порівняти запланований та реальний хід виконання завдань;
- детально проаналізувати реальний хід виконання завдань (наприклад, завдання виконувалося, було припинено, поверталось на доопрацювання тощо).

Методика вперше була представлена у 1910 р. Згодом діаграма Ганта стала головним інструментом, що використовується у календарному плануванні та контролі. У 1990-х роках. методику було вдосконалено: для опису залежностей між завданнями були додані зв'язки.

Розглянемо типи зв'язків:

1. Фініш-старт. Цей зв'язок означає, що операція В не може початися до завершення операції А, або дата закінчення операції А визначає дату початку операції В. Наприклад, треба спочатку написати диплом, а потім його можна захищати.

2. Фініш-Фініш - операція В повинна закінчитися не раніше операції А, або дата закінчення операції А визначає дату закінчення операції В. Наприклад, якщо Ви пишете клієнт-серверний додаток (операція А) і для його налагодження ви брали в оренду сервер (операція В) , то налагодження має завершитись до терміну закінчення оренди сервера.

3. Старт-Старт - операція В починається не раніше операції А, або дата початку операції А визначає дату початку операції В. Наприклад, операції друкування диплома тісно пов'язана з купівлею паперу і завдання повинні вирішуватися практично одночасно.

4. Старт-Фініш – операція В не може закінчитися доки не почнеться операція А, або дата початку операції А визначає дату закінчення операції В. Час, на який запланований захист диплома, визначає, коли мають завершитися передзахисту.

Однак діаграми мають недоліки: за їх допомогою складно планувати багатоваріантні взаємопов'язані ланцюжки робіт (для будівельних, військових, державних проектів та на виробництві). Для таких завдань запропоновано методи мережного планування, або методи вибору «критичного шляху», розроблені військовим відомством США у 1950-х роках. Діаграми зручно застосовувати лише одного ресурсу – часу. Якщо враховувати ще кілька ресурсів, то діаграми Ганта треба сприймати в об'ємному вигляді, що є сенсом для візуальної інтерпретації планів з одного

боку, але ускладнює їхній аналіз з іншого. Для управління проектами зазвичай використовуються інші, потужніші засоби – метод «критичного шляху» (Critical Path Method, CPM) та метод PERT (Program Evaluation Review Technique).

Хід роботи

Завдання 1. Створіть діаграму Ганта за проектом "Зимова сесія".

Послідовність дій.

1. Створити та заповнити таблицю з перерахуванням етапів, датами початку, тривалості кожного етапу та кінця.
2. Для виділеної таблиці створити діаграму (тип "Лінійчаста з накопиченням").
3. На вкладці "Діапазон даних" вибрати "Ряди в стовпцях".
4. На вкладці "Ряд" натиснути кнопку "Додати", встановити курсор поле «Значення» та виділити осередки з тривалостями етапів.
5. Приберіть "Легенду".
6. У контекстному меню вертикальної осі із назвами етапів вибрати пункт «Формат осі».
7. На вкладці «Шкала» поставити дві галочки: «Зворотний порядок категорій» та «Перетин з віссю Y у максимальній категорії».
8. Зробіть подвійне клацання мишею по будь-якому з синіх стовпчиків, встановіть прозору заливку і невидиму рамку.
9. Налаштуйте діапазон даних, що відображаються на діаграмі. Для цього необхідно дізнатися реальний вміст осередків, з яких починається і на яких закінчується тимчасова шкала (перший осередок у стовпці Початок та останній у стовпці Кінець). Справа в тому, що Excel тільки відображає в клітинці дату як день-місяць-рік, а насправді будь-яку дату зберігає в клітинці як кількість днів, що пройшли з 1.1.1900 до поточної дати. Виділіть жовту та зелену комірки та по черзі спробуйте встановити для них «Спільний формат» (меню «Формат» – «Комірки»). Наприклад, вийшло 38350 та 38427, відповідно.

