

Тема 9. Основні фактори, що визначають економічні характеристики програмних систем

Зміст

1. Вступ
2. Основні економічні характеристики програмних систем
3. Фактори, що впливають на вартість виробництва програмних систем
 - 3.1. Розмір і складність програмної системи
 - 3.2. Вибір технологій і методів розробки
 - 3.3. Кваліфікація і досвід персоналу
 - 3.4. Терміни розробки
 - 3.5. Ризики проекту
4. Фактори, що впливають на ефективність використання програмних систем
 - 4.1. Якість програмної системи
 - 4.2. Відповідність програмної системи вимогам користувачів
 - 4.3. Зручність використання програмної системи
 - 4.4. Надійність і безпека програмної системи
5. Моделі прогнозування економічних характеристик програмних систем
6. Висновки
7. Список літератури

1. Вступ

Розробка програмного забезпечення є складним і багатогранним процесом, який вимагає значних інвестицій ресурсів, часу та людського капіталу. В умовах конкурентного ринку та обмежених ресурсів особливого значення набуває розуміння економічних аспектів створення та використання програмних систем.

Програмні системи мають унікальні економічні характеристики, які відрізняють їх від традиційних виробничих товарів. Нематеріальна природа програмного забезпечення, висока вартість початкового розвитку та низька вартість тиражування, залежність від людського фактора та швидкі темпи технологічних змін – все це робить економіку програмних систем особливою областю дослідження.

У цій лекції ми розглянемо основні економічні характеристики програмних систем, фактори, що впливають на вартість їх розробки та ефективність використання, а також моделі, які дозволяють прогнозувати економічні параметри програмних проектів. Розуміння цих факторів допоможе приймати обґрунтовані рішення щодо інвестицій у розробку програмного забезпечення, оптимізації процесів розробки та максимізації економічної цінності програмних продуктів.

2. Основні економічні характеристики програмних систем

Програмні системи мають ряд унікальних економічних характеристик, які визначають специфіку їх розробки, впровадження та використання:

2.1. Висока вартість розробки та низька вартість тиражування

Програмні системи вимагають значних початкових інвестицій для розробки, але після створення вони можуть бути скопійовані практично без додаткових витрат. Ця особливість створює економію масштабу, коли збільшення кількості користувачів призводить до зниження середньої вартості на одного користувача.

2.2. Нематеріальність та цифрова природа

На відміну від фізичних товарів, програмне забезпечення не має матеріальної форми. Це впливає на процеси виробництва, розповсюдження, захисту інтелектуальної власності та ціноутворення.

2.3. Висока додана вартість

Програмні системи зазвичай мають високу додану вартість, оскільки їх створення вимагає значного інтелектуального внеску та спеціалізованих навичок, але при цьому вартість матеріальних ресурсів для їх виробництва мінімальна.

2.4. Залежність від людського фактора

Розробка програмного забезпечення – це насамперед інтелектуальна діяльність, де людський фактор (знання, навички, досвід, креативність) має визначальний вплив на результат та економічні показники.

2.5. Мережеві ефекти

Багато програмних систем характеризуються мережевими ефектами, коли цінність продукту для кожного користувача зростає з збільшенням загальної кількості користувачів (наприклад, месенджери, соціальні мережі, операційні системи).

2.6. Технологічне старіння

Програмні системи схильні до швидкого технологічного старіння через постійний розвиток технологій, що вимагає регулярних інвестицій у підтримку та оновлення.

2.7. Високі транзакційні витрати при зміні постачальника

Після впровадження програмної системи зміна постачальника або перехід на іншу систему часто пов'язані з високими витратами на міграцію даних, навчання персоналу, адаптацію бізнес-процесів, що створює ефект "замикання" (lock-in).

2.8. Складність оцінки вартості та термінів розробки

Розробка програмного забезпечення характеризується високим рівнем невизначеності, що ускладнює точне прогнозування вартості та термінів, що, у свою чергу, створює додаткові економічні ризики.

3. Фактори, що впливають на вартість виробництва програмних систем

3.1. Розмір і складність програмної системи

Розмір і складність програмної системи є одними з найважливіших факторів, що визначають вартість її розробки. Ці фактори можна розглядати з різних аспектів:

3.1.1. Розмір системи

Розмір програмної системи традиційно вимірюється кількістю рядків коду (LOC - Lines of Code) або функціональних точок (FP - Function Points):

- **Рядки коду (LOC):** Простий, але залежний від мови програмування метрик. Дослідження показують, що вартість розробки зростає нелінійно з збільшенням кількості рядків коду, приблизно за законом $n^{1.5}$, де n - кількість рядків коду.
- **Функціональні точки (FP):** Більш універсальний метрик, що вимірює функціональність системи з точки зору користувача. Включає оцінку вхідних даних, вихідних даних, запитів, файлів і зовнішніх інтерфейсів.

3.1.2. Складність системи

Складність системи може бути розглянута в кількох вимірах:

- **Алгоритмічна складність:** Складність алгоритмів і обчислень, що використовуються в системі.
- **Структурна складність:** Складність архітектури системи, взаємозв'язків між компонентами.
- **Когнітивна складність:** Наскільки складно зрозуміти та підтримувати код.
- **Доменна складність:** Складність предметної області, яку автоматизує система.
- **Технологічна складність:** Кількість і складність використовуваних технологій, платформ, фреймворків.
- **Інтеграційна складність:** Кількість і складність інтеграцій з іншими системами.

3.1.3. Вплив розміру і складності на вартість

Вплив розміру і складності на вартість можна описати наступними закономірностями:

1. **Нелінійне зростання вартості:** Вартість розробки зростає нелінійно зі збільшенням розміру та складності системи.

2. **Ефект "комбінаторного вибуху":** При збільшенні кількості компонентів кількість потенційних взаємодій між ними зростає експоненціально, що підвищує складність тестування та інтеграції.
3. **Ускладнення комунікації:** У великих проектах зростає кількість комунікаційних каналів між членами команди, що може призводити до зниження ефективності і підвищення вартості.
4. **Зростання потреби в документації та формальних процесах:** Великі та складні системи вимагають більш детальної документації та формалізованих процесів, що також збільшує вартість.

