УДК 621.317.1

О. М. Величко, д.т.н., О. В. Грабовський, к.т.н., Т. Б. Гордієнко, д.т.н.*Одеська державна академія технічного регулювання та якості, м. Одеса***ОСОБЛИВОСТІ ЗАСТОСУВАННЯ V-МОДЕЛІ ПРИ РОЗРОБЛЕННІ ТА ОЦІНЮВАННІ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЗАСОБІВ ВИМІРЮВАЛЬНОЇ ТЕХНІКИ**

У статті подано результати дослідження особливостей застосування V-моделі при розробленні та оцінюванні якості програмного забезпечення (ПЗ) засобів вимірювальної техніки (ЗВТ). Проведено аналіз можливості застосування каскадної стратегії розроблення ПЗ та застосування її до оцінювання якості розроблюваного ПЗ. Розглянуто недоліки і переваги V-моделі, яка підтримує каскадну стратегію одноразового виконання етапів розроблення ПЗ. Представлено V-модель розроблення та оцінювання якості ПЗ ЗВТ, яка враховує всі специфічні складові випробування.

Ключові слова: програмне забезпечення, V-модель, тестування, оцінювання якості, засоби вимірювальної техніки.

О. Н. Величко, д.т.н., О. В. Грабовский, к.т.н., Т. Б. Гордиенко, д.т.н.**ОСОБЕННОСТИ ПРИМЕНЕНИЯ V-МОДЕЛИ ПРИ РАЗРАБОТКЕ И ОЦЕНКЕ КАЧЕСТВА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СРЕДСТВ ИЗМЕРИТЕЛЬНОЙ ТЕХНИКИ**

В статье представлены результаты исследования особенностей применения V-модели при разработке и оценке качества программного обеспечения (ПО) средств измерительной техники (СИТ). Проведен анализ возможности применения каскадной стратегии разработки ПО и ее применение к оценке качества разрабатываемого ПО. Рассмотрены недостатки и преимущества V-модели, которая поддерживает каскадную стратегию одноразового выполнения этапов разработки ПО. Представлена V-модель разработки и оценки качества ПО СИТ, которая учитывает все специфические составляющие испытания.

Ключевые слова: программное обеспечение, V-модель, тестирование, оценка качества, средства измерительной техники.

О. М. Velychko, DSc, О. V. Hrabovskiy, PhD, Т. В. Gordiyenko, DSc**FEATURES OF APPLICATION OF V-MODEL IN DEVELOPMENT AND ESTIMATION OF MEASURING INSTRUMENTS SOFTWARE**

The research results of the features of the use of the V-model in the development and evaluation of the quality of software of measuring instruments (MI) were presented. The analysis of the possibility of using a cascade software development strategy and its application to assessing the quality of the software being developed was carried out. It is difficult to implement a classical cascade model in its pure form, therefore, variants of such a model with feedback between its separate stages was developed. The disadvantages and advantages of the V-model, which supports the cascade strategy of one-time execution of software development stages, are considered. The main purpose of the V-model is to provide planning testing software in the early stages of its development. V-model is aimed at simplifying the understanding of the complexity of software development. It is used to determine a single procedure for developing software, hardware and user interface. To reduce the shortcomings of the cascade strategy, a number of modifications of the V-model have been developed due to the type of feedback that provides the ability to modify the results of the previous stages of development. Specific guidelines and recommendations of international and regional organizations for legal metrology have been used to determine the components for quality assessment of MI software. The tasks of developing a MI software on the left side of the letter V begin with the definition of requirements to the MI itself, then to the MI software, and only then as a traditional software development. The V-model of the development and evaluation of the MI software quality is proposed, which allows you to adapt this model to a particular MI at the very beginning of the software development. It takes into account all the specific components of software testing for the MI. The degree of verification of the components of the MI software

depends on the risk classes that are assigned to the software, depending on the MI for which it is created. The V-model of development and quality assessment of software of MI which takes into account all the specific components of the test are presented.

Keywords: *software, V-model, software testing, software estimation, measuring instruments.*

DOI 10.32684/2412-5288-2019-1-14-6-11

Вступ

Сучасні інформаційні технології широко впроваджуються в усі сфери діяльності людини. Обсяги і складність програмного забезпечення ПЗ) різного функціонального призначення постійно зростають, що призводить до появи досить складних методів проектування, аналізу і тестування ПЗ [1, 2].