Додамо до дати закінчення ще три дні – отримаємо 38340. Запам'ятаємо ці цифри.

10. Клацніть правою кнопкою миші по горизонтальній осі часу і вибрати "Формат осі" і ввести ці числа на вкладку "Шкала": мінімальне та максимальне значення відповідно.

Завдання 2. Створіть діаграму Ганта за таблицею, що відображає етапи проекту.

За допомогою умовного форматування можна змусити Excel заливати комірку будь-яким вибраним кольором, якщо вона за датою потрапляє між початком та кінцем етапу. Найпростіше для цього використовувати логічну функцію I, яка в даному випадку перевіряє обов'язкове виконання обох умов (5 січня пізніше, ніж 4 і раніше, ніж 8).

Завдання 3 (індивідуально). Створіть діаграму Ганта за програмним проектом, що розробляється, онлайн засобами (Додаток).

Контрольні питання

1. Навіщо проводиться тимчасове планування проекту?
2. Вкажіть призначення діаграм Ганта.
3. У чому особливість діаграм Ганта?
4. Назвіть типи зв'язків на діаграмі Ганта, наведіть приклади.
5. Які програмними продуктами можна побудувати діаграми Ганта?

Після закінчення заняття студент має:

1. Знати про призначення тимчасового планування проекту.
2. Називати методи розроблення тимчасового планування.
3. Створювати діаграми Ганта у різний спосіб.
4. Здійснювати тимчасове планування програмного проекту.

Лабораторна робота 5

Виявлення потреб проєкту

Мета роботи: виявити потреби зацікавлених у проєкті осіб

Теоретичні положення

Процес роботи з вимогами до продукту можна поділити на чотири етапи:

1. Визначення продукту концепції.
2. Збір вимог.
3. Аналіз вимог.
4. Проектування системи.

На етапі визначення концепції товару проводиться робота з його інвестором. Мета етапу - вироблення єдиного бачення майбутнього продукту. По закінченні цього етапу робиться висновок про те, буде цей продукт розроблятися чи ні.

На етапі збору вимог ведеться основна робота із замовником системи та її майбутніми користувачами. Мета етапу – точно визначити функції продукту та способи його інтеграції у існуючі процеси.

Якісне виконання робіт на цьому етапі гарантує, що майбутній продукт буде відповідати очікуванням замовника. Чітка розстановка пріоритетів забезпечує реалізацію найбільш затребуваної функціональності та виключення другорядної/непотрібної функціональності, що заощадить бюджет та терміни.

На етапі аналізу вимог здійснюється структуризація вже зібраних вимог. Мета етапу – надати чіткий список не дубльованих вимог до системи, які мають бути виділені з надмірних і частково дублюючих сценаріїв та історій користувача, отриманих на попередньому етапі.

Правильно згруповані вимоги допоможуть обійтися мінімальною кількістю функціонала для задоволення максимально великої кількості цілей, а це, у свою чергу, допоможе заощадити бюджет проєкту.

На етапі проектування системи група розробки приймає проектні рішення про те, яку функціональність нестиме продукт, щоб задовольнити користувачів. Результатом є закінчене технічне завдання (ТЗ) до продукту: повний опис поведінки майбутнього продукту без неоднозначностей та питань.

На основі ТЗ починається моделювання роботи продукту з кінцевими користувачами та проводиться його тестування. Це дозволяє збільшити якість продукту та знизити його вартість, оскільки вартість внесення змін до технічного завдання завжди менша, ніж у кінцевий продукт.

Однак при роботі із замовниками проекту та майбутніми користувачами програми виникають різні ситуації. Склали ситуації часто називають «синдромами». Виділяють три основні «синдроми».

1. Синдром «Так... Але!». Це виникнення протилежних реакцій користувача, коли він уперше бачить реалізацію системи. З одного боку, це те, про що він говорив, з іншого – це не те, що він очікував побачити.

2. Синдром "невідкритих руїн". Згодом з'являється певна закономірність: чим більше виявлено вимог, тим більше деталей замовник прагне додати до своїх вимог.