3.1.4. Стратегії управління розміром і складністю

Для оптимізації вартості розробки застосовуються різні стратегії управління розміром і складністю:

1. **Модульна архітектура:** Розбиття системи на незалежні модулі з чітко визначеними інтерфейсами.
2. **Інкрементальна розробка:** Поетапна розробка системи з поступовим нарощуванням функціональності.
3. **Повторне використання компонентів:** Використання готових бібліотек, фреймворків та компонентів.
4. **Стандартизація:** Використання стандартних підходів, шаблонів проектування, архітектурних стилів.
5. **Автоматизація:** Використання засобів автоматизації для генерації коду, тестування, документування.

3.2. Вибір технологій і методів розробки

Вибір технологій і методів розробки суттєво впливає на вартість програмної системи на всіх етапах її життєвого циклу.

3.2.1. Вплив вибору технологій

Технології розробки програмного забезпечення включають мови програмування, фреймворки, бібліотеки, бази даних, інструменти розробки та середовища виконання. Їх вибір впливає на вартість наступним чином:

1. **Продуктивність розробки:** Різні технології забезпечують різну швидкість розробки. Наприклад, високорівневі мови програмування та фреймворки можуть значно підвищити продуктивність порівняно з низькорівневими мовами.
2. **Вартість інструментів:** Деякі технології вимагають дорогих ліцензійних інструментів розробки або середовищ виконання.
3. **Доступність кадрів:** Популярні технології мають більший пул доступних фахівців, що може знизити вартість залучення персоналу, але може підвищити витрати на заробітну плату через більшу конкуренцію.
4. **Крива навчання:** Складні або нові технології вимагають більше часу на освоєння, що може збільшити початкові витрати.
5. **Ефективність виконання:** Деякі технології забезпечують кращу продуктивність системи, що може знизити витрати на інфраструктуру при експлуатації.
6. **Масштабованість:** Технології з кращою підтримкою масштабування можуть знизити майбутні витрати на розширення системи.
7. **Довгострокова підтримка:** Технології з гарантованою довгостроковою підтримкою знижують ризики майбутніх витрат на міграцію.

3.2.2. Вплив методів розробки

Методи розробки включають методології, процеси, практики та підходи до організації розробки. Основні методології, що впливають на вартість:

1. **Водоспадна модель (Waterfall):**
 - Детальне планування на початку проекту
 - Послідовне виконання етапів

- Нижча гнучкість при змінах вимог
- Може призводити до перевитрат при неточних початкових оцінках або змінах вимог

2. Гнучкі методології (Agile):

- Ітеративна та інкрементальна розробка
- Активне залучення замовника
- Швидка адаптація до змін
- Може знизити ризики за рахунок раннього виявлення проблем
- Вимагає більше зусиль на комунікацію та координацію

3. DevOps:

- Інтеграція розробки та експлуатації
- Автоматизація процесів доставки та розгортання
- Початкові інвестиції в автоматизацію можуть бути значними, але знижують довгострокові витрати
- Прискорює цикл зворотного зв'язку

4. Бережлива розробка (Lean Development):

- Фокус на мінімізації втрат та максимізації цінності
- Оптимізація потоку створення цінності
- Може знизити вартість за рахунок усунення непродуктивних активностей

3.2.3. Критерії вибору технологій і методів з економічної точки зору

При виборі технологій і методів розробки з економічної точки зору слід враховувати:

1. **Сукупну вартість володіння (TCO):** Включає не тільки початкові витрати на придбання та впровадження, але й витрати на підтримку, обслуговування, навчання, масштабування.
2. **Відповідність проекту:** Технології та методи повинні відповідати характеру проекту, його масштабу, вимогам до якості, обмеженням.
3. **Ризики:** Оцінка ризиків, пов'язаних з використанням конкретних технологій і методів.
4. **Довгострокова перспектива:** Оцінка життєздатності технологій у довгостроковій перспективі.
5. **Екосистема підтримки:** Наявність спільноти, документації, інструментів, сервісів підтримки.

3.3. Кваліфікація і досвід персоналу

Кваліфікація і досвід персоналу є одними з найважливіших факторів, що впливають на вартість та якість розробки програмних систем.

3.3.1. Вплив кваліфікації персоналу на продуктивність

Дослідження показують, що різниця в продуктивності між розробниками може складати до 10 разів і більше. Ця різниця пояснюється:

1. **Технічними навичками:** Знання мов програмування, фреймворків, інструментів.
2. **Досвідом у предметній області:** Розуміння бізнес-процесів та вимог галузі.
3. **Аналітичними здібностями:** Вміння аналізувати проблеми та знаходити ефективні рішення.
4. **Навичками проектування:** Здатність створювати ефективну архітектуру та дизайн системи.
5. **Soft skills:** Комунікативні навички, здатність до роботи в команді, самоорганізація.

3.3.2. Економічні аспекти залучення висококваліфікованого персоналу

1. **Вища вартість залучення:** Висококваліфіковані фахівці коштують дорожче, їх залучення може вимагати додаткових витрат на рекрутинг, релокацію тощо.
2. **Зниження загальної вартості проекту:** Незважаючи на вищу вартість, залучення висококваліфікованих фахівців може знизити загальну вартість проекту за рахунок:
 - Вищої продуктивності

- Меншої кількості помилок
 - Кращих архітектурних рішень, що знижують майбутні витрати на підтримку
 - Швидшого вирішення проблем
3. **Ефект "10х-інженер"**: Концепція, згідно з якою найкращі розробники можуть бути до 10 разів продуктивнішими за середніх, що робить економічно виправданим їхнє залучення навіть за значно вищу заробітну плату.

3.3.3. Структура команди та її вплив на вартість

1. **Розмір команди**: Згідно із законом Брукса, «додавання людей до запізненого проекту затримує його ще більше» через збільшення комунікаційних витрат. Оптимальний розмір команди залежить від характеру проекту.
2. **Баланс навичок**: Оптимальне співвідношення старших та молодших фахівців, спеціалістів різних профілів.
3. **Розподіл ролей**: Чіткий розподіл ролей та відповідальності може підвищити ефективність та знизити вартість.
4. **Стабільність команди**: Висока плинність кадрів збільшує витрати на рекрутинг, адаптацію, передачу знань.

3.3.4. Стратегії розвитку та утримання персоналу

1. **Інвестиції в навчання та розвиток**: Можуть підвищити продуктивність існуючої команди та знизити плинність кадрів.
2. **Програми менторства**: Дозволяють ефективно передавати знання від досвідчених фахівців до початківців.
3. **Створення команд із змішаним рівнем досвіду**: Баланс між досвідченими та початкуючими розробниками може оптимізувати витрати при збереженні якості.
4. **Програми утримання талантів**: Можуть знизити витрати, пов'язані з плинністю кадрів.