Знайшли широке використання три основні стратегії розроблення ПЗ: каскадна, інкрементна і еволюційна. Значну популярність здобули два базові принципи розроблення ПЗ: модульний та об'єктно-орієнтований. В США створений і успішно діє Інститут якості ПЗ (Software Quality Institute, SQI) [3], одною із сфер діяльності якого є розроблення та впровадження сучасних методів оцінювання якості ПЗ. На міжнародному рівні ведуться активні роботи зі стандартизації оцінювання якості ПЗ та їх тестування [4, 5].

Актуальним питанням при розробленні та тестуванні ПЗ залишається вибір стратегії та оцінювання якості ПЗ конкретного призначення.

Аналіз публікацій та досліджень

Каскадна стратегія розроблення ПЗ базується на одноразовому проході всіх етапів розроблення ПЗ, а кожен такий етап починається після завершення попереднього. Повернення до вже виконаних етапів не передбачено. Таку стратегію розроблення ПЗ реалізують традиційна каскадна модель і V-подібна модель (далі – V-модель) [1, 6].

Основними перевагами каскадної стратегії розроблення ПЗ є: стабільність встановлених користувачем вимог протягом всього життєвого циклу розроблення ПЗ; простота застосування за рахунок лише одного проходження етапів розроблення ПЗ; простота планування, контролю та управління розробленням ПЗ тощо.

До основних недоліків цієї стратегії слід віднести: складність формулювання необхідних вимог на початковому етапі розроблення ПЗ і неможливість їх зміни у подальшому; лінійність структури процесу розроблення ПЗ; непридатність використання проміжних продуктів у розробленні ПЗ; недостатня участь користувача в процесі розроблення ПЗ тощо.

Інкрементна стратегія розроблення ПЗ базується на багаторазовому проході етапів розроблення ПЗ із запланованим поліпшенням ре-

зультату кожного етапу, який називається інкрементом. Особливістю цієї стратегії є велика кількість етапів розроблення ПЗ при незначній їх тривалості і невеликих відмінностях між інкрементами сусідніх етапів. Сучасною реалізацією такої стратегії є екстремальне програмування [3, 7].

Основними перевагами інкрементної стратегії розроблення ПЗ є: можливість отримання функціонального продукту після реалізації кожного інкремента; коротка тривалість і стабільність вимог під час створення певного інкремента; зниження ризиків у порівнянні з каскадною стратегією; залучення до процесу розроблення ПЗ користувача тощо.

До основних недоліків цієї стратегії слід віднести: необхідність повного функціонального визначення ПЗ на початку життєвого циклу для забезпечення планування інкрементів і управління розробленням ПЗ; складність планування і розподілу робіт з розроблення ПЗ; значний вплив людського фактору при вирішенні складних проблем розроблення ПЗ, що може знизити якість розробленого ПЗ.

Еволюційна стратегія розроблення ПЗ також базується на багаторазовому проході етапів розроблення ПЗ, однак реалізує лише часткове визначення вимог до ПЗ на початковому етапі його розроблення. Вимоги поступово уточнюються в послідовних циклах розроблення ПЗ. Для цієї стратегії характерна істотно менша кількість циклів розроблення ПЗ при більшій тривалості певного етапу в порівнянні з інкрементною стратегією [1, 2].

Основними перевагами еволюційної стратегії розроблення ПЗ є: можливість уточнення і внесення нових вимог в процесі розроблення ПЗ; придатність проміжного продукту для повноцінного використання; можливість управління ризиками; забезпечення широкої участі користувача в процесі розроблення ПЗ; реалізація переваг попередніх стратегій тощо.

До недоліків цієї стратегії слід віднести: невідомість точної кількості необхідних ітерацій (етапів) і складність визначення критеріїв для продовження етапів розроблення ПЗ при наступній ітерації; складність планування і управління розробленням ПЗ; необхідність активної участі користувачів у розробленні ПЗ тощо.

Постановка завдання

Метою дослідження є здійснення аналізу можливостей застосування каскадної стратегії розроблення ПЗ та застосування її до оцінювання якості розроблюваного ПЗ. Результатом цього аналізу повинна стати V-модель розроблення та оцінювання якості ПЗ засобів вимірювальної техніки (ЗВТ).

Методи дослідження

Перші публікації щодо класичної каскадної моделі з'явилися у 1970-х рр., які вперше формалізували структуру етапів розроблення ПЗ. У загальному випадку такі моделі є процесами розроблення ПЗ з різним ступенем деталізації. Наприклад, міжнародним стандартом ISO/IEC 12207 [8] встановлена каскадна модель, яка має умовну назву водоспадної.

