

БАЗОВІ ПОНЯТТЯ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

Еталонну модель програмної інженерії можна визначити як взаємодію трьох факторів:

- процесів;
- продуктів;
- ресурсів.

Кожна програмна система протягом свого існування проходить з певною послідовністю фази або стадії від задуму до його втілення в програми, експлуатацію та вилучення. Така послідовність фаз називається *життєвим циклом розробки (Software life cycle processes)*. На кожній фазі відбувається певна сукупність процесів, кожен з яких породжує певний продукт, використовуючи певні ресурси.

Усі продукти всіх процесів програмної інженерії являють собою певні описи — тексти вимог до розробки, погодження домовленостей, документацію, тексти програм, інструкції з експлуатації тощо.

Головними ресурсами програмної інженерії, які визначають ефективність її розробок, є час і вартість цих розробок.

Різновиди діяльності, котрі становлять процеси життєвого циклу програмної системи, зафіксовано в міжнародному стандарті ISO/IEC 12207 : 1995—0801 : Informational Technology - Software life cycle processes [1].

Згідно з наведеним стандартом, усі процеси поділено на три групи:

- головні процеси;
- допоміжні процеси;
- організаційні процеси.

До головних процесів віднесено такі:

- процес придбання, який ініціює життєвий цикл системи та визначає організацію-покупця автоматизованої системи, програмної системи або сервісу;
- процес розроблення, який означає дії організації — розробника програмного продукту;
- процес постачання, який означає дії під час передавання розробленого продукту покупцеві;
- процес експлуатації, який означає дії з обслуговування системи під час її використання — консультації користувачів, вивчення їхніх побажань тощо;
- процес супроводження, який означає дії з керування модифікаціями, підтримки актуального стану та функціональної придатності, інсталяцію та вилучення версій програмних систем у користувача.

У свою чергу, до процесу розроблення входять такі процеси:

- інженерія вимог до системи;
- проектування;
- кодування й тестування.

До допоміжних процесів віднесено ті, що так чи інакше забезпечують якість продукту. Терміном *якість продукту* позначено сукупність властивостей, які зумовлюють можливість задоволення потреб замовника, котрий сформулював їх у формі вимог на розробку.

До організаційних процесів віднесено менеджмент розробки, створення структури організації, навчання персоналу, визначення відповідальності кожного з учасників процесів життєвого циклу розробки.

Стандарт, розглянутий нами вище, є головним чинником визначення змісту діяльності у сфері програмної інженерії, і всі знання, яких потребують професіонали з програмної інженерії, формулюються стосовно процесів, визначених стандартом ISO/IEC 12207:1995 - 0801: Informational Technology - Software life cycle processes.

Зупинимося докладніше на процесах розробки програмного забезпечення, які в сукупності мають забезпечити шлях від усвідомлення потреб замовника до передавання йому готового продукту. На цьому шляху виділяють низку характерних робіт.

Визначення вимог. Збір та аналіз вимог замовника виконавцем і подання їх у нотації, яка є зрозумілою як для замовника, так і для виконавця.

Проектування. Перетворення вимог до розробки в послідовність проектних рішень щодо способів реалізації вимог: формування загальної архітектури програмної системи та принципів її прив'язки до конкретного середовища функціонування; визначення детального складу модулів кожної з архітектурних компонент.

Реалізація. Перетворення проектних рішень на програмну систему, яка реалізує такі рішення.

Тестування. Перевірка кожного з модулів та способів їхньої інтеграції; тестування програмного продукту в цілому (так звана верифікація); тестування відповідності функцій працюючої програмної системи *вимогам (requirements)*, поставленим до неї замовником (так звана валідація).

Експлуатація та супроводження готової програмної системи.

За десятиріччя досвіду з побудови програмних систем напрацьовано низку типових схем послідовності наведених вище робіт. Такі схеми назвали моделями життєвого циклу. Історично першою застосовувалася так звана *водоспадна* або *каскадна модель*, за якою вважалося, що кожна з робіт виконується один раз і в тому порядку, в якому їх перелічено вище. Інакше кажучи, робилося припущення, що кожна з робіт буде виконано настільки ретельно, що після її завершення й переходу до наступної роботи повернення до попередньої не потрібно. На рис. 2.1 показано послідовність робіт за *водоспадною (каскадною) моделлю*. Повернення до початкової стадії робіт передбачається, як бачимо, лише як результат супроводження.

Цінність такої моделі полягає в тому, що вперше було зафіксовано послідовність процесів розроблення та стадії готовності програмного продукту, а недоліком є те, що в її концепцію покладено модель фабрики, коли продукт проходить стадії від задуму до виробництва, після чого передається замовникові як готовий виріб, зміну якого непередбачено, хоча й можлива заміна на інший подібний продукт у разі рекламації.