3.4. Терміни розробки

Терміни розробки мають значний вплив на вартість програмних систем. Цей вплив проявляється в різних аспектах та може бути як позитивним, так і негативним.

3.4.1. Закон Брукса та його економічні наслідки

Закон Брукса стверджує, що «додавання людських ресурсів до запізненого проекту робить його ще більш запізненим». Економічні наслідки цього закону:

1. **Нелінійне зростання комунікаційних витрат**: При збільшенні команди кількість потенційних комунікаційних каналів зростає за формулою $n(n-1)/2$, де n - кількість членів команди.
2. **Витрати на введення нових учасників у курс справ**: Нові учасники вимагають часу на ознайомлення з проектом, що тимчасово знижує продуктивність існуючих учасників.
3. **Розподіл задач**: Не всі задачі можуть бути розпаралелені, що обмежує ефективність збільшення команди.

3.4.2. Залежність між термінами, вартістю та якістю

У проектній практиці існує концепція «трикутника обмежень» або «залізного трикутника», що включає терміни, вартість та якість (або обсяг робіт). Зміна одного з цих параметрів впливає на інші:

1. **Скорочення термінів при фіксованій якості**: Зазвичай призводить до збільшення вартості через:
 - Необхідність залучення додаткових ресурсів
 - Оплату понаднормової роботи
 - Використання більш продуктивних, але дорожчих технологій
 - Паралельне виконання робіт, що може підвищити ризики
2. **Збільшення термінів**: Може мати різний економічний ефект:
 - Зниження піків навантаження на ресурси, що може знизити вартість

- Можливість більш ретельного планування та оптимізації
- Але також може призвести до втрат через пізніший вихід на ринок, упущену вигоду

3.4.3. Оптимальні терміни розробки

1. **Концепція "найменшої сукупної вартості":** Існує точка, в якій сума витрат на розробку та втрат від затримки виходу на ринок мінімальна.
2. **Економічні переваги ітеративної розробки:** Ітеративний підхід дозволяє:
 - Отримувати цінність від системи раніше через поетапне впровадження
 - Знижувати ризики через ранній зворотний зв'язок
 - Оптимізувати розподіл ресурсів
3. **Стратегія "Time-to-Market":** Для деяких систем швидкий вихід на ринок є критичним фактором економічного успіху, що виправдовує додаткові витрати на прискорення розробки.

3.4.4. Методи оптимізації термінів без значного збільшення вартості

1. **Визначення мінімально життєздатного продукту (MVP):** Фокус на найбільш важливих функціях, що дозволяє швидше випустити першу версію.
2. **Паралельна розробка незалежних компонентів:** Оптимізація розподілу завдань для максимального паралелізму.
3. **Автоматизація рутинних процесів:** Інвестиції в автоматизацію тестування, збірки, розгортання.
4. **Використання готових компонентів:** Повторне використання існуючого коду, бібліотек, фреймворків.
5. **Безперервна інтеграція та розгортання (CI/CD):** Скорочення часу на інтеграцію та виявлення проблем.
6. **Гнучке управління обсягом робіт:** Пріоритизація функцій та готовність відкласти менш важливі функції.

3.5. Ризики проекту

Ризики проекту мають значний вплив на економічні характеристики програмних систем, оскільки вони можуть призвести до додаткових витрат, затримок та проблем з якістю.

3.5.1. Типи ризиків та їх економічний вплив

1. **Технічні ризики:**
 - Складність або новизна технологій
 - Технічні обмеження або несумісності
 - Проблеми з продуктивністю або масштабованістю
 - **Економічний вплив:** Додаткові витрати на дослідження, експерименти, переробку, оптимізацію.
2. **Ризики вимог:**
 - Неповні або неоднозначні вимоги
 - Зміни вимог у процесі розробки
 - Конфлікт інтересів зацікавлених сторін
 - **Економічний вплив:** Витрати на переробку, додаткові функції, невикористані функції.
3. **Організаційні ризики:**
 - Недостатня підтримка з боку керівництва
 - Зміни в пріоритетах або стратегії
 - Конфлікти між підрозділами
 - **Економічний вплив:** Зниження ефективності використання ресурсів, затримки в прийнятті рішень.
4. **Ресурсні ризики:**
 - Недостатність або недоступність необхідних ресурсів
 - Плинність кадрів

- Залежність від ключових осіб
- **Економічний вплив:** Додаткові витрати на залучення ресурсів, навчання, передачу знань.

5. Зовнішні ризики:

- Зміни в законодавстві або стандартах
- Дії конкурентів
- Економічні зміни
- **Економічний вплив:** Необхідність адаптації системи, зміна бізнес-моделі.

3.5.2. Методи оцінки ризиків

1. Якісний аналіз ризиків:

- Визначення ймовірності та впливу ризиків
- Ранжування ризиків за пріоритетом
- **Економічна цінність:** Дозволяє зосередитися на найбільш критичних ризиках.

2. Кількісний аналіз ризиків:

- Оцінка очікуваного грошового значення ризику (EVM - Expected Monetary Value)
- Моделювання Монте-Карло для оцінки ймовірнісних сценаріїв
- Аналіз чутливості для визначення ключових факторів ризику
- **Економічна цінність:** Дозволяє визначити необхідний розмір резервів та рентабельність заходів з управління ризиками.

3.5.3. Стратегії управління ризиками та їх економічне обґрунтування

1. Уникнення ризику:

- Зміна плану проекту для усунення ризику
- Відмова від ризикованих функцій або підходів
- **Економічне обґрунтування:** Порівняння потенційних втрат від ризику з вартістю альтернативних підходів.

2. Передача ризику:

- Аутсорсинг ризикованих компонентів
- Придбання страхування
- Контрактні гарантії
- **Економічне обґрунтування:** Порівняння вартості передачі з очікуваними втратами.

3. Зменшення ризику:

- Додаткове тестування або прототипування
- Залучення експертів
- Використання перевірених технологій
- **Економічне обґрунтування:** Оцінка зниження ймовірності або впливу ризику відносно інвестицій у заходи зменшення.

4. Прийняття ризику:

- Активне: Створення резервів часу або бюджету
- Пасивне: Прийняття наслідків ризику без заходів
- **Економічне обґрунтування:** Оцінка вартості резервів відносно потенційних втрат.

3.5.4. Економічна цінність управління ризиками

1. **Зниження непередбачених витрат:** Ефективне управління ризиками знижує ймовірність та вплив непередбачених подій.
2. **Оптимізація резервів:** Точна оцінка ризиків дозволяє встановити оптимальний розмір резервів (ні надто великий, ні надто малий).
3. **Підвищення прогнозованості:** Зниження варіативності кінцевих результатів проекту.

