

Дидич С.О.,
Домбровський О.І.,
Коцюбан Г.П.,
Олійник О.С.

ПРОБЛЕМИ ВПРОВАДЖЕННЯ ПРИКЛАДНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Національного авіаційного університету

slavko_dydych@ukr.net

Розглянуто основні підходи до вирішення проблеми впровадження програмного забезпечення. За останній час комп'ютерні технології набули широкого використання для управління підприємством або організацією будь-якого типу. Постає проблема впровадження програмного продукту в систему роботи компанії. Сучасними методами вирішення цієї проблеми є процеси проектування спеціалізованого програмного продукту та процеси реінжинірингу. Розглянуто основні труднощі, які виникають в процесі впровадження програмного забезпечення, проаналізовано переваги та недоліки кожного з методів. Наведений аналіз статистичних даних які впливають на розробку програмного продукту

Ключові слова: програмне забезпечення, проектування, реінжиніринг, затрати на програмування, впровадження програмного забезпечення

Вступ

На сучасному етапі розвитку інформаційних технологій у багатьох як міжнародних, так і українських компаніях, управління, контроль, організація, проектування будь-яких аспектів роботи відбувається за допомогою того чи іншого програмного забезпечення. Відмітимо стрімке зростання ролі даного об'єкту на підприємствах протягом останніх років: оскільки компанії постійно здійснюють значні витрати на розробку, удосконалення, тестування та впровадження у виробництво програмного забезпечення. Звичайно, зрозумілим і є той факт, що розробка програмного забезпечення, стала об'єктом діяльності багатьох ІТ – компаній, які під замовлення розробляють якісне програмне забезпечення.

Якщо розглядати витрати на впровадження програмного продукту на підприємстві, на етапі дослідження проводиться оцінка цілей та мети, пов'язаних зі створенням даного програмного продукту. На даній стадії відбувається розробка бізнес-плану, встановлюються види необхідного матеріального забезпечення, ви-

значається технологія (необхідна для досягнення бажаних функцій), досліджуються шляхи досягнення бажаних функцій, проводиться порівняння альтернативних пропозицій постачальників та консультантів, визначається юридичне забезпечення тощо.

Стадія досліджень завершується отриманням певного наукового результату, який є передумовою початку розробки програмного продукту. Дана операція є наступною стадією процесу створення програмного продукту. На даному етапі компанія вже визначилася щодо способу ведення робіт по створенню майбутнього програмного продукту [8].

З огляду на різноплановість функціональних призначень створюваних програмних продуктів визначити загальні види робіт на даному етапі важко. До витрат, які слід капіталізувати, можна віднести прямі матеріальні витрати (матеріали, комплектуючі вироби, запасні частини тощо) та послуги виробничого характеру сторонніх організацій, що використовуються під час розробки програмного забезпечення, витрати пов'язані з виплатою заробітної плати працівникам, зайнятим у

виробництві даного об'єкту, відрахування на соціальні заходи, витрати на утримання та експлуатацію устаткування, загально-виробничі витрати тощо.

Актуальність проблеми

З огляду на зазначене, проблема розробки програмного забезпечення набула великого розмаху, оскільки на сучасному ринку існує велика кількість програмних продуктів і чи не найефективнішим спрямуванням коштів підприємства, буде придбання готового програмного продукту та проведення удосконалення певних модулів пакету, ніж проектування вузько направленої системи?

Дослідження та порівняльний аналіз можливих методів впровадження програмного забезпечення

Спробуємо дослідити необхідні витрати, проблеми та цілі при розробці програмного забезпечення та при удосконаленні придбаного програмного продукту, так званого реінжинірингу (*reengineering* – *eng.*: реорганізація бізнес-процесів, перестройка фірми, удосконалення програмного забезпечення) [1].

Введення в роботу підприємства певного програмного продукту, для керування чи організації роботи, доволі складний процес, який включає велику кількість аналізу, збору інформації та прийняття відповідних рішень. Розглянемо процес проектування певного програмного пакету під замовлення (широкого поширення набуло сленгове “з нуля”). Відразу ж можна виділити переваги такого методу:

- підприємство-замовник може відразу розробити вимоги до майбутнього продукту і має можливість замовити тільки необхідний вузько направлений функціонал;
- можливість чітко розрахувати терміни виконання розробки продукту;
- на сучасному етапі існує велика варіативність засобів розробки, тому завжди є можливість вибору ;

- можливість найму програмістів необхідної кваліфікації, є можливість заощадити при нескладних задачах та обов'язковій необхідності висококваліфікованих спеціалістів;

- можливість варіювання в методах та засобах, в залежності від фінансування підприємства чи компанії.