Реалізувати класичну каскадну модель у чистому вигляді важко, тому розроблені варіанти такої моделі із зворотними зв'язками між її окремими етапами. Це дає можливість виправлення продуктів попередніх етапів розроблення ПЗ. Однак, до недоліків такого підходу відносяться складності планування та фінансування розроблення ПЗ, досить високий ризик порушення графіка розроблення ПЗ. У міжнародному стандарті ISO/IEC TR 15271 [9] наведено варіант каскадної моделі, орієнтований на колективне розроблення ПЗ.

Концепція V-моделі незалежно розроблена в Німеччині та США у 1980-х рр. [1, 6]. Сучасною версією V-моделі є модель ХТ, затверджена в 2005 р. Основним призначенням V-моделі є забезпечення планування тестування ПЗ на ранніх стадіях його розроблення. V-модель спрямована на спрощення розуміння складнощів розроблення ПЗ. Вона використовується для визначення єдиної процедури розробки ПЗ, апаратного забезпечення та інтерфейсу користувача.

V-модель підтримує каскадну стратегію одноразового виконання етапів розроблення ПЗ і базується на попередньому повному формуванні вимог до ПЗ. Відмінністю V-моделі від традиційної каскадної моделі є те, що в ній виділені зв'язки між етапами розроблення ПЗ і відповідними етапами тестування.

У V-моделі завдання розроблення ПЗ йдуть зверху вниз по лівій стороні літери V, а завдання тестування ПЗ – вгору по правій стороні літери V. Усередині цієї літери V наводяться горизонтальні лінії, що показують, як результати кожного з етапів розроблення ПЗ впливають на розвиток загальної системи тестування ПЗ на кожному з його етапів.

У V-моделі у ході розроблення ПЗ ставляться такі основні завдання: мінімізація ризиків; підвищення та гарантії якості розробленого ПЗ; зменшення загальної вартості розроблення ПЗ; підвищення якості взаємодії між користувачем і розробниками ПЗ. V-модель робить розроблення ПЗ більш прозорим і підвищує якість його контролю шляхом стандартизації проміжних цілей і результатів. Це дозволяє виявляти відхилення при розробленні ПЗ і ризики на його ранніх етапах.

Перевагами V-моделі є: участь користувача у розробці ПЗ і підтримці цієї моделі; на початку розроблення ПЗ модель може бути адаптована саме під це розроблення; модель дозволяє розподілити розроблення ПЗ на окремі етапи, кожен з яких буде включати в себе необхідні для нього дії. V-модель визначає продукти, які повинні бути отримані в результаті кожного з етапів розроблення ПЗ, причому кожен отриманий результат піддають спеціальному тестуванню.

До недоліків V-моделі можна віднести: модель не передбачає роботу з паралельними подіями; у моделі не передбачено внесення вимоги динамічних змін на різних етапах життєвого циклу; у модель не входять дії, спрямовані на аналіз ризиків; деякий результат можна отримати тільки при досягненні низу літери V.

З метою скорочення недоліків каскадної стратегії розроблений ряд модифікацій V-моделі, обумовлених різновидом зворотних зв'язків, які забезпечують можливість зміни результатів попередніх етапів розробки. Ця модель має наступні додаткові переваги: планування оцінювання якості (тестування) на ранніх стадіях розроблення ПЗ; спрощення перевірки проміжних результатів розроблення ПЗ; спрощення управління та контролю етапів розроблення ПЗ.

На рис. 1 наведена базова модель визначення якості розроблюваного ПЗ, яка демонструє зв'язки моделі якості ПЗ за міжнародним стандартом ISO/IEC 25010 [4] та моделлю вимірювання за міжнародним стандартом ISO/IEC 15939 [5].

Якісні властивості ПЗ є невід'ємними його властивостями і можуть бути поділені на одну або декілька характеристик якості (ХЯ), які мають відповідні підхарактеристики якості (ПХЯ). Вони вимірюються відповідним методом, який є логічною послідовністю операцій, що використовуються для кількісного визначення певних властивостей щодо встановленої шкали.

Рисунок 1 – Базова модель визначення якості розроблюваного ПЗ

Характеристики якості і ПХЯ можна визначити кількісно, застосовуючи функції вимірювання – алгоритм, що використовується для об'єднання елементів вимірювання якості. Результатом застосування функції вимірювання є показники якості (ПЯ) ПЗ, тобто кількісні ХЯ і ПХЯ. Для вимірювання ХЯ можна використовувати більше ніж один ПЯ ПЗ, які складаються з певних елементів, або ПХЯ.

