**УДК 004.413**

**ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБКИ ТА ВПРОВАДЖЕННЯ ЄДИНОГО ІНФОРМАЦІЙНОГ ПРОСТОРУ УНІВЕРСИТЕТУ**

**ECONOMIC EFFICIENCY OF DEVELOPMENT AND INTRODUCTION OF UNIQUE INFORMACIYNOG OF SPACE OF UNIVERSITY**

***І.М. Матвєєв, О.І. Куліш***

***І.М. Маtweew, О.І. Кulish***

***Анотація:*** *в даній статті розглядаються економічні аспекти розробки та впровадження єдиного інформаційного простору університету, створеного на архітектурі Клієнт - Сервер.*

***Аннотация:*** *в данной статье рассматриваются экономические аспекты разработки и внедрения единого информационного пространства университета, созданного на архитектуре Клиент - Сервер.*

***Annotation:*** *the economic aspects of development and introduction of single informative space of university, created on architecture Client is Server are examined in this article.*

***Ключові слова:*** *бази даних, інтерфейс, програмне забезпечення, інструменти для розробки, єдиний інформаційний простір.*

***Ключевые слова:*** *базы данных, интерфейс, программное обеспечение, инструменты для разработки, единое информационное пространство.*

***Keywords:*** *bases given, interface, software, instruments for development, single informative space.*

**Постановка проблеми.** Автоматизація роботи вузів, як і будь-яких інших структур науки і бізнесу досить актуальна. Вона дозволяє зосередити зусилля на першочергових справах, а інші перекласти на техніку. В результаті непроста рутинна робота виконується швидко і ефективно.

Автоматизація організації дає гарний ефект, але зазвичай використовується автоматизація робочих місць, це дозволяє певному співробітнику працювати більш ефективніше. Але в глобальному масштабі ефективність залишається досить малою, так як АРМи не можуть обмінюватись між собою даними. Що на практиці виявилось досить важливим. Виходить, що робота зроблена в одному АРМі повинна заново виконуватись в іншому, а значить АРМ не вирішує проблему в цілому.

Щоб роботу зробити більш ефективною, потрібно всі робочі місця об’єднати в єдину систему. Така система називається єдиний інформаційний простір. Інформація, яка потрапила в систему буде доступна кожному підрозділу, який їх потребує. Підрозділи будуть добре взаємодіяти між собою, що дає більш глобальний масштаб ніж автоматизація робочих місць. Але головне що дасть така система - це картина стану справ в реальному часі, що дозволяє ефективно впливати на ситуацію. У не автоматизованій або частково автоматизованій системі на виявлення реального стану речей йде багато часу і робочої сили.

Для реалізації поставленої задачі потрібно створити єдину базу даних, для роботи з нею створити АРМи, які б дозволили розподілити виконання задач між працівниками і при цьому дозволяли доступ тільки до тих даних, які необхідні даному співробітнику для вирішення його задачі. Тобто створити систему безпеки, яка б з одного боку захищала від несанкціонованого доступу, а з іншого дозволяла виконувати поставлені задачі закріпленим за ними особам. Дана система повинна бути мережною з базою даних на головному сервері.

Оскільки система буде побудована на єдиній базі даних, то її можна умовно розділити на дві частини. На головну, дані якої будуть використовуватись в усіх або майже у всіх АРМах. До таких даних відносяться дані про студентів і дані відділу кадрів, так як студенти, викладачі, працівники кафедр, деканатів і т.д. безпосередньо приймають участь в навчальному процесі. На основі цих даних організовується навчальний процес і система доступу. До іншої частини будуть віднесені специфічні набори таблиць призначені для збереження даних конкретних АРМів.

Функції програми, які повинні бути реалізовані:

* збереження найрізноманітніших даних (загальні дані студентів і викладачів, дані про успішність (журнали, залікові книги), навчальні плани, розклад, дані контрольні тощо);
* введення і редагування даних;
* статистична обробка даних;
* віддалений доступ до даних (так як навчальний заклад має 4 корпуси);
* призначення обмежень на доступ до даних;
* захист даних від несанкціонованого доступу;
* резервне копіювання.