3. Синдром «Говорити мовою користувача». Під час спілкування із замовником не слід використовувати професіоналізм – терміни, які можуть бути невідомі користувачеві або неправильно ним витлумачені.

Дисципліна управління проектами пропонує різні методи виявлення вимог, проблем та бажань замовника:

- інтерв'ю та анкети;
- семінари/наради (зацікавлені особи збираються для інтенсивних, насичених дискусій);
- сценарії програми (використання візуальних/графічних інструментів для демонстрування поведінки системи) для виключення синдрому «Так... Але!»;

- рольові ігри (кожному члену групи призначається певна роль, зазвичай роль одного з користувачів);
- метод "мозкового штурму";
- використання прототипів (якомога раніше надати користувачеві інтерфейс);
- сценарії використання (Use Cases – взаємодія між користувачем та системою, представлена у вигляді послідовності кроків);
- аналіз існуючих документів (витяг інформації з документів Microsoft Word, електронної пошти та записів);
- спостереження та демонстрування завдань (спостереження за користувачами, що виконують певне завдання);
- аналіз існуючих систем (збір вимог від морально застарілих систем або від систем, розроблених у ході конкуренції).

Розглянемо особливості способу інтерв'ю. Його перевага – інтерактивність, що надає можливість внесення доповнень чи доопрацювання питань залежно від отриманих відповідей. На сьогоднішній день це хороший спосіб зібрати вимоги щодо зручності використання системи, надійності, продуктивності та зручності супроводу. Зазвичай замовники не згадують цих нефункціональних вимог, поки їх явно не запитати про це.

При виборі заінтересованих осіб для інтерв'ю слід переконатися, що команда розробників розуміє, яку саме групу заінтересованих осіб вони представляють.

Можна слідувати наступним порадам.

1. Рекомендується написати список запитань.
2. Бажано проговорити відповіді своїми словами, щоб переконатися, що Ви розумієте сенс.
3. Не слід пропонувати відповідь на це запитання. (Який час реакції системи Ви очікуєте? Три секунди?).

4. Не слід поєднувати кілька запитань до одного. (Чи потрібно Вам друкувати відповідь, надсилати її електронною поштою та факсом?).

Можливо, користувачеві потрібна лише можливість друку звіту та надсилання його електронною поштою, але немає потреби у факсі.

5. Не слід запитувати користувача про деталі реалізації. (Ви віддаєте перевагу list-box або radio-buttons для вибору методу оплати?).

6. Не слід використовувати надто довгі та складні питання.

7. Не рекомендується ставити наступне питання, якщо ще не отримано відповіді на попереднє.

8. У ситуації, коли відповідь незрозуміла, варто поставити додаткові питання, навіть якщо їх немає у сценарії інтерв'ю.

9. Не варто перебивати користувачів, коли вони відхиляються від теми. Треба дозволити їм висловити свої думки, на яку б тему вони не думали. Якщо відповіді на початкове питання не отримано, слід поставити його знову.

10. Фіксувати кожен згаданий користувачем вимогу, навіть якщо зараз вона здається недоречною.

11. Обов'язково слід запитати користувачів щодо додаткової інформації (екранні форми системи).

12. При розмові із замовниками не варто говорити, чи буде їхня вимога виконана чи ні. Це рішення можна ухвалити пізніше.

13. Наприкінці розмови обов'язково поставте запитання для отримання додаткової інформації. (Що ще я маю знати?).

14. Після отримання списку вимог слід з'ясувати у зацікавленої особи пріоритет кожної вимоги.

15. Під час інтерв'ю варто робити примітки та/або використовувати записуючий пристрій.

16. Питання мають бути контекстно-вільними, тобто не містити бажаної відповіді.

17. Усі вимоги заносяться до «архіву вимог», тобто мають документуватися.