Але і певні недоліки:

- в залежності від кількості необхідного функціоналу розробка бажаного програмного продукту може зайняти тривалий час;

- якість програмного забезпечення в повній мірі буде залежати від кваліфікації виконавців;

- при зміні вимог, критеріїв чи переплануванні підприємства чи компанії, необхідним є доопрацювання програмного продукту, або повна його заміна;

- при складності нового програмного забезпечення є необхідністю навчання персоналу для роботи новим програмним забезпеченням.

- в штаті компанії необхідний працівник певної кваліфікації для технічної підтримки розробленого продукту.

Також розглянемо економічні витрати на розробку, спланованого програмного забезпечення. При розробці та введенні програмного продукту на підприємстві, необхідно враховувати сумарні затрати C_s протягом всього життєвого циклу програмного забезпечення, які складаються з затрат на розробку C_p , експлуатацію C_e і підтримку C_n [9]. Маємо: $C_s = C_p + C_e + C_n$

Для формування повного уявлення про затрати при розробці програмного забезпечення, обов'язково розглянемо і факт допущення, виявлення і виправлення помилок при розробці. Для подальшого конкретного порівняння з процесом реінжинірингу, наведемо графік затрат на виявлення і усунення дефектів на кожному етапі життєвого циклу програмного продукту рис.1.

Відсотковий вміст помилок, %

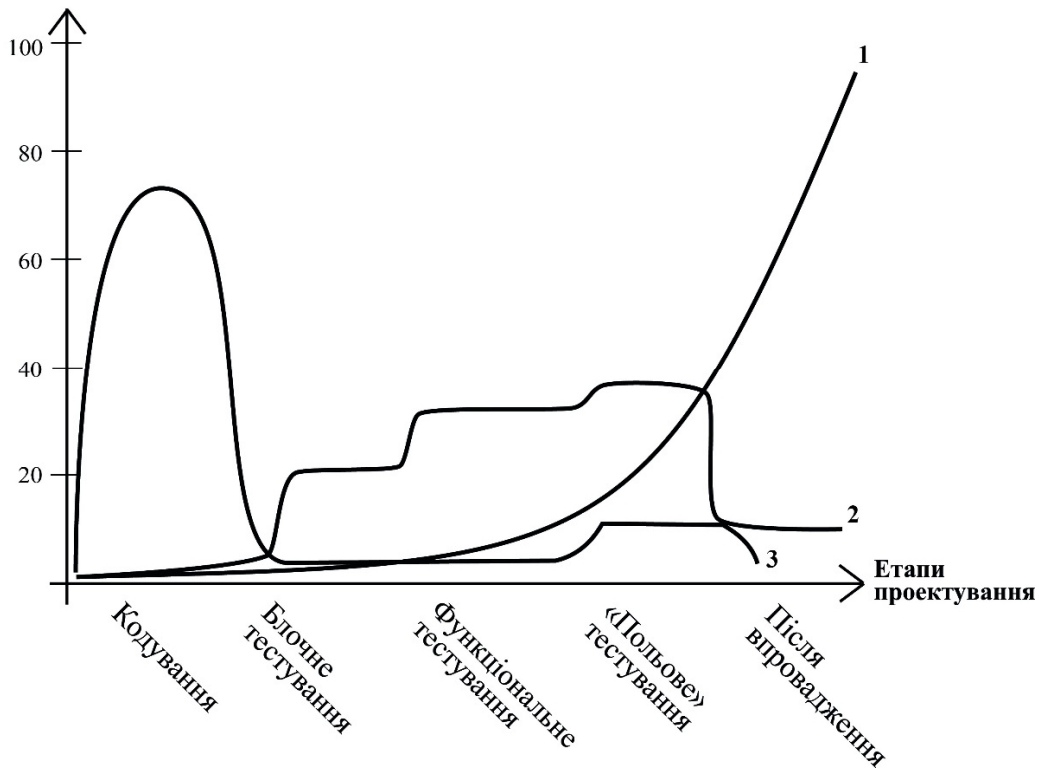


Рис. 1 Вартість усунення помилок

По мірі руху проекту вартість усунення помилок експоненціально зростає. Що дає змогу приблизно оцінити зміну затрат на різних етапах життєвого циклу програмного забезпечення:

- вартість усунення дефекту на даному етапі

підвищується в 10 разів з кожним етапом виявлення дефектів на даному етапі

Отже, систематизуємо витрати на розробку, та подамо у вигляді таблиці табл. 1. Розглядаючи фактори фінансового боку підприємства-замовника [7].