Така система може не обмежуватись тільки роботою вузу, до неї можна включити і контрольно-пропускну систему, систему відео спостереження і навіть такі системи, як система автоматичного поливу клумб і газонів. Тобто як видно з усього вище сказаного можливості автоматизації і централізованого управління обмежуються тільки потребами, фантазією і фінансовими можливостями, а технології вже дано дозволяють це робити.

**Аналіз останніх досліджень.** Широке коло питань, пов’язаних з дослідженням перспектив розвитку принципів єдиного інформаційного простору організації, відображено в роботах вітчизняних і зарубіжних дослідників, таких як: Сілін А.Л., Сладкова О.Б., Столяров Ю.Н., Нісневич ЮЛ., Берестова Т.Ф.

На сучасному ринку інформаційних продуктів досить багато продуктів створені для автоматизації певних функцій навчального процесу. Дані продукти дозволяють виконувати певні задачі. Наприклад, перевіряти знання студентів - програми для тестування, або формувати розклад тощо. Але недоліком цих програм є те, що вони не зв’язані між собою. Тобто студент пройде тестування, потім викладач заносить ці дані в журнал. Для створення статистики, ці дані потрібно внести до спеціальної програми, яка дозволить побудувати якісь діаграми, для аналізу успішності і вже на основі них можна провести аналіз ефективності програми навчання. Такий стан речей не дозволяє ефективно керувати процесом навчання, так як дані оброблюються дуже повільно. Іншим недоліком такої системи є її економічна недоцільність бо в даному випадку, задіють додатковий персонал або час викладачів витрачається на складення паперових звітів. Які потім аналізуються іншим персоналом, який також отримує заробітну плату. Тому доцільне створення єдиного інформаційного простору.

**Мета дослідження.** Розглянути принципи побудови єдиного інформаційного простору університету, створеного на архітектурі Клієнт – Сервер та економічні аспекти його розробки та впровадження.

**Виклад основного матеріалу.** Метою створення єдиного інформаційного простору є об’єднання всіх відділів, які займаються організацією навчального процесу в єдину систему. Причиною такого об’єднання є те, що ці відділи дуже тісно зв’язані між собою. Є два варіанти такого зв’язку: дані одного відділу потрібні іншому, для виконання поставлених задач (наприклад, деканати користується даними прийомної комісії, так як студенти відбираються і реєструються саме нею) або відділ користується статистичними даними по іншим відділам, для впровадження якогось управлінського рішення (наприклад, аналіз успішності студентів може показати ефективність нової програми навчання і допомогти прийняти рішення по її вдосконаленні). Для такого об’єднання потрібна єдина система, яка б дозволяла виконувати всі поставлені задачі конкретним особам, причому система має бути досить гнучкою, так як певні задачі можуть призначатися іншим особам, при скороченні чи розширенні відділів або в зв’язку з іншими непередбачуваними обставинами. Крім того система повинна мати хорошу систему захисту і резервного копіювання.

Для єдиного інформаційного простору, однією з вимог якого є не прив'язаність працівника до робочого місця найдоцільнішим вибором є архітектура "Клієнт-Сервер". Такий вибір обумовлений тим, що для роботи такої системи потрібен централізований підхід збереження даних. Це означає що СУБД і сама база даних повинні зберігатися на окремому сервері, а клієнт повинен мати можливість доступу до цих даних віддалено.

Сам підхід "Клієнт-Сервер" має багато видів реалізації, але зважаючи на те, що фізично локальна мережа нашого університету побудована на основі витої пари, то навіть при встановлені високошвидкісних мережних карт на машини клієнтів мережа буде перевантажена. Така ситуація виникає із за особливостей роботи такої архітектури. Наприклад якщо БД і СУБД встановлені на сервері, а ПЗ клієнта встановлене на віддаленому ПК, то клієнт відправляє запит до сервера отримуватиме дані по каналу зв’язку. А зважаючи на об’єми цих даних і кількість користувачів, які одночасно працюють в системі, то швидкість обміну буде дуже низькою. А значить користувачі більше чикатимуть на обробку даних аніж працюватимуть. Тому користуючись досвідом адміністратора, підприємства на якому я проходив практику, архітектура мережі якого аналогічна запропонованій. Було прийнято рішення організувати архітектуру "Клієнт-Сервер" за принципом: "Товстого і тонкого клієнтів". В даному випадку СУБД, БД і ПЗ клієнта встановлюються на сервері, а клієнт зі свого терміналу або персонального комп’ютера з’єднується з сервером і працює на ньому. В даному випадку мережне обладнання фактично передає зображення для клієнта, який працює за терміналом, що суттєво розвантажує мережу[1, c. 120].