Хід роботи

Завдання 1.Зверніться до списку користувачів та зацікавлених осіб для проекту Автоматизована інформаційна система «Університет» (АІС «Університет»). (За бажанням студентів можна здійснити реалізацію іншого програмного проекту. Наприклад, "Склад", "Поліклініка", "Сайт організації", "Пошук туру" тощо).

Завдання 2. Розподіліть у групі ролі згідно зі списком, отриманим Завдання 1. Наприклад, Ректор, Проректор, Декан, Завідувач кафедри, Викладач кафедри, Студент, Секретар деканату тощо. Вивчіть можливі посадові обов'язки (пошук у мережі Інтернет), обговоріть з викладачем можливі потреби, проблеми, що виникають з виконанням посадових обов'язків, складіть легенду вашого користувача.

Завдання 3. Реалізуйте ділову гру. Кожен учасник по черзі відіграватиме роль обраного ним користувача, решта – члени команди розробників. Методом інтерв'ювання виявите потреби, проблеми користувача, що підлягають вирішенню у проекті. Пам'ятайте, що з користувачем необхідно спілкуватися його мовою. Обов'язково ведіть документування даних.

Завдання 4. Відповідно до своєї ролі заповніть таблицю опису зацікавлених осіб та користувачів (таблиця 9).

Контрольні питання

1. Хто такі зацікавлені особи проекту?
2. Як пов'язані поняття «зацікавлена особа» та «користувач»?
3. Якими методами здійснюється виявлення потреб зацікавлених осіб проекту?

4. Які переваги методу інтерв'ювання?
5. Вкажіть основні засади проведення інтерв'ювання.
6. Чи завжди замовник проекту має уявлення про реальні потреби?

Таблиця 9 - Характеристика зацікавленої особи

Представник	Хто у проекті є представником користувача? Можна, можливо посилатися на зацікавлених осіб
Опис	Короткий опис типу користувача
Тип	Рівень знань користувача, його технічна освіта та ступінь обізнаності. Наприклад, гуру, випадковий користувач
Відповідальність	Список ключових відповідальностей користувача по відношенню до системи, що розробляється, тобто. фіксує деталі, складає звіти, координує роботу тощо.
Критерій успіху	Як користувач бачить успіх? Як компенсується працю користувача?
Залучення	Як користувач може бути залучений до проекту (рецензування вимог, архітектурних та технічних рішень, тестування ПЗ і т.д.)?
Постачані артефакти (документи)	Чи існують якісь вихідні артефакти, які потрібні користувачеві? Якщо так, то які (наприклад, звіти про..., зведення за ... і т.д.)?
Коментарі/ Проблеми	Проблеми, що заважають досягненню успіху, та будь-яка подібна інформація. Можна включати тенденції, які роблять роботу користувача простіше чи важче.

Після закінчення заняття студент має:

1. Знати основні переваги методу інтерв'ювання виявлення потреб зацікавлених осіб проекту.
2. Знати поняття «зацікавлені особи проекту».

4. Мати уявлення про засади проведення інтерв'ю з користувачем.
5. Вміти проводити інтерв'ю з користувачем.
6. Здійснювати ведення документації: виявлених та зафіксованих потреб зацікавлених осіб проекту.

Списки інформаційних систем для проектування

Варіант 1. Реєстратура поліклініки.

Варіант 2. Обробка звернень громадян до служби ЖКГ.

Варіант 3. Електронний документообіг невеликої фірми.

Варіант 4. Продаж авіаквитків.

Варіант 5. Туристична фірма.

Варіант 6. Квиткова система (Музеї).

Варіант 7. Квиткова система (Театр).

Варіант 7. Порушення правил дорожнього руху.

Варіант 8. Розробити програмний модуль «Облік успішності студентів». Програмний модуль призначений для оперативного обліку успішності студентів у сесію деканом, заступниками декана та співробітниками деканату. Відомості про успішність студентів повинні зберігатися протягом усього терміну їх навчання та використовуватися при складанні довідок про прослухані курси та додатки до диплому.