4. **Підвищення довіри стейкхолдерів:** Демонстрація систематичного підходу до управління ризиками підвищує довіру інвесторів та замовників.
5. **Формування бази знань:** Документування ризиків та стратегій їх подолання створює цінний актив для майбутніх проектів.

4. Фактори, що впливають на ефективність використання програмних систем

4.1. Якість програмної системи

Якість програмного забезпечення є одним із ключових факторів, що визначають економічну ефективність його використання. Висока якість дозволяє отримати максимальну віддачу від інвестицій у розробку та впровадження програмної системи.

4.1.1. Економічні аспекти якості програмного забезпечення

1. **Співвідношення витрат та якості:**
 - Закон "10:1": Виправлення помилки на етапі експлуатації коштує в 10 разів дорожче, ніж на етапі розробки.
 - Прогресуюче зростання вартості виправлення дефектів на пізніх етапах життєвого циклу.
2. **Додаткові витрати при низькій якості:**
 - Простої в роботі користувачів
 - Витрати на підтримку та виправлення помилок
 - Втрата репутації та довіри користувачів
 - Можливі юридичні наслідки та фінансові збитки
3. **Інвестиції в якість:**
 - Превентивні заходи (навчання, стандарти кодування, архітектурні рішення)
 - Виявлення дефектів (рев'ю коду, тестування, статичний аналіз)
 - Баланс між інвестиціями в якість та швидкістю виведення продукту на ринок

4.1.2. Стандарти якості програмного забезпечення

1. **ISO/IEC 25010:2011** (раніше ISO 9126) визначає модель якості програмного продукту, що включає:
 - Функціональна придатність
 - Ефективність виконання
 - Сумісність
 - Зручність використання
 - Надійність
 - Захищеність
 - Супроводжуваність
 - Переносимість
2. **Економічна цінність відповідності стандартам:**
 - Зниження ризиків для замовників та користувачів
 - Можливість сертифікації для певних ринків та галузей
 - Підвищення довіри користувачів та ринкової вартості продукту

4.1.3. Метрики якості та їх економічна інтерпретація

1. **Метрики дефектів:**
 - Щільність дефектів (кількість дефектів на одиницю розміру коду)
 - Коефіцієнт дефектності (відсоток дефектних компонентів)
 - Середній час між відмовами (MTBF)
 - **Економічна інтерпретація:** Прогнозування майбутніх витрат на підтримку та виправлення.
2. **Метрики структурної якості:**
 - Цикломатична складність (складність потоку керування)
 - Зв'язаність та зчеплення модулів
 - Дублювання коду

- **Економічна інтерпретація:** Прогнозування витрат на модифікацію та розширення системи.

3. Метрики процесу:

- Ефективність виявлення дефектів
- Ефективність виправлення дефектів
- Швидкість регресії
- **Економічна інтерпретація:** Оцінка ефективності процесів забезпечення якості.

4.1.4. Економічно ефективні підходи до забезпечення якості

1. Зміщення акценту на ранні етапи:

- Більше уваги до аналізу вимог та проектування
- Рання верифікація та валідація
- Економічний ефект: Зниження вартості виправлення дефектів

2. Автоматизація контролю якості:

- Автоматизоване тестування
- Інструменти статичного аналізу коду
- Безперервна інтеграція (CI)
- Економічний ефект: Зниження трудомісткості та підвищення покриття контролю якості

3. Фокус на критичних компонентах:

- Ідентифікація компонентів з високим ризиком або впливом
- Концентрація зусиль з забезпечення якості на цих компонентах
- Економічний ефект: Оптимізація співвідношення витрат та результатів

4. Впровадження культури якості:

- Навчання та підвищення кваліфікації розробників
- Заохочення відповідальності за якість на всіх рівнях
- Економічний ефект: Зниження кількості дефектів у довгостроковій перспективі

4.2. Відповідність програмної системи вимогам користувачів

Відповідність програмної системи вимогам користувачів є ключовим фактором, що визначає її економічну цінність та рентабельність інвестицій. Система, яка точно відповідає потребам користувачів, забезпечує максимальну віддачу від інвестицій та мінімізує витрати на доопрацювання та підтримку.

4.2.1. Економічні наслідки невідповідності вимогам

1. Прямі фінансові втрати:

- Витрати на переробку та доопрацювання системи
- Втрати від невикористовуваних функцій ("золочення")
- Зниження продуктивності користувачів через незручні або неоптимальні процеси

2. Непрямі фінансові втрати:

- Затримка або неможливість отримання очікуваних бізнес-переваг
- Зниження задоволеності користувачів та клієнтів
- Потенційна втрата ринкових можливостей

3. Ризики невдачі проекту:

- Відмова від використання системи користувачами
- Скасування або призупинення проекту
- Втрата інвестицій

4.2.2. Економічна цінність точного виявлення вимог

1. Зниження витрат на зміни:

- Чим раніше виявлені справжні потреби, тим дешевше їх задовольнити

- Правило "1:10:100" - вартість виправлення помилки у вимогах зростає на порядок на кожному наступному етапі (аналіз:розробка:експлуатація)
- 2. **Оптимізація обсягу розробки:**
 - Фокус на найбільш цінних функціях (принцип Парето: 20% функцій забезпечують 80% цінності)
 - Уникнення "золочення" (надмірної функціональності)
 - Пріоритизація функціональності за співвідношенням цінність/вартість
- 3. **Підвищення прийняття системи користувачами:**
 - Зниження опору змінам
 - Підвищення задоволеності та продуктивності
 - Швидше досягнення бізнес-цілей впровадження
- 4.2.3. **Методи забезпечення відповідності вимогам та їх економічна ефективність**
 1. **Залучення користувачів:**
 - Участь користувачів у визначенні та валідації вимог
 - Раннє та часте отримання зворотного зв'язку
 - **Економічна ефективність:** Інвестиції в залучення користувачів багатократно окупаються за рахунок зниження витрат на переробку.
 2. **Ітеративна розробка:**
 - Розробка системи невеликими інкрементами
 - Регулярна демонстрація та валідація
 - **Економічна ефективність:** Раннє виявлення відхилень від потреб, можливість коригування курсу з мінімальними втратами.
 3. **Прототипування:**
 - Створення прототипів для валідації концепцій та інтерфейсів
 - Еволюційне або одноразове прототипування
 - **Економічна ефективність:** Інвестиції в прототипування знижують ризики дорогих помилок у дизайні.
 4. **Гнучке управління вимогами:**
 - Баланс між стабільністю та адаптивністю
 - Формалізований процес зміни вимог
 - **Економічна ефективність:** Оптимізація співвідношення між задоволенням нових потреб та контролем витрат.
- 4.2.4. **Оцінка економічного ефекту від відповідності вимогам**
 1. **Кількісні показники:**
 - Кількість запитів на зміни після впровадження
 - Ступінь використання функціональності
 - Продуктивність користувачів до та після впровадження
 - ROI від впровадження системи
 2. **Якісні показники:**
 - Задоволеність користувачів
 - Відповідність бізнес-цілям
 - Готовність рекомендувати систему іншим (NPS - Net Promoter Score)
 - Вплив на конкурентоспроможність організації
 3. **Методи оцінки:**
 - Порівняння з базовим рівнем (до впровадження)
 - A/B тестування різних версій або функцій
 - Аналіз тенденцій використання та продуктивності
 - Опитування та інтерв'ю з користувачами

