

# Життєвий цикл розробки програмного забезпечення

Д.Г. Шевченко, студент,  
О.Л. Лєвошко, ст. викладач  
Кіровоградський національний технічний університет

**Життєвий цикл програмного забезпечення** - період часу, який починається з моменту прийняття рішення про необхідність створення програмного продукту і закінчується в момент його повного вилучення з експлуатації. Цей цикл - процес побудови і розвитку ПЗ.

**Цикл розробки програмного забезпечення** - структурований поділ, покладений в процесі розробки програмного продукту. Існують кілька моделей для цього процесу, кожна з яких описує свій підхід до елементів поділу, присутніх у процесі розробки програмного забезпечення.

## Дії у процесі написання програмного забезпечення

Дії у процесі написання, представлені в моделі водоспаду (рисунок 1). Є й інші моделі, які по іншому описують цей процес.

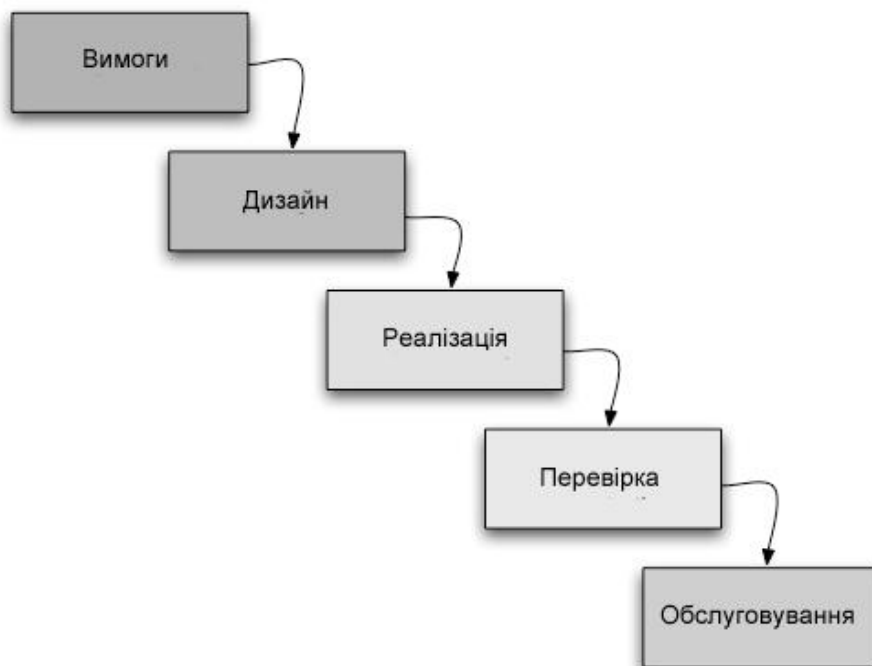


Рисунок 1 – Водоспадна модель

## Аналіз вимог до продукту

Найважливіше завдання при створенні програмного продукту - це вироблення вимог, або аналіз вимог до продукту. Замовник найчастіше представляє досить розмиту ідею про те, яким повинен бути кінцевий результат, і не має уявлення про те, як повинна працювати програма. Незакінчені, безглузді, а іноді суперечать один одному вимоги розпізнаються добрими інженерами на цій стадії. Часта демонстрація живого коду може зменшити ризик того, що початкові вимоги були не вірні. Один з методів знаходження проблем такого роду - це аналіз елементів програмного забезпечення.

Коли загальні вимоги отримані від клієнта, їх необхідно уточнити і відобразити в документі. Реалізована функціональність може відрізнятися від визначеної, в результаті високої вартості розробки та / або незрозумілих вимогах до продукту. Якщо розробка проводиться поза фірмою замовника, то даний документ може використовуватися для вирішення питань пов'язаних з функціональністю продукту.

Аналіз області роботи часто є першою сходинкою проектування нового фрагмента програмного забезпечення, незалежно від того, чи є він додаванням до вже існуючого додатка або новим додатком, підсистемою або зовсім новою системою.

### **Специфікація**

Специфікація це завдання, що описує до дрібниць програмний продукт, який буде написаний, можливо, у вигляді суворого опису. На практиці більшість вдалих специфікацій написані для того, щоб зрозуміти і відточити вже написані програми. Специфікації найбільш важливі для зовнішніх інтерфейсів, які повинні залишатися стабільними. Гарний спосіб визначити, чи добра специфікація - це попросити третю сторону провести аналіз, щоб переконатися що вимоги і способи їх рішень логічно вірні.

### **Архітектура**

Архітектура системи програми створюється для того, щоб бути впевненим, що програмне забезпечення буде виконувати вимоги які покладені на нього, а також залишає можливість для того, щоб додавати рішення для нових вимог. Так само на етапі архітектури вирішуються проблеми інтерфейсів між програмним забезпеченням і операційною системою, чи обладнанням.

### **Проектування, реалізація і тестування**

Проектування - процес створення загальної архітектури і алгоритмів відповідно до специфікацій. Реалізація (імплементация) - це та частина процесу, під час якої програмісти власне створюють програмний код продукту. Тестування - всеосяжна і важлива частина процесу розробки програмного забезпечення. Ця частина процесу полягає в тому, щоб виявити і вирішити різні помилки. Документування проводиться для того, щоб у майбутньому було простіше підтримувати і покращувати програмний продукт. Це також може в себе включати опис зовнішніх або внутрішніх програмних інтерфейсів.

### **Розповсюдження та підтримка**

Поширення починається після того, як код достатньо відтестований, і визнаний готовим до релізу.

Технічна підтримка та навчання важливі, тому що великий відсоток проектів провалюється тому, що багато розробників не розуміють, що скільки б не було витрачено часу на програмний продукт, він буде безглуздим, якщо його ніхто не використовує. Люди часто чинять опір і уникають змін програмних продуктів, так що дуже важливо провести навчання нових клієнтів.

Підтримка та поліпшення продукту разом з виправленням знайдених помилок може займати більше часу, ніж власне процес розробки цього продукту. Може бути корисним підкорегувати код, який не підходить по дизайну - це може спростити знаходження помилок і їх виправлення до того, як їх помітять користувачі.

## **Список літератури**

1. ИТ с высоты птичьего полета / Блог Сергея Орлика / <http://sorlik.blogspot.com>
2. Цикл разработки программного обеспечения – Википедия. / [tp://ru.wikipedia.org/wiki/Цикл\\_разработки\\_программного\\_обеспечения](http://ru.wikipedia.org/wiki/Цикл_разработки_программного_обеспечения)