На рис. 2 наведена загальна V-модель розроблення та оцінювання якості ПЗ. У цій моделі завдання розроблення ПЗ йдуть зверху вниз по лівій стороні літери V, а завдання оцінювання якості ПЗ – вгору по правій стороні літери V. У середині цієї літери V наведені горизонтальні лінії, що показують, як перевіряються результати кожного з етапів розроблення ПЗ за програмою випробувань (ПВ). ПЗ приймається у випадку реалізації всіх завдань ПВ ПЗ.

Доцільно застосувати загальну V-модель розроблення та оцінювання якості ПЗ для ПЗ ЗВТ.

Результати досліджень

У працях [10, 11] описані особливості оцінювання якості ПЗ для ЗВТ та наведені основні складові для оцінювання якості ПЗ ЗВТ. При визначенні складових для оцінювання якості ПЗ ЗВТ використані спеціальні настанови та рекомендації міжнародних та регіональних організацій з питань законодавчої метрології [12-15].

На рис. 3 наведена пропонувана V-модель розроблення та оцінювання якості ПЗ ЗВТ. Завдання розроблення ПЗ ЗВТ по лівій стороні літери V починаються з визначення вимог до самого ЗВТ, потім до ПЗ ЗВТ, а вже потім як традиційного розроблення ПЗ (як на рис. 2).

При формуванні вимог до ПЗ ЗВТ повинні враховуватися як загальні вимоги до ПЗ, вбудованого у ЗВТ (Р), так і загальні вимоги до ПЗ, встановленого на універсальних комп'ютерах (U). Це залежить від особливостей самого ЗВТ. Крім того, особливістю ПЗ ЗВТ є також необхідність встановлення рівня розділення ПЗ (S) на законодавчо значиму та інші частини.

Завдання оцінювання якості ПЗ по правій стороні літери V починаються з перевірки зчитування (D) ПЗ ЗВТ, а вже потім перевірок запам'ятовуючих пристроїв (L) та пристроїв передачі даних (T). Всі ці перевірки повинні бути передбачені ПВ ПЗ для ЗВТ.

Лише при успішному завершенні всіх необхідних перевірок ПЗ ЗВТ, можна стверджувати про завершення загальної оцінки якості ПЗ ЗВТ. Випробування ПЗ ЗВТ є складовою приймальних випробувань самого ЗВТ, за результатами якого це ЗВТ приймається.

Ступінь перевірки складових ПЗ ЗВТ залежить від класів ризиків, які присвоюються ПЗ залежно від ЗВТ, для якого воно створюється.

Висновки

Аналіз можливостей застосування каскадної стратегії розроблення ПЗ та застосування її до оцінювання якості розроблюваного ПЗ показав таку доцільність.

Запропонована V-модель розроблення та оцінювання якості ПЗ ЗВТ дозволяє вже на початку розроблення ПЗ адаптувати цю модель під конкретне ПЗ для конкретного ЗВТ. При цьому враховуються всі специфічні складові випробування ПЗ для ЗВТ.

Рисунок 2 – Загальна V-модель розроблення та оцінювання якості ПЗ

Рисунок 3 – V-модель розроблення та оцінювання якості ПЗ ЗВТ

Список використаних джерел

1. Бахтизин В. В. Технология разработки программного обеспечения: учеб. пособие / В. В. Бахтизин, Л. А. Глухова. – Минск: БГУИР, 2010. – 267 с.

2. Software Development Methodology Today. Software Development Strategies and Life-Cycle Models [Електронний ресурс]. – 2008. – Режим доступу: <http://www.informit.com/articles/article.aspx?p=605374&seqNum=2>.