4.3. Зручність використання програмної системи

Зручність використання (юзабіліті) програмної системи є критичним фактором, що впливає на її економічну ефективність. Системи з високим рівнем юзабіліті забезпечують

вищу продуктивність користувачів, нижчі витрати на навчання та підтримку, підвищують задоволеність користувачів.

4.3.1. Економічний вплив зручності використання

1. Підвищення продуктивності користувачів:

- Зниження часу на виконання завдань
- Зменшення кількості помилок користувачів
- Прискорення освоєння системи
- **Економічний ефект:** До 50% підвищення продуктивності при використанні систем з хорошим юзабіліті.

2. Зниження витрат на навчання та підтримку:

- Менша потреба в формальному навчанні
- Скорочення часу на освоєння системи
- Зменшення кількості звернень до служби підтримки
- **Економічний ефект:** Зниження витрат на навчання та підтримку на 20-40%.

3. Підвищення прийняття та використання системи:

- Зниження опору до використання нової системи
- Підвищення частоти і глибини використання функціональності
- Зменшення тенденції до повернення до старих методів роботи
- **Економічний ефект:** Максимізація віддачі від інвестицій у розробку та впровадження.

4. Підвищення задоволеності користувачів:

- Покращення сприйняття бренду та організації
- Підвищення лояльності користувачів
- Збільшення NPS (Net Promoter Score)
- **Економічний ефект:** Зниження відтоку клієнтів, підвищення довгострокової цінності клієнта (LTV).

4.3.2. Компоненти зручності використання та їх економічна цінність

1. Ефективність:

- Мінімізація кроків для виконання завдань
- Оптимізація потоків роботи
- **Економічна цінність:** Пряма економія робочого часу користувачів.

2. Легкість навчання:

- Інтуїтивно зрозумілий інтерфейс
- Логічна організація функцій
- Доступні та зрозумілі підказки
- **Економічна цінність:** Зниження витрат на навчання, швидше досягнення продуктивності.

3. Запам'ятовуваність:

- Можливість легко згадати, як використовувати систему після перерви
- Послідовність та стандартизація інтерфейсу
- **Економічна цінність:** Мінімізація втрат продуктивності при поверненні до системи.

4. Запобігання помилкам:

- Дизайн, що мінімізує можливість помилок
- Ефективний зворотний зв'язок
- Підтвердження критичних дій
- **Економічна цінність:** Зниження витрат на виправлення помилок та пов'язаних бізнес-втрат.

5. Суб'єктивне задоволення:

- Естетично приємний дизайн
- Позитивний емоційний досвід

- **Економічна цінність:** Підвищення мотивації використання, лояльності та задоволеності.

4.3.3. Методи оцінки та покращення зручності використання

1. Юзабіліті-тестування:

- Спостереження за реальними користувачами
- Аналіз часу виконання завдань та помилок
- Збір якісного зворотного зв'язку
- **Співвідношення витрат та вигод:** Інвестиції в юзабіліті-тестування окупаються в 10-100 разів.

2. Експертна оцінка:

- Евристичний аналіз
- Когнітивний аналіз
- Аналіз відповідності стандартам
- **Співвідношення витрат та вигод:** Ефективний метод виявлення основних проблем з мінімальними витратами.

3. Аналітика користувачів:

- Збір та аналіз даних про взаємодію користувачів з системою
- A/B тестування альтернативних версій інтерфейсу
- **Співвідношення витрат та вигод:** Дозволяє оптимізувати найбільш критичні аспекти взаємодії з мінімальними витратами.

4.3.4. Інвестиції в зручність використання та їх рентабельність

1. Оптимальний рівень інвестицій:

- Закон спадної віддачі: після певного рівня додаткові інвестиції дають все менший ефект
- Фокус на критичних функціях та сценаріях використання
- Баланс між ідеальним дизайном та практичними обмеженнями

2. Розрахунок ROI від покращення юзабіліті:

- Оцінка вартості часу користувачів
- Оцінка втрат від помилок користувачів
- Оцінка витрат на навчання та підтримку
- Порівняння з витратами на покращення юзабіліті

3. Стратегічний підхід до інвестицій в юзабіліті:

- Встановлення метрик та цілей
- Регулярне вимірювання та аналіз
- Безперервне вдосконалення на основі даних
- Інтеграція зручності використання в процес розробки з самого початку

4.4. Надійність і безпека програмної системи

Надійність і безпека програмної системи є критичними факторами, що визначають її економічну ефективність та сукупну вартість володіння. Недостатня увага до цих аспектів може призвести до значних фінансових втрат, репутаційних ризиків та юридичних наслідків.

4.4.1. Економічний вплив надійності програмного забезпечення

1. Витрати, пов'язані з відмовами системи:

- Прямі втрати від простоїв (втрачений прибуток, неможливість надання послуг)
- Витрати на відновлення роботи системи
- Витрати на компенсацію збитків клієнтам або партнерам
- **Масштаб впливу:** Для критичних бізнес-систем вартість простою може складати від тисяч до мільйонів доларів на годину.

2. Втрати продуктивності користувачів:

- Час, витрачений на очікування відновлення системи

- Необхідність повторного введення даних або виконання операцій
- Психологічний дискомфорт та фрустрація
- **Масштаб впливу:** Множення часу простою на кількість користувачів та їх ставку оплати праці.

