

## СЕКЦІЯ 7. АВТОМАТИЗАЦІЯ ТА КОМП'ЮТЕРНО-ІНТЕГРОВАНІ ТЕХНОЛОГІЇ

УДК 621.396.67

### ЖИТТЄВИЙ ЦИКЛ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНЖЕНЕРНИХ ОБ'ЄКТІВ

Васюк В. В., к.т.н., доц., e-mail: [vasyuk@nubip.edu.ua](mailto:vasyuk@nubip.edu.ua)

Національний університет біоресурсів і природокористування України

**Актуальність дослідження.** Необхідність застосування для вирішення відповідних завдань комплексу що охоплює програмну та технічну складову, обумовлює потребу впочатку реалізації процесів поступової зміни станів програмного забезпечення. Дані процеси за певними ознаками можна поділити на чіткі стадії, послідовність яких створює життєвий цикл програмного забезпечення. При цьому змістовна суть життєвого циклу програмного забезпечення не зводиться до виконання конкретних процедур і процесів у певний проміжок часу, а спрямована на зміну стану цього програмного забезпечення, і така зміна станів має бути спрямована на досягнення конкретної мети – отримання об'єкта, функціонування якого відповідає певним вимогам і побажанням. При цьому слід пам'ятати, що будь-яке програмне забезпечення слід розглядати як складну систему. Як складну систему необхідно розглядати також і категорію якості будь-якого програмного забезпечення – інженерного об'єкта з заданими властивостями [1-4].

**Метою дослідження** є проведення аналізу життєвого циклу програмного забезпечення інженерних об'єктів. Це призводить до необхідності планувати та підтримувати, починаючи з початкових етапів життєвого циклу розробки програмного забезпечення, належний рівень якості впровадження всіх процесів і процедур, спрямованих не лише на безпосереднє написання програми, а й з точки зору матеріально-технічне, організаційне та соціальне забезпечення зазначеного життєвого циклу, адже, будучи системною категорією інженерних об'єктів (програмного забезпечення в даному контексті), якість програмного забезпечення залежить від компонентів, які ні за яких обставин не можуть бути вбудовані в цей інженерний об'єкт (розроблене програмне забезпечення).

**Основні матеріали досліджень.** Реалізація процесів і процедур, спрямованих на забезпечення, підтримку та адекватне управління рівнями якості програмного забезпечення на різних етапах життєвого циклу останнього, обов'язково спирається на модель зазначеного життєвого циклу як методологічну основу для досягнення мети – зробити кінцевий продукт в умовах експлуатації та середовищі до відповідного рівня споживчої якості. Модель життєвого циклу в контексті програмної інженерії визначається як структура, яка визначає (встановлює) послідовність реалізації та взаємозв'язки всіх процесів і процедур різної природи (технічних, організаційних, фінансових тощо), що виникають під час здійснення будь-яких дій щодо програмного забезпечення, починаючи з появи потреби в такому програмному забезпеченні та закінчуючи його остаточним виведенням з експлуатації. Модель життєвого циклу обов'язково «прив'язана» до змісту, розміру та складності відповідного проекту, а також до конкретних умов і обставин, за яких програмне забезпечення було створено та експлуатується. Ретельно розроблені та адаптовані до чинних умов і обставин моделі життєвого циклу що стають основою для впровадження загального управління якістю в рамках відповідного проекту програмного забезпечення та в цей час розглядаються як методологія для отримання відповідності в результатах проекту (між замовником і виконавцем) вимоги та/або основа для об'єктів сучасних стандартів якості.

На початкових етапах життєвого циклу програмне забезпечення можна характеризувати набором бажаних (для замовника) споживчих атрибутів, тобто набором споживчих атрибутів, запропонованих замовником для програмного забезпечення, незалежно від можливостей програмного забезпечення віртуального підприємства (тобто проектування таких можливостей програмного забезпечення, його подальше тестування, робота тощо). За результатами обговорення всіма зацікавленими сторонами заявлених вимог і побажань замовника програмне забезпечення переходить у наступний стан, який характеризується набором споживчих атрибутів (точніше, узгоджених споживчих атрибутів)

програмного забезпечення. У майбутньому програмне забезпечення переходитиме до стану, який характеризується наявністю відповідних специфікацій, програмних текстів, документів тощо.