3. Software Quality Institute, SQI [Електронний ресурс]. – 2019. – Режим доступу: <https://isqi.org/ua/en/>.
4. ДСТУ ISO/IEC 25010:2015 Інженерія програмних засобів і систем. Вимоги щодо якості та оцінювання систем і програмного продукту (SQuaRE). Моделі якості системи та програмних засобів (ISO/IEC 25010:2011, IDT).
5. ДСТУ ISO/IEC 15939:2008 Інженерія систем і програмних засобів. Процес вимірювання (ISO/IEC 15939:2007, IDT).
6. Фатрелл Р., Шафер Д., Шафер Л. Управление программными проектами: достижение оптимального качества при минимуме затрат. – М.: Вильямс, 2003. – 1125 с.
7. Брауде Э. Технология разработки программного обеспечения. – СПб.: Питер, 2004. – 655 с.
8. ISO/IEC 12207:2008 Systems and software engineering – Software life cycle processes.
9. ISO/IEC TR 15271:1998, Software Engineering – Software life cycle processes – Guide for ISO/IEC 12207 (Software Life Cycle Processes).
10. Velychko O., Gordiyenko T., Hrabovskiy O. Testing of measurement instrument software on the national level // Eastern-European Journal of Enterprise Technologies. Information and controlling systems. – 2018. – № 2/9(92). – P. 13–20.
11. Velychko O., Gaman V., Gordiyenko T., Hrabovskiy O. Testing of measurement instrument software with the purpose of conformity assessment // Eastern-European Journal of Enterprise Technologies. Information and controlling systems. – 2019. – № 1/9(97). – P. 19–26.
12. OIML D 31:2008. General Requirements for Software Controlled Measuring Instruments. OIML. Paris, 2008. – 53 p.
13. WELMEC 7.1. Informative Document: Development of Software Requirements.
14. WELMEC 7.2. Software Guide (Measuring Instruments Directive 2004/22/EC).
15. WELMEC 2.3. Guide for Examining Software (Non-automatic Weighing Instruments).
16. COOMET R/LM/10:2004. COOMET Recommendation: Software for Measuring Instruments: General Technical Specifications. COOMET. 2004. – 10 p.
- Software Development Strategies and Life-Cycle Models [Електронний ресурс]. – 2008. – Режим доступу: <http://www.informit.com/articles/article.aspx?p=605374&seqNum=2>.
3. Software Quality Institute, SQI [Електронний ресурс]. – 2019. – Режим доступу: <https://isqi.org/ua/en/>.
4. DSTU ISO/IEC 25010:2015 Inzheneriya programnykh zasobiv i system. Vymogy shchodo yakosti ta otsinyuvannya system i programnogo produktu (SQuaRE). Modeli yakosti systemy ta programnykh zasobiv (ISO/IEC 25010:2011, IDT). (Ukrainian)
5. DSTU ISO/IEC 15939:2008 Inzheneriya system i programnykh zasobiv. Proces vymiryuvannya (ISO/IEC 15939:2007, IDT). (Ukrainian).
6. Fatrell R., Shafer D., Shafer L. Upravlenie programnymi proektami: dostizhenie optimal'nogo kachestva pri minimume zatrat. – М.: Vil'yams, 2003. – 1125 s. (Russian).
7. Braude E'. Texnologiya razrabotki programnogo obespecheniya. – SPb.: Piter, 2004. – 655 s. (Russian).
8. ISO/IEC 12207:2008 Systems and software engineering – Software life cycle processes.
9. ISO/IEC TR 15271:1998, Software Engineering – Software life cycle processes – Guide for ISO/IEC 12207 (Software Life Cycle Processes).
10. Velychko O., Gordiyenko T., Hrabovskiy O. Testing of measurement instrument software on the national level // Eastern-European Journal of Enterprise Technologies. Information and controlling systems. – 2018. – № 2/9(92). – P. 13–20.
11. Velychko O., Gaman V., Gordiyenko T., Hrabovskiy O. Testing of measurement instrument software with the purpose of conformity assessment // Eastern-European Journal of Enterprise Technologies. Information and controlling systems. – 2019. – № 1/9(97). – P. 19–26.
12. OIML D 31:2008. General Requirements for Software Controlled Measuring Instruments. OIML. Paris, 2008. 53 p.
13. WELMEC 7.1. Informative Document: Development of Software Requirements.
14. WELMEC 7.2. Software Guide (Measuring Instruments Directive 2004/22/EC).
15. WELMEC 2.3. Guide for Examining Software (Non-automatic Weighing Instruments).
16. COOMET R/LM/10:2004. COOMET Recommendation: Software for Measuring Instruments: General Technical Specifications. COOMET. 2004. 10 p.

References

1. Baxtizin V. V. Texnologiya razrabotki programnogo obespecheniya: ucheb. posobie / V. V. Baxtizin, L. A. Gluxova. – Minsk: BGUIR, 2010. – 267 s. (Russian).
2. Software Development Methodology Today.

Надійшла до редакції 23.04.2019