3. Репутаційні втрати:

- Зниження довіри клієнтів та партнерів
- Негативні відгуки та публічна критика
- Погіршення сприйняття бренду
- **Масштаб впливу:** Довгострокові наслідки для бізнесу, які важко точно кількісно оцінити.

4.4.2. Економічний вплив безпеки програмного забезпечення

1. Витрати, пов'язані з інцидентами безпеки:

- Прямі фінансові втрати від шахрайства, крадіжки або розкриття інформації
- Витрати на розслідування та усунення наслідків інциденту
- Штрафи та судові витрати за порушення законодавства про захист даних
- **Масштаб впливу:** Середня вартість витоку даних становить мільйони доларів для середніх та великих організацій.

2. Непрямі втрати від порушень безпеки:

- Втрата інтелектуальної власності
- Розкриття комерційної таємниці
- Витрати на відновлення репутації
- Втрата конкурентної переваги
- **Масштаб впливу:** Може бути катастрофічним для бізнесу, особливо для інноваційних компаній.

3. Відповідність регуляторним вимогам:

- Витрати на забезпечення відповідності (GDPR, PCI DSS, HIPAA, тощо)
- Штрафи за невідповідність
- Обмеження на ведення бізнесу в певних регіонах
- **Масштаб впливу:** Штрафи можуть складати до 4% глобального річного обороту (GDPR).

4.4.3. Інвестиції в надійність та безпеку та їх рентабельність

1. Стратегії інвестування в надійність:

- Архітектурні рішення (резервування, відмовостійкість, гаряча заміна)
- Моніторинг та попередження проблем
- Автоматизоване тестування та забезпечення якості
- Плани відновлення після збоїв

2. Стратегії інвестування в безпеку:

- Безпека на етапі проектування (Security by Design)
- Регулярний аудит та тестування на проникнення
- Програми підвищення обізнаності персоналу
- Моніторинг та реагування на загрози

3. Оцінка рентабельності інвестицій в надійність та безпеку:

- Аналіз ризиків: ймовірність інциденту × потенційний збиток
- Порівняння з вартістю превентивних заходів
- Врахування непрямих та довгострокових переваг
- **Типові показники:** Інвестиції в превентивні заходи безпеки окупаються в 2-10 разів порівняно з витратами на усунення наслідків.

4.4.4. Баланс між надійністю, безпекою та іншими якостями системи

1. Компроміси між надійністю та іншими якостями:

- Надійність vs. продуктивність
- Надійність vs. швидкість виходу на ринок

- Надійність vs. вартість розробки
 - **Економічний аспект:** Визначення оптимального рівня надійності, коли додаткові інвестиції перестають давати пропорційну віддачу.
- 2. Компроміси між безпекою та іншими якостями:**
- Безпека vs. зручність використання
 - Безпека vs. швидкодія
 - Безпека vs. гнучкість та відкритість
 - **Економічний аспект:** Пошук балансу, який забезпечує достатній рівень захисту без надмірного обмеження функціональності.
- 3. Диференційований підхід до інвестицій:**
- Класифікація компонентів системи за критичністю
 - Концентрація ресурсів на найбільш критичних компонентах
 - Ризик-орієнтований підхід до визначення необхідного рівня захисту
 - **Економічний аспект:** Оптимізація розподілу обмежених ресурсів для максимізації загальної надійності та безпеки системи.

4.4.5. Кількісна оцінка якості надійності та безпеки

- 1. Метрики надійності:**
- Середній час між відмовами (MTBF)
 - Середній час відновлення (MTTR)
 - Доступність системи (% робочого часу)
 - Частота відмов на певний обсяг операцій
- 2. Метрики безпеки:**
- Кількість виявлених вразливостей
 - Середній час усунення вразливостей
 - Кількість успішних/невдалих спроб несанкціонованого доступу
 - Рівень відповідності стандартам безпеки
- 3. Економічна інтерпретація метрик:**
- Переведення часу простою в фінансові втрати
 - Оцінка потенційних втрат від різних типів порушень безпеки
 - Розрахунок економічного ефекту від покращення показників
 - Визначення оптимальних цільових значень з економічної точки зору

5. Моделі прогнозування економічних характеристик програмних систем

Прогнозування економічних характеристик програмних систем є важливим аспектом планування та управління проектами розробки програмного забезпечення. Використання відповідних моделей дозволяє приймати обґрунтовані рішення щодо інвестицій, ресурсів та термінів розробки.

5.1. Моделі оцінки вартості програмного забезпечення

5.1.1. Модель COSOMO (CONstructive COst MOdel)

- 1. COSOMO I (Базова модель):**
- Розроблена Баррі Боемом у 1981 році
 - Формула: $\text{Effort} = a \times (\text{KLOC})^b$, де KLOC - тисячі рядків коду, а a і b - константи, що залежать від типу проекту
 - Три режими розробки: органічний, напівінтегрований та вбудований
 - **Економічна значимість:** Проста модель для початкової оцінки вартості проекту
- 2. COSOMO II (Розширена модель):**
- Оновлена версія, що враховує сучасні підходи до розробки
 - Включає фактори масштабу та зусиль
 - Формула: $\text{Effort} = A \times \text{Size}^E \times \prod \text{EM}_i$, де A - константа, Size - розмір у тисячах рядків коду або функціональних точках, E - показник масштабу, EM_i - множники зусиль

- **Економічна значимість:** Більш точна модель, що враховує більше факторів впливу

5.1.2. Модель функціональних точок (Function Point Analysis)

1. Опис методу:

- Розроблений Аланом Альбрехтом у IBM
- Вимірює функціональність з точки зору користувача
- Компоненти: зовнішні входи/виходи, зовнішні запити, внутрішні логічні файли, зовнішні інтерфейсні файли
- **Економічна значимість:** Дозволяє оцінювати вартість до початку розробки, незалежно від технології реалізації

2. Процес оцінки:

- Ідентифікація та підрахунок компонентів
- Класифікація складності кожного компонента
- Розрахунок невідкоригованих функціональних точок
- Оцінка технічної складності
- Розрахунок відкоригованих функціональних точок
- Конвертація в людино-години або вартість

3. Переваги для економічного аналізу:

- Незалежність від мови програмування
- Можливість використання на ранніх етапах проекту
- Наявність історичних даних для порівняння
- Фокус на цінності з точки зору бізнесу

5.1.3. Модель SLIM (Software Lifecycle Management)

1. Основні принципи:

- Розроблена Ларрі Патнемом
- Базується на законі Райлі-Нордена щодо зусиль та термінів
- Враховує складність програмного забезпечення, обмеження часу, технологічний індекс