Варіант 9. Розробити програмний модуль «Особисті справи студентів». Програмний модуль призначений для отримання відомостей про студентів співробітниками деканату, профкому та відділу кадрів. Дані повинні зберігатися протягом усього терміну навчання студентів та використовуватися при складанні довідок та звітів.

Варіант 10. Розробити програмний модуль «Рішення комбінаторно-оптимізаційних завдань». Модуль повинен містити алгоритми пошуку циклу мінімальної довжини (завдання комівояжера), пошуку найкоротшого шляху та пошуку мінімального сполучного дерева.

Варіант 11. Розробити програмний модуль «Кафедра», що містить відомості про співробітників кафедри (ПІБ, посада, науковий ступінь, дисципліни, навантаження, громадська робота, сумісництво та ін.). Модуль призначений для використання співробітниками відділу кадрів та деканату.

Варіант 12. Розробити програмний модуль «Лабораторія», що містить відомості про співробітників лабораторії (ПІБ, стать, вік, сімейний стан, наявність дітей, посада, науковий ступінь). Модуль призначений для використання співробітниками профкому та відділу кадрів.

Варіант 13. Розробити програмний модуль "Хімчистка". При записі на обслуговування заповнюється заявка, в якій зазначаються ПІБ власника, опис виробу, вид послуги, дата прийому замовлення та вартість послуги. Після виконання робіт роздруковується квитанція.

Варіант 14 .Розробити програмний модуль "Облік порушень правил дорожнього руху". Для кожної машини (і її власника) в основі зберігається перелік порушень. Для кожного порушення фіксується дата, час, вид порушення та розмір штрафу. При сплаті всіх штрафів машина видаляється з бази.

Варіант 15. Розробити програмний модуль "Картотека автомагазину", призначений для використання працівниками агентства. В базі містяться відомості про автомобілі (марка, об'єм двигуна, дата випуску та ін). При надходженні заявки на покупку проводиться пошук відповідного варіанта. Якщо такого немає, клієнт заноситься в базу клієнта і повідомляється, коли варіант з'являється.

Варіант 16. Розробити програмний модуль "Картотека абонентів АТС". Картотека містить відомості про телефони та їх власників. Фіксує заборгованості з оплати (абонентської та погодинної). Вважається, що погодинну оплату місцевих телефонних розмов уже запроваджено.

Варіант 17. Розробити програмний модуль «Автокас», що містить відомості про наявність вільних місць на автобусні маршрути. В базі повинні міститися відомості про номер рейсу, маршрут, водія, тип автобуса, дату і час відправлення, а також вартість квитків. При надходженні заявки на квитки програма здійснює пошук відповідного рейсу.

Варіант 18. Розробити програмний модуль «Книгарня», що містить відомості про книги (автор, назва, видавництво, рік видання, ціна). Покупець оформляє заявку потрібні йому книжки, якщо таких немає, він заноситься у основу і сповіщається, коли необхідні книжки надходять у магазин.

Варіант 19. Розробити програмний модуль "Автостоянка". У програмі міститься інформація про марку автомобіля, його власника, дату і час в'їзду, вартість стоянки, знижки, заборгованість з оплати та ін.

Варіант 20. Розробити програмний модуль «Кадрове агентство», що містить відомості про вакансії та резюме. Програмний модуль призначений як пошуку співробітника, відповідального вимогам керівників фірми, так пошуку відповідної роботи.