Таблиця.1 Витрати підприємства на розробку програмного продукту

Складові витрат	Групи витрат	Основні фактори які впливають на витрати при розробці
На безпосередню розробку комплексу програми	- Витрати по етапам розробки. - Витрати по категоріям спеціалістів.	- Об'єм, надійність. - Довготривалість розробки. - Довготривалість експлуатації. - Рівень технології розробки. - Рівень мови програмування.
На виготовлення пробного зразка	- Витрати на носії інформації. - Витрати на виготовлення і оформлення документації.	- Об'єм. - Рівень технологій розробки. - Спосіб матеріалізації програм.
На технології та програмні засоби автоматизації розробки	- Витрати на технології проектування. - Витрати на впровадження. - Витрати на експлуатацію.	- Об'єм, надійність. - Довготривалість експлуатації. - Рівень технології розробки. - Рівень мови програмування.

Недоречним буде ставити завдання чітко визначати витрати підприємства на проектування програмного продукту, оскільки це досить довготривалий та складний процес, який має свої особливості та методи для кожної конкретно поставленої задачі. Ще одним методом розробки програмного забезпечення є придбання готового пакету програмного продукту та проведення його адаптації до вимог замовника. Цей процес набув широко вживаної назви "реінженіринг". Проведемо порівняння опираючись на наведені факти.

Суть реорганізації програмного забезпечення полягає в покращенні або трансформації існуючого програмного забезпечення з метою легшого освоєння, покращення управління та використання його в майбутньому. Необхідність реінженірингу стає очевидною, коли існуючі програмні продукти застарівають з точки зору їх архітектури, платформи на якій вони працюють і їх нездатності задовольнити потреби користувачів. Складність полягає в розумінні існуючої системи. За-

звичай вимоги, дизайн і документація є неактуальними або дуже застарілими, так що не зовсім ясно які функції повинні бути впроваджені [8].

Реінженіринг починається з вихідного коду існуючої системи і завершується вихідним кодом цільової системи. З одного боку цей процес може бути таким же простим як і переведення коду з однієї мови на іншу, або з однієї операційної системи в іншу. З іншого боку процес реінженірингу може бути дуже складним і включати в себе використання існуючого вихідного коду для відтворення дизайну, визначення вимог до існуючої системи з подальшим їх порівнянням з необхідними вимогами, видалення ті, котрі більше не використовуються, реструктурування і редизайн системи (з використанням об'єктно-орієнтованого проектування), і, нарешті, створення коду для бажаної цільової системи. [2]

На рис.2 представлено загальну методику реорганізації програмного забезпечення [4].



Рис. 2 Структура реінженіринга

Систематичний аналіз реінженірингу проводиться над успадкованою системою до тих пір, поки «оцінка спроби» (рис. 3) не надасть такі альтернативи, що

на кроці «аналіз рішення» (рис. 4) не буде обрана одна з альтернатив: підтримка поточного стану успадкованої системи чи розробка нової системи.