2. Формула Патнема:

- $E = (LOC \times V)^{1/3} \times t^{4/3}$, де E - зусилля, LOC - рядки коду, V - спеціальний скейлінг-фактор, t - час розробки
- **Економічна значимість:** Демонструє фундаментальний компроміс між зусиллями та часом розробки

3. Використання в економічному аналізі:

- Оптимізація співвідношення між зусиллями та термінами
- Визначення точки мінімальної сукупної вартості
- Аналіз альтернативних сценаріїв розробки

5.1.4. Методи оцінки, засновані на експертних судженнях

1. Метод Дельфі:

- Анонімне опитування експертів
- Ітеративний процес з зворотним зв'язком
- **Економічна значимість:** Ефективний для проектів з високою невизначеністю

2. Planning Poker:

- Колективна оцінка складності задач
- Використання карт з числами Фібоначчі для вибору оцінок
- Обговорення розбіжностей та досягнення консенсусу
- **Економічна значимість:** Поєднує експертну оцінку з колективною мудрістю

3. Аналогове оцінювання:

- Порівняння з подібними завершеними проектами
- Коригування з урахуванням відмінностей

- **Економічна значимість:** Використання реальних історичних даних для підвищення точності оцінок

5.2. Моделі оцінки ефективності програмного забезпечення

5.2.1. Аналіз витрат і вигод (Cost-Benefit Analysis)

1. Методологія:

- Ідентифікація всіх витрат (прямих та непрямих)
- Ідентифікація всіх вигод (прямих та непрямих)
- Кількісна оцінка витрат та вигод у грошовому вираженні
- Дисконтування майбутніх потоків до поточної вартості
- Розрахунок чистої приведеної вартості (NPV)

2. Складнощі при застосуванні до програмного забезпечення:

- Нематеріальність багатьох вигод
- Невизначеність майбутніх вигод та витрат
- Складність оцінки непрямих ефектів
- **Економічна значимість:** Забезпечує комплексну оцінку доцільності інвестицій

5.2.2. Розрахунок повернення інвестицій (ROI)

1. Базова формула:

- $ROI = (Вигоди - Витрати) / Витрати \times 100\%$
- **Економічна значимість:** Простий та зрозумілий показник ефективності інвестицій

2. Методи оцінки вигод від програмного забезпечення:

- Підвищення продуктивності праці
- Скорочення операційних витрат
- Підвищення доходів або прибутку
- Зниження ризиків та уникнення втрат
- Покращення якості продуктів або послуг

3. Особливості ROI для програмного забезпечення:

- Необхідність врахування загальної вартості володіння (TCO)
- Важливість часового горизонту оцінки
- Необхідність врахування ризиків та невизначеності

5.2.3. Сукупна вартість володіння (Total Cost of Ownership, TCO)

1. Компоненти TCO для програмного забезпечення:

- Початкові витрати на розробку або придбання
- Витрати на впровадження та міграцію
- Витрати на обладнання та інфраструктуру
- Витрати на ліцензії та підписки
- Витрати на інтеграцію з іншими системами
- Витрати на навчання користувачів
- Витрати на підтримку та обслуговування
- Витрати на оновлення та модернізацію
- **Економічна значимість:** Забезпечує повну картину витрат протягом життєвого циклу

2. Методи розрахунку TCO:

- Облік усіх прямих та непрямих витрат
- Розподіл витрат за часовими періодами
- Дисконтування майбутніх витрат
- Порівняння альтернативних варіантів

5.2.4. Реальні опціони (Real Options)

1. Концепція реальних опціонів:

- Адаптація фінансової теорії опціонів до інвестиційних рішень

- Врахування цінності гнучкості та можливості прийняття рішень у майбутньому
 - **Економічна значимість:** Дозволяє кількісно оцінити стратегічну цінність проектів
- 2. Типи реальних опціонів у програмних проектах:**
- Опціон на розширення (можливість масштабування системи)
 - Опціон на відмову (можливість припинення проекту)
 - Опціон на відстрочку (можливість відкласти рішення)
 - Опціон на перемикання (можливість зміни технології)
- 3. Методи оцінки реальних опціонів:**
- Біноміальна модель
 - Модель Блека-Шоулза
 - Моделювання Монте-Карло
 - **Економічна значимість:** Більш точна оцінка проектів з високою невизначеністю або стратегічною цінністю

5.3. Моделі оцінки ризиків та їх економічних наслідків

5.3.1. Кількісний аналіз ризиків

- 1. Очікувана монетарна вартість (Expected Monetary Value, EMV):**
- $EMV = \text{Ймовірність виникнення ризику} \times \text{Вплив ризику у грошовому вираженні}$
 - **Економічна значимість:** Дозволяє порівнювати ризики та приймати рішення щодо їх пріоритизації
- 2. Моделювання Монте-Карло:**
- Симуляція різних сценаріїв з випадковими значеннями параметрів
 - Аналіз розподілу можливих результатів
 - **Економічна значимість:** Забезпечує розуміння діапазону можливих результатів та їх ймовірностей
- 3. Аналіз дерева рішень:**
- Графічне представлення послідовності рішень та їх можливих наслідків
 - Розрахунок очікуваної вартості кожного шляху
 - **Економічна значимість:** Допомогає обрати оптимальну стратегію в умовах невизначеності

5.3.2. Оцінка вартості управління ризиками

- 1. Витрати на зменшення ризиків:**
- Превентивні заходи
 - Пом'якшення наслідків
 - Передача ризиків (страхування)
- 2. Економічна доцільність управління ризиками:**
- Порівняння вартості заходів з очікуваним зниженням ризику
 - Оптимізація портфеля заходів управління ризиками
 - **Економічна значимість:** Визначення оптимального рівня інвестицій в управління ризиками

5.3.3. Моделі аналізу сценаріїв

- 1. Трьох-сценарний аналіз:**
- Оптимістичний, песимістичний та найбільш ймовірний сценарії
 - Оцінка результатів для кожного сценарію
 - **Економічна значимість:** Проста модель для оцінки впливу невизначеності
- 2. Аналіз чутливості:**
- Визначення впливу змін окремих параметрів на результати
 - Ідентифікація найбільш критичних факторів

- **Економічна значимість:** Фокусування уваги на найбільш важливих факторах ризику

5.4. Інтегровані моделі для комплексної оцінки

5.4.1. Збалансована система показників (Balanced Scorecard)