СПИСОК ЛІТЕРАТУРИ

1. Андрієнко О. Управління проектами в бізнес-об'єднаннях малих і середніх підприємств: посібник Київ: 2017. - 77 с.
2. Буріменко Ю. І., Галан Л. В., Лебедева І. Ю. та ін. Управління проектами: навч. посіб. /за ред. Ю. І. Буріменко. Одеса: ОНАЗ ім. О. С. Попова, 2017. - 208 с.
3. Гонтарева І. В. Управління проектами: підручник Харків ХНЕУ, 2011. - 444с.
4. Козик В.В., Тимчишин І.Є. Практикум з управління проектами: навч. посібник. Львів: Видавництво Львівської політехніки, 2012. – 180 с.
5. Матвіїшин Є.Г. Планування проектних дій: навч. посіб. Київ: «Хай-Тек Прес», 2008. - 216 с.
6. Міцура О.О., Олефіренко О.М. Управління інноваційними проектами: конспект лекцій. Суми: Сумський державний університет, 2012. - 92 с.
7. Ноздріна Л.В., Ящук В.І., Полотай О.І. Управління проектами: підручник. Київ: Центр учбової літератури, 2010. – 432 с.
8. Петрович Й.М., Новаківський І.І. Управління проектами: підручник Львів: Видавництво Львівської політехніки, 2018. - 396 с.
9. Веретенников В. І. Управління проектами: Навч. посібник / В. І. Веретенников, Л. М. Тарасенко, Г. І. Гевлич. — К.: Центр навчальної літератури, 2006. — 280 с.
10. Тянь Р.Б., Холод Б.І., Ткаченко В.А. Управління проектами: підручник.- К. Центр навчальної літератури, 2003. – 224 с.
11. Корнієнко Б.Я. Дослідження імітаційного полігону захисту критичних інформаційних ресурсів методом IRISK. Моделювання та інформаційні технології. 2018. Вип. 83. С. 34-41.
12. Корнієнко Б.Я. Побудова та тестування імітаційного полігону захисту критичних інформаційних ресурсів. Наукоємні технології. 2017. № 4 (36). С. 316 - 322.

13. Korniyenko B., Yudin A., Galata L. Risk estimation of information system. *Wschodnioeuropejskie Czasopismo Naukowe*. 2016. № 5. P. 35 - 40.
14. Корнієнко Б.Я., Юдін О.К., Снігур О.С. Безпека аутентифікації у web-ресурсах. *Захист інформації*. 2012. № 1 (54). С. 20 -25. DOI: 10.18372/2410-7840.14.2056 (ukr).
15. Корнієнко Б.Я., Максимов Ю.О., Марутовська Н.М. Прикладні програми управління інформаційними ризиками. *Захист інформації*. 2012. № 4 (57). С. 60 – 64. DOI: 10.18372/2410-7840.14.3493 (ukr).
16. Galata, L., Korniyenko, B., Yudin, A.: Research of the simulation polygon for the protection of critical information resources. In: *CEUR Workshop Proceedings, Information Technologies and Security, Selected Papers of the XVII International Scientific and Practical Conference on Information Technologies and Security (ITS 2017)*, 30 Nov 2017, Kyiv, Ukraine. vol. 2067. pp. 23–31., urn:nbn:de:0074-2067-8.
17. Корнієнко Б.Я. Безпека інформаційно-комунікаційних систем та мереж. Навчальний посібник для студентів спеціальності 125 «Кібербезпека». – К.: НАУ, 2018. – 226 с.
18. Корнієнко Б.Я., Галата Л.П. Оптимізація системи захисту інформації корпоративної мережі. *Математичне та комп'ютерне моделювання*. Серія: Технічні науки, Випуск 19, 2019. - С. 56-62.
19. Korniyenko B., Galata L. Implementation of the information resources protection based on the CentOS operating system. *Conference Proceedings of 2019 IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON -2019)* July 2 – 6, 2019, Lviv, Ukraine. - pp. 1007-1011.
20. Галата Л.П., Корнієнко Б.Я., Заболотний В.В. Математична модель протидії загрозам у системі захисту критичних інформаційних ресурсів. *Наукоємні технології*, Том 43, № 3, 2019. – С. 300 – 306.

