

*Шафорост В. В., студент 2
курсу спеціальності 122
«Комп'ютерні науки»
Потапова Н. А., к.е.н., доцент,
доцент кафедри інформаційних
технологій*

АЛГОРИТМІЗАЦІЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Донецький національний університет імені Василя Стуса, м. Вінниця

Актуальність розробки програмного забезпечення сприяє в його адаптації до потреб кінцевого споживача які доволі часто змінюються у часі, враховуючи специфіку використання для конкретного підприємства і його стратегічних цілей. Основу розробки програмного забезпечення становить ефективність алгоритму. Тому, поєднання процесів алгоритмізації та розробки програмного забезпечення є одним із найважливіших завдань успішної реалізації програмного продукту та його освоєння на ринку.

Алгоритмізація – це розділ інформатики. Метод системного опису або процесу формування алгоритму діяльності системи. Алгоритмізація процесу – це опис процесу математичними символами для одержання алгоритму.

Розробка програмного забезпечення — це управління проектами, методологіями та практиками, спрямованими на створення та забезпечення продуктивності та гарантованої якості програмного забезпечення. [1]

Програмне забезпечення має різні цілі, виконувати функції та формати реалізації. У цьому сенсі програмне забезпечення є складною і досить унікальною системою. Однак є деякі загальноприйняті принципи, які потрібно використовувати у своєму програмному забезпеченні:

1. Частотний принцип. Він заснований на визначенні та частоті використання алгоритмів і оброблених структур даних. Для спільних дій під час роботи програмного забезпечення передбачені умови для їх швидкої реалізації. Часто доступні дані забезпечують найшвидший доступ і намагаються скоротити такі операції.

2. Модульний принцип. Під цим принципом у загальному випадку передбачаються функціональні елементи системи. Він має повний дизайн і вимоги та засоби підключення до подібних або високорівневих елементів системи. Можливий спосіб розділення програмних компонентів на окремі модулі, використання поділу на основі функцій. Здебільшого поділ системи на модулі застосовується методом проектування програмного забезпечення.

3. Принцип функціональної вибірковості. Вказаний принцип є наступником попередньо перерахованих принципів і впроваджується при розробці програмного забезпечення, яке значно перевищує обсяг доступної оперативної пам'яті. Програмне забезпечення містить кілька важливих модулів і

повинно завжди бути готовим для перспективної роботи обчислювального процесу. Ця частина програмного забезпечення називається монітором. При утворенні відповідності монітора повинні бути виконані дві суперечливі вимоги. Згідно з монітором, окрім чистих модулів управління слід використовувати найбільш часто використовувані модулі. Кількість модулів повинна гарантувати, що монітор не займає занадто багато пам'яті. Програмні продукти, що є невід'ємною частиною монітора, завжди знаходяться в пам'яті. Інша частина програмного забезпечення постійно зберігається в інших збережених розширеннях, завантажується в оперативну пам'ять лише за необхідності, а в деяких випадках перекривається один з одним.

Розробка програмного забезпечення передбачає утворення, проектування та сервісну підтримку комп'ютерного програмного забезпечення. Багато років проблема полягала в пошуку повторюваних і передбачуваних процесів або методологій, які покращують продуктивність, якість і надійність. Деякі люди намагаються систематизувати і формалізувати цей, здавалося б, непередбачуваний процес. Наступні застосували до даного процесу методи управління проектами та розробки програмного забезпечення. Інші думали, що без жорсткого контролю споживачів створення програмного забезпечення вийшло б з-під контролю, що забиравало б додатковий час і гроші. [2]

Навички створення програмного продукту відображені у спеціальних методичних посібниках. Якщо ваша розробка використовує кілька стандартів або правил, то має сенс створити профіль. Ця методологія спрямована на вирішення проблем на вашому комп'ютері. Це схожа технологія для вітчизняних студентів і програмістів для розробки алгоритмів і програм, що використовувалися на змаганнях з програмування з 1970-х років, використовуючи тести та структурний псевдокод для документування програм в ІВМ. Методології проектування структурованого програмного забезпечення можна використовувати на різних мовах та інструментах програмування для розробки надійних програм для різних цілей. Одним з таких проектів була розробка бортового програмного забезпечення для космічного корабля *Blanc*. Вперше був використаний бортовий комп'ютер для автоматичного керування космічним кораблем, який успішно запустив та посадив його. У виборі методу розробки програмного забезпечення потрібно звертати увагу на складність методів порівняно із складністю будови програми, а для продукту такої складності розвивається невиправдана складність методології. Наприклад, у сучасній методиці розробки є проблемно-орієнтоване проектування. [3]

Найпоширеніші проблеми, які виникають у процесі розробки програмного забезпечення:

- Відсутність прозорості. Завжди важко сказати, що це за проект і який відсоток виконання проекту. Ця проблема виникає через погане планування структури майбутніх програмних продуктів. Часто це є наслідком нестачі фінансування проекту.

- Відсутність контролю. Якщо ви не точно оціните хід розробки, ваш робочий графік буде порушено, а бюджет буде перевищено. Важко оцінити обсяг виконаної роботи та роботу, що залишилася.

Ця проблема з'являється у випадку, коли більше половини проекту завершено, а розробка триває навіть після того, як було здійснено додаткове фінансування без оцінки завершеності проекту.

- Відсутність моніторингу. Ви не можете стежити за ходом розробки в режимі реального часу, оскільки не можете контролювати стадію розробки свого проекту. За допомогою певних інструментів керівники проектів схвалюють обґрунтовані рішення, враховуючи дані в реальному часі.

Цей недолік з'являється через те, що кінцеві користувачі вагаються щодо використання певного програмного середовища. Наприклад, при створенні системи клієнт/сервер споживач встановлює вимоги не тільки до клієнтського комп'ютера, а й до операційної системи серверного комп'ютера. [4]

Таким чином можна зробити висновок, що ефективність застосування принципів і методологій розробки програмного забезпечення мають значний успіх при використанні їх в особистому досвіді великої кількості компаній та підприємств на ринку інформаційних технологій.

Список літератури

1. *U Distance Learning*. URL: <https://dl.sumdu.edu.ua/textbooks/108989/459139/index.html>.
2. Цибульник, С. О. *Технології розробки програмного забезпечення*. Київ: КПІ ім. Ігоря Сікорського, 2021. 25 с.
3. Зелінська О.В., Потапова Н.А., Волонтир Л.О. *Інформаційні системи та технології в галузі. Навчальний посібник*. Вінниця: ВНАУ, 2020. 253 с.
4. Галіцин В.К., Сидоренко Ю.Т., Потапенко С.Д. *Г 15 Технологія програмування і створення програмних продуктів: Навч. посіб. К.: КНЕУ, 2009. 372 с.*

УДК 004:378

*Якубич К. О., студентка
Зелінська О.В., к.т.н., доцент, доцент
кафедри інформаційних технологій*

РОЗРОБКА ВЕБ-ДОДАТКУ МОНІТОРИНГУ ЗАКЛАДІВ ОСВІТИ МІСЦЕВОГО РІВНЯ

Донецький національний університет імені Василя Стуса, м. Вінниця

В Україні активно розвивається цифрова галузь, що обумовлено стрімким процесом діджиталізації. Це глибинне проникнення цифрових та інноваційних технологій до бізнес-процесів, господарства, комунікацій. Перехід економіки до ери цифрових можливостей привело до необхідності діджиталізації управлінських процесів, запровадження цифрових продуктів з метою задоволення попиту на ринку [1].