Крім вирішення проблеми з перевантаженням мережі, така система дає можливість використовувати застаріле обладнання, ресурсів якого замало, щоб використовувати його як персональний комп’ютер. А в подальшому дозволить економити на комп'ютером обладнанні. Так як комп’ютер-термінал значно дешевший ніж персональний. А зважаючи на те, що більшість комп’ютерів використовуються як друкарські машинки, то більшість відділів без проблем перейде на термінали.

Що до обладнання сервера, то до нього досить специфічні вимоги. Він повинен реалізовувати RAID - масиви, які збільшують стійкість до відмови обладнання, що захищає інформацію від виходу з ладу накопичувачів інформації. А також на сервері повинно бути встановлене високошвидкісне мережне обладнання. Для визначення інших параметрів а також оптимального співвідношення ціни - продуктивності проводиться дослідження ринку серверів.

Підсумовуючи вищесказане, можна дійти до висновку. Що на етапі розробки і тестування, досить апаратного забезпечення яке вже закуплено університетом. Для цього буде досить одного з серверів, які вже працюють і машини клієнта. А для впровадження єдиного інформаційного простору потрібно закупити серверне обладнання, так як можливостей вже існуючого серверного обладнання недостатньо. Особливо це відноситься до таких моментів, як резервне копіювання інформації. Так як серверне обладнання, яке використовується побудовано на архітектурі персонального комп’ютера. А така система не захищає дані від фізичного виходу з ладу носіїв інформації і не призначена для безперервної роботи, що дуже важливо для роботи обраної архітектури. Особливо зважаючи на те що деякі відділи навчального закладу працюють значно довше ніж основні. Що до програмного забезпечення то на сервері і персональних комп’ютерах які будуть клієнтами для даної системи буде використовуватися операційна система Windows (Windows Server 2003-для сервера і Windows XP для персональних комп’ютерів), таке рішення прийняте тому, що дані системи закуплені університетом. Ще одним аргументом на користь цього рішення є те, що система єдиного інформаційного простору буде написана під ці операційні системи. А от, що до терміналів то на них можна використати операційну систему Linux, головною перевагою такої системи є її безкоштовність. Що до інших аспектів, то в даній області проводиться дослідження.

Що до розробки програмного забезпечення, то було проаналізовано багато продуктів, які дозволяють створювати БД і реалізувати програмне забезпечення клієнта. Серед них були: Delphi 7, Microsoft Visual Basic 6, Microsoft Visual Basic 2005 Express Edition, Microsoft SQL Server 2005 Express Edition, Visual Fox Pro 9.0, Microsoft Office Access 2003. Delphi 7, Microsoft Visual Basic 6 - були відкинуті по причині моральної застарілості і закінчення ліцензії на дані продукти. Visual Fox Pro 9.0 дозволяє створити і клієнт і БД але він дає мало можливостей для захисту інформації. Тому було обрано три продукти, а саме: Microsoft Visual Basic 2005, Microsoft SQL Server 2005 Express Edition, Microsoft Office Access 2003. Microsoft SQL Server 2005 Express Edition - має достатній рівень захисту і вбудовані утиліти для копіювання і відновлення даних, крім того дана версія безкоштовна. Після вичерпання його можливостей можна перейти на необмежену версію закупивши ліцензію. Microsoft Office Access 2003 - дана програма обрана як конструктор для прототипів. Головною перевагою такого клієнта є те, що він дозволяє досить швидко вносити зміни до продукту, а значить таку систему можна швидко підганяти під потреби. Це дає можливість до кінця визначити вимоги до продукту і реалізувати всі потрібні функції. Але дана програма має один суттєвий недолік, вона платна. Хоча для роботи програмного забезпечення створеного на ній досить Access Runtime, але коли діло доходе до створення інсталяційного пакету з цим виникає проблема. Тому було прийнято рішення проводити розробку в два етапи. Спочатку виготовити прототип, а потім створити реальний продукт на Microsoft Visual Basic 2005. Головною перевагою такого підходу я те, що Visual Basic дозволить більш гнучко використовувати базу даних, подбати про систему захисту, створювати інсталяційні пакети тощо[2, c. 36].