21. Корнієнко Б.Я. Modeling of information security system in computer network. *Безпека інформаційних систем і технологій*, Том №1 (1), 2019. – С.36-41.
22. Korniyenko B., Galata L., Ladieva L. Research of Information Protection System of Corporate Network Based on GNS3. *Conference Proceedings of 2019 IEEE International Conference on Advanced Trends in Information Theory (IEEE ATIT -2019) Dezember 18 – 20, 2019, Kyiv, Ukraine.* - pp. 244-248.
23. Korniyenko B., Galata L., Ladieva L. Mathematical model of threats resistance in the critical information resources protection system. *CEUR Workshop Proceedings, Selected Papers of the XIX International Scientific and Practical Conference "Information Technologies and Security" (ITS 2019) Kyiv, Ukraine, November 28, 2019. Vol-2577. P.281-291.*
24. Корниенко Б.Я. Кибернетическая безопасность – операционные системы и протоколы. ISBN 978-3-330-08397-4, LAMBERT Academic Publishing, Saarbrucken, Deutschland. – 2017. – 122 с.
25. Korniyenko B.Y., Galata L.P. Design and research of mathematical model for information security system in computer network. *Науковий журнал «Наукоємні технології».* – 2017, № 2 (34), С. 114 - 118.
26. Корниенко Б.Я. Информационная безопасность и технологии компьютерных сетей : монография. ISBN 978-3-330-02028-3, LAMBERT Academic Publishing, Saarbrucken, Deutschland. – 2016. – 102 с.
27. Korniyenko B., Galata L., Kozuberda O. Modeling of security and risk assessment in information and communication system. *Sciences of Europe.* – 2016. – V. 2. – No 2 (2). – P. 61 -63.
28. Korniyenko B. The classification of information technologies and control systems. *International scientific journal.* – 2016. –№ 2. – P. 78 - 81.
29. Корнієнко Б.Я. Інформаційні технології оптимального управління виробництвом мінеральних добрив: монографія. – К.: Вид-во Аграр Медіа Груп, 2014. – 288 с.

30. Korniyenko B., Galata L., Ladieva L. Security Estimation of the Simulation Polygon for the Protection of Critical Information Resources / B. Korniyenko, //CEUR Workshop Proceedings, Selected Papers of the XVIII International Scientific and Practical Conference "Information Technologies and Security" (ITS 2018) Kyiv, Ukraine, November 27, 2018, Vol-2318, - P. 176-187, urn:nbn:de:0074-2318-4
31. Kornienko Y.M., Liubeka A.M., Sachok R.V., Korniyenko B.Y. Modeling of heat exchangement in fluidized bed with mechanical liquid distribution // ARPN Journal of Engineering and Applied Sciences. 2019. №14(12). P. 2203-2210.
32. Korniyenko B., Galata L., Ladieva L. Security Estimation of the Simulation Polygon for the Protection of Critical Information Resources // CEUR Workshop Proceedings, Selected Papers of the XVIII International Scientific and Practical Conference "Information Technologies and Security" (ITS 2018). Kyiv. Ukraine. 2018. №2318. P. 176-187.
33. Корнієнко Б.Я. Дослідження моделі взаємодії відкритих систем з погляду інформаційної безпеки // Наукоємні технології. 2012. №3(15). С. 83-89.
34. Korniyenko B., Yudin O., Novizkij E. Open systems interconnection model investigation from the viewpoint of information security // The Advanced Science Journal. 2013. №8. P. 53-56.
35. Zhulynskyi A.A., Ladieva L.R., Korniyenko B.Y. Parametric identification of the process of contact membrane distillation // ARPN Journal of Engineering and Applied Sciences Volume 14. 2019. №17. P. 3108-3112.
36. Korniyenko, B., Ladieva, L., Galata, L. Control system for the production of mineral fertilizers in a granulator with a fluidized bed. ATIT 2020 - Proceedings: 2020 2nd IEEE International Conference on Advanced Trends in Information Theory, 2020, pp. 307–310.
37. Корнієнко Б. Я., Галата Л. П. Дослідження імітаційного полігону захисту критичних інформаційних ресурсів методом IRISK. Моделювання та інформаційні технології. 2018. Вип. 83. С. 34–42.