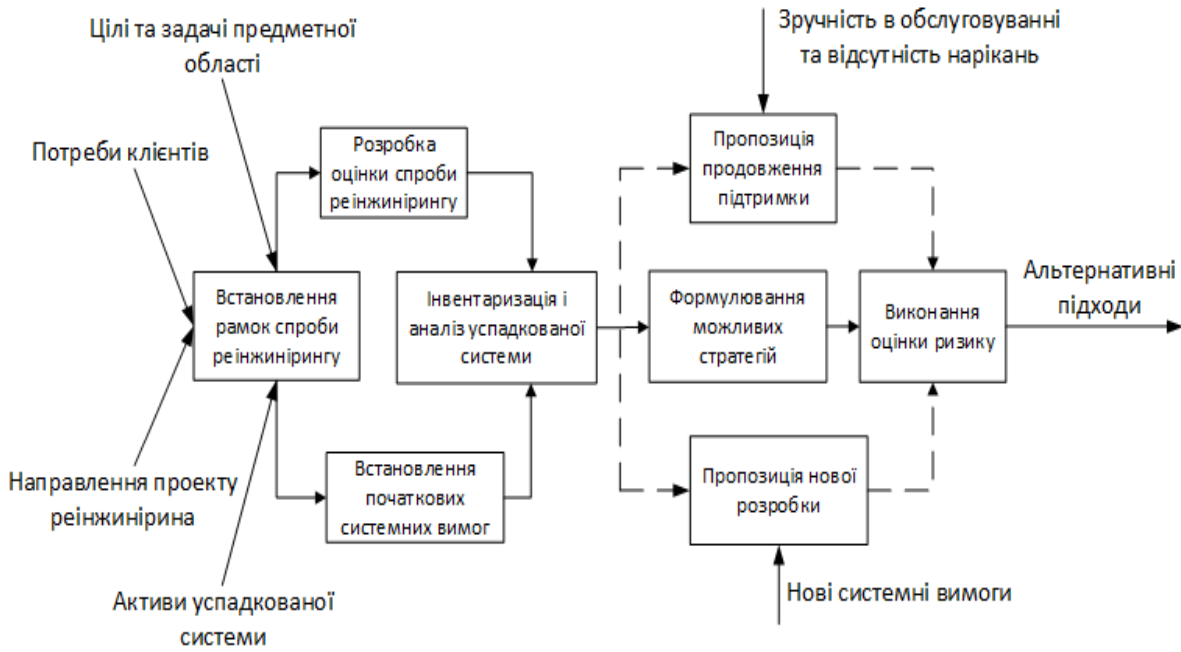


Рис. 3. Оцінка спроби реінжинірингу

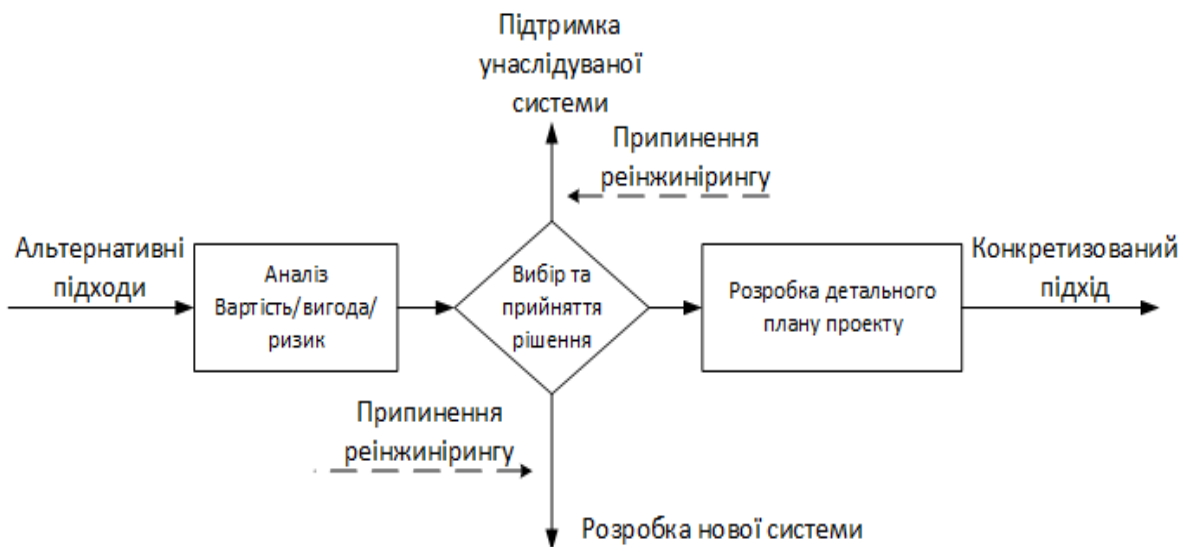


Рис. 4. Аналіз рішення

Найчастіше, реінжиніринг обходиться дорожче, ніж розробка програми. Причиною цього є те, що потрібно дотримуватися сумісності нової версії зі старою або ж реалізовувати конвертер старих даних в нові, а так само необхід-

ність обходити обмеження, нав'язані попередніми версіями програм.

Додаткові витрати на розробку програмного забезпечення шляхом реорганізації подамо у вигляді таблиці табл. 2 [7].

Таблиця 2. Додаткові витрати підприємства, які з'являються при реінжинірингу програмного забезпечення

Складові витрат	Групи витрат	Основні фактори які впливають на витрати при розробці
На підготовку підвищення кваліфікації спеціалістів		Предметна кваліфікація. Технологічна кваліфікація. Програмістська кваліфікація.
На виявлення та усунення існуючих проблем отриманого програмного забезпечення	Виявлення дефектів. Усунення дефектів.	Етап виявлення проблеми Значущість впливу дефекту на функціонал. Терміни виконання. Об'єм необхідної роботи.
На придбання готового програмного продукту	Придбання програмного забезпечення	Об'єм та функціонал цільового програмного забезпечення

Загалом досить невелика кількість компаній реально займається реінжинірингом програмного забезпечення. При консультації з розробником щодо реорганізації програмного забезпечення, найвірогіднішою порадою буде, почати проектування нового програмного продукту. У результаті замовник отримує іншу програму, яка може і вирішить ті проблеми які були, але яка вже, можливо, буде володіти новими проблемами, і не обов'язково програмного рішення.