**Висновок.** Отже, відповідно до концепції реалізації життєвого циклу, запропонованого на деякому віртуальному підприємстві, програмне забезпечення послідовно переходить від одного стану до іншого, від заявлених клієнтом споживчих атрибутів до готового програмного продукту з реальними споживчими атрибутами, підкріпленими відповідними характеристиками; це програмне забезпечення використовується до тих пір, поки воно морально не застаріє, і врешті-решт перейде в стан, що характеризується оновленням або припиненням використання програмного забезпечення.

Життєвий цикл програмної складової програмно-технологічного комплексу значною мірою збігається з життєвим циклом програмного продукту широкого застосування і є предметом програмно-інженерних досліджень. При цьому, безумовно, всі процеси кожного етапу життєвого циклу програмних компонентів програмно-технічного комплексу (це переважно програмне забезпечення що належить до класу замовленого) повинні бути прив'язані за цілями та термінами виконання до відповідних процесів життєвого циклу апаратно-технічної складової, а також до програмно-технічного комплексу в цілому.

Моделі життєвого циклу для програмних («віртуальних») і об'єктів матеріальної інженерії мають спільні риси, але багато в чому вони суттєво відрізняються через суттєві відмінності в характері змісту зазначених об'єктів. На відміну від об'єктів матеріальної інженерії, програмні («віртуальні») інженерні об'єкти (програмного забезпечення) не мають ознак фізичного зносу, однак під час експлуатації програмного забезпечення помилки зазвичай виникають за тих чи інших умов, які з необхідністю потребують виправлення. Зазначені помилки є не тільки результатом некоректного виконання певних процедур у процесі розробки програмного забезпечення, але можуть бути спричинені зміною умов використання зазначеного програмного забезпечення, тобто «прив'язані» до конкретних умов та обставин застосування відповідного програмно-технічного комплексу.

Невідповідність споживчих властивостей програмно-технічного комплексу умовам застосування, що за об'єктивних причин змінилися з початку експлуатації даного комплексу, переважно ідентифікується як його моральне старіння. При цьому постає питання модернізації (або заміни на нову модель) цього програмно-технологічного комплексу. Якщо ця модернізація також торкнеться програмних компонентів програмно-технологічного комплексу, то наявне програмне забезпечення визнається непридатним для використання та класифікується як «застаріле». У цьому випадку цілком правомірно ідентифікувати факт «старіння» програмного забезпечення, маючи на увазі, звичайно, його моральне старіння.

Пильна увага до життєвого циклу програмного забезпечення, призначеного для «широкого кола користувачів» і використовуваного у складі конкретного програмно-технологічного комплексу, полягає, зокрема, у різних видах і цілях, пов'язаних із сучасними підходами до досягнення високого рівня якості продукції. Зазначена методологія враховує системний характер як категорії якості, так і життєвого циклу програмного забезпечення, в значною мірою затверджує чітку реалізацію, а також контроль не результатів здійснення визначених процесів та процедур в межах вказаного життєвого циклу, а саме процесів, що супроводжують життєвий цикл відповідного продукту.

#### ПЕРЕЛІК ПОСИЛАНЬ

1. Жильцов А. В., Мірських Г. О. Основи забезпечення якості технічних пристроїв та систем. К: «Політехніка», 2013. 383 с.
2. Бабенко Л. П., Лавріщева К. М. Основи програмної інженерії: Навч. посіб. К.: Т-во «Знання», КОО, 2001. 269 с.
3. Олян В. В., Кравченко О. К. Порівняння моделей життєвих циклів програмного забезпечення з метою виявлення найефективнішого. Системи обробки інформації, 2. 2019. С.63-70.
4. Мірських Г. О., Васюк В. В., Книжка Т. С. Ймовірність в оцінках надійності електроенергетичних комплексів. Енергетика та автоматика. №. 3. 2020. С. 58-77.