Рисунок 2.1 - Водоспадна модель життєвого циклу програмних систем

Для програмного продукту така модель не підходить з кількох причин. По-перше, висловлення вимог замовником - це суб'єктивний, неформалізований процес, який, як засвідчує багаторічний досвід, може багаторазово уточнюватися протягом розроблення і навіть після її завершення та випробовування, якщо з'ясується, що замовник "хотів зовсім інше". По-друге, змінюються обставини та умови використання системи, тому загально визнаним законом програмної інженерії є закон *еволюції*, котрий можна сформулювати так: кожна діюча програмна система з часом потребує змін або перестає використовуватися.

Зважаючи на необхідність еволюції, водоспадну модель можна розглядати як модель життєвого циклу лише для першої версії розробки. Враховуючи, що на кожній стадії робіт може виникнути потреба змін, і цю потребу має бути задоволено таким чином, щоб: документація, яка є продуктом кожної стадії (опис вимог, опис проекту тощо), відповідала дійсному стану розробки після внесення змін, було створено так звану спіральну модель розвитку робіт, відміною якої є можливість багаторазового повернення до стадії формулювання вимог до розробки з будь-якої стадії робіт, якщо виявиться необхідність внесення змін.

На рис. 2.2 подано зображення спіральної моделі розробки, в якій кожний виток спіралі відповідає одній з версій розробки. На кожній стадії розроблення аналізується потреба змін, а внесення змін на будь-якій стадії обов'язково починається з внесення змін до попередньо зафіксованих вимог.

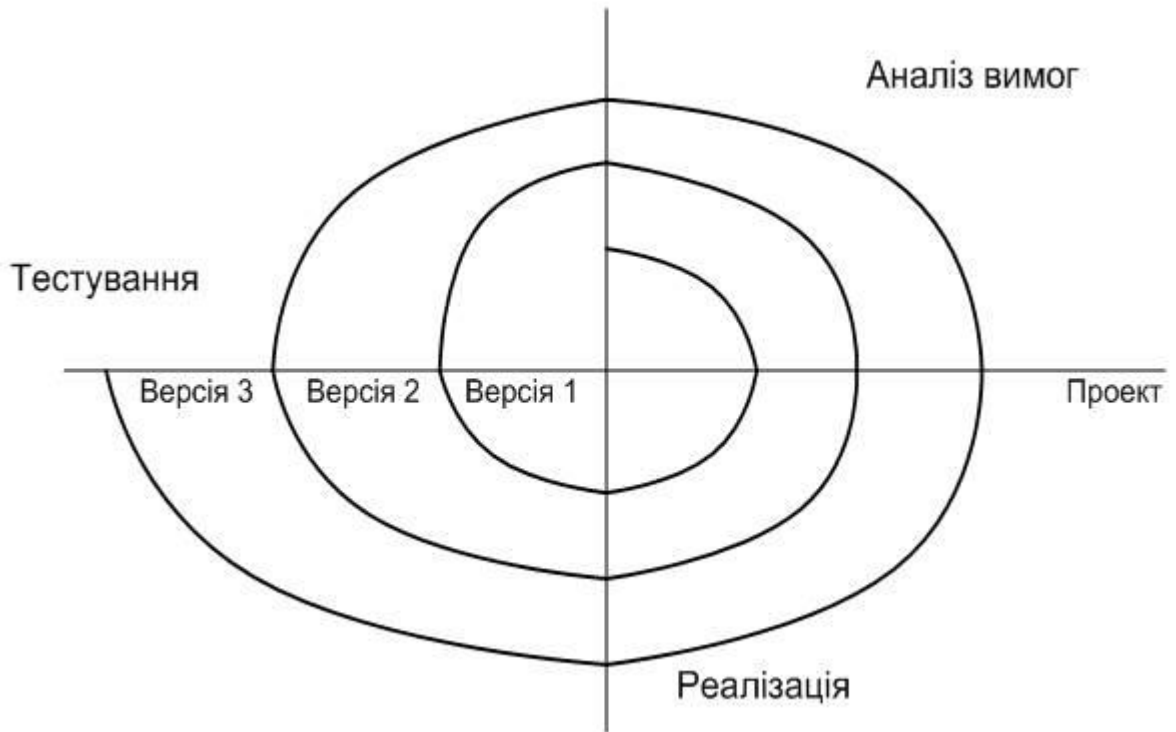


Рисунок 2.2 - Спіральна модель життєвого циклу розробки

Контрольні запитання і завдання

1. Які три чинники визначають еталонну модель програмної інженерії?
2. Який вигляд мають продукти програмної інженерії?
3. Назвіть головні ресурси програмної інженерії, що визначають ефективність розробок програмного забезпечення.
4. Сформулюйте визначення життєвого циклу розробки програмного забезпечення.
5. Назвіть три основні групи процесів життєвого циклу.
6. Перелічіть процеси кожної з груп.
7. Який міжнародний стандарт визначає перелік і зміст процесів життєвого циклу програмного продукту?
8. Чи всі процеси, зазначені в стандарті, має бути виконано в кожній розробці програмного забезпечення або чи надає стандарт такі можливості, які можуть бути актуальними для конкретного випадку?
9. Назвіть етапи процесу розроблення програмного забезпечення.
10. Основні моделі життєвого циклу розробки програмного забезпечення і їх відмінності.