Як правило, реінженіринг досить складний процес, який включає наступні недоліки [3]:

- програмісти не люблять розбиратися в чужому вихідному тексті;
- реінжиніринг не може зробити програміст низької та середньої кваліфікації. Для грамотного реінжинірингу потрібні експерти - програмісти з великим досвідом переробки програм і знанням різних технологій;
- виправляти чужу програму - велика відповідальність, тому що можна не помітити або не зрозуміти призначення певних алгоритмів, реалізованих попереднім програмістом;
- програмісту може знадобитися розбиратися з технологіями, з якими він не працює, але які використовуються в програмі, що підтверджує необхідність спеціалістів високої кваліфікації;

- в деяких випадках вартість базової програми та затрат на її реінжиніринг може виявитись значно вищою ніж вартість альтернативного програмного продукту, створеного "з нуля". Це, як правило, зумовлюється великим обсягом робіт, пов'язаних з доопрацюванням готової програми, необхідністю обійти обмеження попередніх версій, але при цьому дотримуватися сумісності за зростанням версій, тобто надати можливість конвертувати дані зі старих версій у нову.

Не зважаючи на ряд недоліків, програмний реінжиніринг має і свої ключові переваги [3]. Серед них:

- зниження ризиків. Оскільки програмний продукт вже існує, ризики пов'язані з допущенням похибок значно нижчі в порівнянні з розробкою нової програми;
- дозволяє мінімізувати витрати на написання системи з нуля, використовуючи функціональність програми колишніх версій;
- підвищення бізнес-вартості нового додатка, повторивши важливі дані та бізнес-логіку;
- так як нові системи засновані на оригінальних бізнес-логіках, це гарантує безперервність ключових бізнес-процесів;
- підвищення надійності системи і задоволення кінцевого користувача.

Висновок

Очевидно, реінжиніринг має безперечні переваги для замовника, що використовує напрацьований роками досвід, бізнес-логіку, а також функціонал, що автоматизує унікальні процеси підприємства. Для виконавця ж реінжиніринг є досить складним завданням, що вимагає високої кваліфікації розробників. Саме тому вартість реінжинірингу найчастіше вище вартості розробки з нуля. Аналізуючи статистику (Рис. 1) можна зауважити, що в процесі реінжинірингу існують абсолютно всі представлені затрати, а також додаються затрати на виявлення та усунення помилок реалізованого програмного коду. Не зважаючи на це, в результаті отримуємо стабільно працюючий програмний продукт і з значно дешевшим обслуговуванням. У той же час, якщо спочатку програма володіла суворою і ясною архітектурою, то провести реінжиніринг буде простіше. Порівнявши витрати на розробку та реорганізацію програмного забезпечення бачимо, що в процесі реінжинірингу додаються нові складові, які потребують залучення додаткових коштів. Тому при проектуванні, як правило, виділяють чимало часу на детальний аналіз рентабельності придбання та удосконалення програмного продукту у порівнянні з перевагами написання вузько направленої програмного забезпечення під конкретні цілі організації.

Список літератури

1. *Accounting Standards Committee. Accounting for Goodwill ED 47 (London: ASC, 1990), P. 30.*

2. *Eigler, Frank Ch., "Mudflap: Pointer Use Checking for C/C++", Proceedings of the GCC Developers Summit 2003, P. 57-70.*

3. *Software Technology Support Center, "STS Reengineering Technology Report, Vol 1", Hill Air Force Base, UT, 10/05.*

4. *Software Technology Support Center, "STS Reengineering Technology Report, Vol 2", Hill Air Force Base, UT, 10/05.*

5. *Бевз О.М., Папінов В.М., Скідан Ю.А. Проектування програмних засобів систем управління. Вінниця ВНТУ 2010. - С. 10-66.*

6. *Зубкова Т.М. Технология разработки программного обеспечения. Оренбург: ГОУ ОГУ, 2004. – 101 с.*

7. *Подкаменный Д.В. Методика адаптации унаследованных систем и ее инструментальная поддержка / Информационные технологии в науке и управлении / Труды XIII Байкальской Всероссийской конференции, ч. 2. – Иркутск: ИСЭМ СО РАН, 2008. – С. 211-219.*

8. *Положення (стандарт) бухгалтерського обліку 8 «Нематеріальні активи» // Бухгалтерія. – 2005. – №29(652).– С. 48-51.*

9. <http://www.rae.ru/monographs/141-4633>.

Статтю подано до редакції 2.04.2014