Програмне забезпечення для роботи з базами даних зазвичай будується за такими принципами:

1. будується один програмний продукт, який реалізує всі можливі функції, клієнт використовує тільки ті функції в яких він компетентний;
2. будується декілька програмних продуктів, які реалізують ті функції, які потрібні певному підготовленому спеціалісту для реалізації поставлених задач.

Для даного випадку доцільніше було б використати другий підхід. Але його недолік в обмеженості функцій. Наприклад, якщо одна особа замінює іншу, то для її роботи потрібно встановити їй додаткову копію ПЗ, яка б дозволила тимчасове виконання функцій. Крім того, потрібно дозволити цій особі доступ до таблиць, які вона раніше не використовувала. Такий підхід не дозволяє швидко реагувати на такі обставини. Крім того кожен з цих підходів має суттєвий недолік. Він відноситься більше до програмування ніж до експлуатації. Суть його в тому, що великий продукт, який має безліч функцій важко піддається корегуванню помилок. Оскільки в продукті досить багато коду, то потрібно витрачати час щоб зрозуміти, яка саме частина цього коду викликає збій. А якщо бути більш точним, то де і в яких випадках виникає помилка. Тому була розглянута ідея, яка була подана одним з викладачів, який досить часто стикався з такими проблемами, при супроводженні великих програмних продуктів.

Суть ідеї в створенні в використанні Plugin - модулів. Plugin - це зазвичай окремий файл, або декілька файлів, які поставляються автором продукту або сторонніми виробниками і дозволяють розширювати можливості готового програмного забезпечення без потреби переписувати програмний код цього продукту. Тобто, це модуль який програмується спеціальним чином і в потрібний момент може бути підключений до готового програмного продукту, для розширення його функцій. Мова програмування Microsoft Visual Basic 2005 підтримує реалізацію такої можливості[3, c. 78].

Якщо розглянути програмні продукти, які працюють з базами даних, то можна виявити цікаву закономірність, а саме - всі функції, які реалізуються, мало зв’язані. Тобто, якщо потрібно відредагувати певний довідник, то викликається окрема форма, яка дозволяє працювати саме з цією таблицею. Зазвичай, ця форма мало зв’язана з іншими і може бути відділена в окремий модуль. Тепер, якщо декілька різних АРМів працюють з цим довідником, то до них потрібно просто приєднати цей модуль. Якщо в ньому виникають помилки, то для їх пошуку і усунення потрібно значно менше часу, так як цей модуль містить набагато менше коду і, що більш важливо, мінімально взаємодіє з іншими часинами програмного коду[4, c. 126].

Такий підхід дозволяє не створювати окремі АРМ, а створювати набори Plugin - модулів для їх реалізації. Що в свою чергу дозволяє без зусиль створювати максимально функціональні продукти, не відволікаючи користувачів додатковими меню і функціями. Причому для створення таких продуктів не потрібно змінювати код, досить просто приєднати потрібний Plugins-модуль.

На основі вищесказаного видно, що продукт має складатися з двох частин: головного модуля і набору Plugins - Модулів, які дозволяють виконувати спеціалісту поставлені задачі. Головний модуль повинен включати в себе три механізми, а саме:

* механізм безпеки;
* механізм доступу до бази даних;
* механізм для підключення Plugins - модулів.

В свою чергу механізм доступу до БД, повинен включати механізм авторизації на Microsoft SQL Server 2005, а також механізми читання і запису даних. Механізм безпеки повинен давати можливість, перевіряти особистий підпис користувача і на основі таблиці безпеки підключати лише дозволені Plugins - модулі. Механізм для підключення Plugins - модулів - реалізується програмно і дозволяє Plugins - модулям, через механізм доступу до БД брати звідти дані, обробляти і виводити всю потрібну інформацію в головному модулі[5, c. 145].