1. Адаптація для оцінки програмних систем:

- Фінансова перспектива (ROI, TCO, економія витрат)
- Клієнтська перспектива (задоволеність користувачів, відповідність вимогам)
- Перспектива внутрішніх процесів (ефективність розробки, якість)
- Перспектива інновацій та навчання (розвиток компетенцій, технологічне лідерство)
- **Економічна значимість:** Комплексна оцінка цінності програмної системи в різних аспектах

2. Застосування в управлінні портфелем проектів:

- Балансування інвестицій між різними типами проектів
- Відстеження прогресу та результатів на стратегічному рівні
- **Економічна значимість:** Узгодження інвестицій у IT зі стратегічними цілями організації

5.4.2. Інформаційна економіка (Information Economics)

1. Методологія Паркера та Бенсона:

- Оцінка бізнес-домену (узгодження зі стратегією, бізнес-ризиків, можливості зростання)
- Оцінка технологічного домену (архітектурна відповідність, технічні ризики, стратегічні технології)
- Зважування та ранжування проектів
- **Економічна значимість:** Враховує як традиційні фінансові метрики, так і стратегічні та ризикові фактори

2. Застосування для пріоритизації інвестицій:

- Порівняння проектів з різними характеристиками
- Балансування портфеля IT-інвестицій
- **Економічна значимість:** Забезпечує більш збалансований підхід до прийняття інвестиційних рішень

5.4.3. Моделі зрілості економічних процесів

1. Моделі зрілості управління IT-інвестиціями:

- Рівні зрілості: від ad-hoc до оптимізованого
- Процеси оцінки, планування, реалізації та вимірювання цінності
- **Економічна значимість:** Визначення шляхів удосконалення процесів управління IT-інвестиціями

2. Моделі зрілості управління бізнес-цінністю (Val IT):

- Узгодження IT-інвестицій з бізнес-цілями
- Управління портфелем інвестицій
- Реалізація цінності від інвестицій
- **Економічна значимість:** Максимізація віддачі від інвестицій у програмні системи

6. Висновки

У даній лекції ми розглянули основні фактори, що визначають економічні характеристики програмних систем. Розуміння цих факторів є критично важливим для успішного планування, розробки, впровадження та використання програмного забезпечення в умовах обмежених ресурсів та високої конкуренції.

Основні висновки:

1. **Програмні системи мають унікальні економічні характеристики, що відрізняють їх від традиційних виробничих товарів:** нематеріальність, висока вартість розробки при низькій вартості тиражування, залежність від людського фактора, мережеві ефекти, технологічне старіння, високі трансакційні витрати при зміні постачальника та складність оцінки вартості та термінів розробки.
2. **Вартість виробництва програмних систем визначається комплексом взаємопов'язаних факторів:**
 - Розмір і складність програмної системи впливають на вартість нелінійно, що вимагає застосування спеціальних стратегій управління ними.
 - Вибір технологій і методів розробки має суттєвий вплив на продуктивність, якість та вартість розробки.
 - Кваліфікація і досвід персоналу можуть визначати до 10-кратної різниці в продуктивності та якості.
 - Терміни розробки та вартість пов'язані нелінійно, і необхідно шукати оптимальний баланс між ними.
 - Ризики проекту мають бути враховані економічно обґрунтованим чином, з визначенням оптимальних стратегій управління ними.
3. **Ефективність використання програмних систем залежить від ключових якісних характеристик:**
 - Якість програмної системи визначає її надійність, продуктивність та вартість підтримки.
 - Відповідність вимогам користувачів забезпечує максимальну цінність системи для бізнесу.
 - Зручність використання впливає на продуктивність користувачів, витрати на навчання та рівень прийняття системи.
 - Надійність і безпека критично важливі для запобігання фінансовим втратам, репутаційним ризикам та юридичним наслідкам.
4. **Для прогнозування економічних характеристик програмних систем розроблено різноманітні моделі:**
 - Моделі оцінки вартості (COCOMO, функціональні точки, SLIM) дозволяють прогнозувати зусилля та витрати на розробку.
 - Моделі оцінки ефективності (аналіз витрат і вигод, ROI, TCO, реальні опції) допомагають оцінити економічну доцільність та ефективність інвестицій.
 - Моделі оцінки ризиків дозволяють врахувати невизначеність та прийняти більш обґрунтовані рішення.
 - Інтегровані моделі забезпечують комплексну оцінку з урахуванням різних аспектів цінності програмних систем.

Ефективне управління економічними аспектами програмних систем вимагає балансу між різними факторами та компромісів, що базуються на розумінні їх взаємозв'язків та впливу на кінцевий результат. Застосування відповідних моделей та методів дозволяє приймати обґрунтовані рішення, оптимізувати інвестиції та максимізувати цінність програмних систем для бізнесу.

7. Список літератури

1. Boehm, B. W. (2000). Software Cost Estimation with COCOMO II. Prentice Hall.
2. Jones, C. (2008). Applied Software Measurement: Global Analysis of Productivity and Quality. McGraw-Hill.
3. McConnell, S. (2006). Software Estimation: Demystifying the Black Art. Microsoft Press.
4. Brooks, F. P. (1995). The Mythical Man-Month: Essays on Software Engineering. Addison-Wesley.
5. Sommerville, I. (2015). Software Engineering. Pearson.

6. Nielsen, J. (2000). *Designing Web Usability: The Practice of Simplicity*. New Riders Publishing.
7. ISO/IEC 25010:2011. *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models*.
8. DeMarco, T., & Lister, T. (2013). *Peopleware: Productive Projects and Teams*. Addison-Wesley.
9. McGibbon, T. (1999). *A Business Case for Software Process Improvement Revised*. DoD Data Analysis Center for Software.
10. Biffl, S., Aurum, A., Boehm, B., Erdogmus, H., & Grünbacher, P. (2006). *Value-Based Software Engineering*. Springer.
11. Reifer, D. J. (2002). *Making the Software Business Case: Improvement by the Numbers*. Addison-Wesley.
12. Highsmith, J. (2009). *Agile Project Management: Creating Innovative Products*. Addison-Wesley.
13. Hubbard, D. W. (2014). *How to Measure Anything: Finding the Value of Intangibles in Business*. Wiley.
14. Kaplan, R. S., & Norton, D. P. (1996). *The Balanced Scorecard: Translating Strategy into Action*. Harvard Business Review Press.
15. Parker, M. M., & Benson, R. J. (1988). *Information Economics: Linking Business Performance to Information Technology*. Prentice Hall.