Даний підхід максимально виправдовує себе при використанні термінального сервера. Оскільки на накопичувачі буде знаходитись один програмний файл, який реалізовуватиме головний модуль і Plugins - модулі. Щоб налагодити цей продукт для різних користувачів, адміністратору досить відредагувати таблиці безпеки. В них він повинен призначити ім’я і пароль, для доступу до електронного підпису і таблицю Plugins - модулів користувача. Коли користувач введе ім’я і пароль для доступу до особистого підпису, система отримає доступ до таблиці Plugins - модулів користувача, з якої прочитає інформацію про потрібні модулі і задіє їх. Ті в свою чергу створять головне меню. Така система дозволяє швидко реагувати на непередбачувані обставини і змінювати можливості користувачів не перевстановлюючи програмне забезпечення.

Для реалізації програмного забезпечення такого рівня потрібно звернути особливу увагу на питання безпеки. Бо навмисне змінювання або знищення даних такої системи матиме негативні наслідки.

Обладнання сервера підлягає більш жорстким стандартам якості і напрацювання на відмову. Тобто використовуються більш якісні матеріали, додаткові системи дублювання інформації, системи безперервного живлення, високошвидкісне обладнання. А головне така система набагато потужніша за звичайний комп’ютер, так як все навантаження перекладається на неї. Все це зумовлює більш високі ціни на таке обладнання.

Наступним позитивним ефектом в такій технології є економія на програмному забезпечені. Оскільки все ПЗ встановлюється на сервері, то і витрати на ліцензії значно знизяться. З іншого боку розроблене ПЗ перейде у власність університету, а унікальність продукту дозволить в подальшому створити свою торгову марку і продавати його подібним структурам. В такому разі сума витрачена на розробку і впровадження окупиться за декілька років. Крім цього не потрібно витрачати кошти на навчання персоналу і оновлені версії так, як цим будуть займатись самі працівники ІТ-відділу.

Крім цього звільняться площі, які використовуються для збереження паперової документації, так як більшість документів відтепер буде реалізована в електронному вигляді.

Ще однією перевагою такої системи є можливість віддаленої взаємодії викладача і студента, тобто використання дистанційного навчання.

**Висновки**

Отже, як видно, така система дає багато переваг як суто практичних так і економічних.

Перше, що дає така система, це можливість перегляду і редагування даних, відповідно до допуску, з будь-якого робочого місця, або загального терміналу (для студентів, так як вони постійного робочого місця не мають). Зменшується час підготовки стандартних звітів, так як ця операція автоматизована і для їх підготовки потрібно натиснути відповідну кнопку. Ще однією можливістю є отримання статистичних даних і оцінок роботи, як навчального закладу в цілому так і окремих його частин. Це дасть можливість корегувати роботу навчального закладу.

Друге, що дає така система, це зменшення впливу людського фактору на роботу, збільшення конфіденційності і безпеки даних. Так, як система дозволяє обмінюватися даними в електронному вигляді. А це в свою чергу унеможливлює втрату даних, або перегляд їх сторонніми особами, що можливо при паперовому зберіганні. Централізоване збереження даних на сервері, дозволяє їх ефективно захищати від несанкціонованого доступу. Крім того реалізується функція резервного копіювання, яка дуже ускладнена, якщо система складається з окремих АРМів, бази даних яких знаходяться у кінцевого користувача.

**Джерела та література**

1. Кириллов В.В. Основи проектування реляційних баз данных / Володимир Володимирович Кириллов. — М.: Видавництво «Вільямс», 2005 – 305 с.
2. Інформаційні системи і технології в економіці. Посібник для студентів вищих навчальних закладів / [За редакцією Пономаренка В.С.] – К.: Видавничий центр “Академія”, 2002. – 544 с.
3. Липаев В.В. Оценки затрат на разработку программных средств / Липаев В.В., Потапов А.И. – М.: Финансы и статистика, 2001 -348 с.
4. Новоженов Ю.В. Объектно-ориентированные технологии разработки сложных программных систем /Новоженов Ю.В. –М., Наука, 1996. – 242 с.
5. О’Брайен Т. Microsoft Access. Разработка приложений /О’Брайен Т. –СПб.: БХВ - Санкт-Петербург, 1999. – 640 с. : ил.