

2. Галимова Е.Ю. Методика выбора автоматизированного, ручного и смешанного способа тестирования программного продукта, основаная на критериях качества. *Известия ТулГУ. Технические науки*. 2019. № 7. С. 248-256.
3. Галимова Е.Ю., Коваленко А.Н. Метод выбора между ручным и автоматизированным тестированием, основанный на свойствах программного продукта. *Вестник Донского государственного технического университета*. 2016. № 4(87). С. 134-139.
4. Milad Hanna, Amal Elsayed Aboutabl, Mostafa-Sami M. Mostafa Automated Software Testing Frameworks: A Review. *International Journal of Computer Applications*, 2018. Vol. 179. No.46. P. 22-28.

Нагорний П.В., студент групи ПІ-181

Національний університет «Чернігівська політехніка», inn5665@gmail.com

Науковий керівник: Злобін С.В., викладач циклової комісії дисциплін професійної підготовки спеціальностей 122, 123

ВСП «Фаховий коледж транспорту та комп'ютерних технологій» Національного університету «Чернігівська політехніка», s.zlobin75@gmail.com

ФОРМАЛІЗАЦІЯ ВИМОГ ЯК ЗАСІБ ПІДВИЩЕННЯ ЯКОСТІ ТЕСТУВАННЯ

Невід'ємний етап розробки будь-якого програмного забезпечення (ПЗ) – аналіз та розробка вимог. Міжнародна комісія з кваліфікацій тестування програмного забезпечення ISTQB визначає вимоги як «умови або можливості, необхідні користувачу для вирішення певних задач чи досягнення конкретних цілей, які мають бути досягнені для виконання контракту, стандартів, специфікації або інших формальних документів...» [1]. Фактично, на визначених вимогах ґрунтується як подальша розробка, так і тестування ПЗ. Саме тому дослідження особливостей формулювання та конкретизації вимог є актуальним у сфері тестування ПЗ.

Існує прямий зв'язок між якістю сформульованих вимог та ефективністю подальшої розробки ПЗ. Так формулювання високоякісних вимог призводить до

- ❖ пришвидшення процесу розробки та тестування;
- ❖ полегшеного планування та розподілу часу;
- ❖ реалізації тільки потрібних фрагментів коду;
- ❖ зменшення кількості дефектів як у самих вимогах, так і в розробленому кодї.

Водночас формулювання неякісних вимог може призводити до появи певних негативних наслідків під час тестування. Такі наслідки систематизовано в таблиці 1.

Отже, неякісно сформульовані вимоги здатні дуже негативно впливати на процес тестування. Проте, існує підвищення якості тестування – формалізація вимог [4]. За своєю суттю формалізація вимог є представленням вербальної форми подання вимог у формальній, математично обґрунтованій. Незважаючи на певну простоту означення, процес формалізації є досить складним. Єдиного алгоритму формалізації умов немає. Ю. Липко в [4] пропонує наступну послідовність дій:

- трансформація вербальних речень у таблиці за граматичною структурою (підмет, присудок, додаток тощо);
- заповнення пустих комірок у відповідності до встановлених правил;
- формування листу акторів – осіб, що взаємодіють з ПЗ;
- визначення в листі акторів порядку дій;
- проектування графу елементів вимог.

Головний недолік описаного алгоритму – використання графів, що часто буває досить незручно, особливо в цілях аналізу тексту. В [3] пропонується інший варіант формалізації вимог – у вигляді опису полів, призначених для вводу даних. В загальному вигляді формальний опис такого поля може виглядати наступним чином:

"Поле - загальний опис": {

```

"Тип": "поле",
"Група": "назва_групи",
"Залежне": "статус_залежності",
"Обов'язкове": "статус_обов'язковості",
"Формат": "тип_поля",
"Значення поля": "значення_поля"
}
    
```

Таблиця 1. Наслідки формулювання неякісних вимог для тестування ПЗ

Ризик	Характеристика	Наслідки для тестування
Недостатня залученість користувачів	Замовник вважає, що в достатній мірі знає потреби користувачів, тому не приділяє достатньої уваги аналізу вимог користувачів.	Тестувальник приймає за зразок поставлені перед ПЗ вимоги, тому може схвалити неякісний з точки зору користувачів продукт. Якщо ж тестувальник поверне ПЗ на доопрацювання – це приведе до додаткових втрат ресурсів та часу.
Поступове збільшення обсягу вимог	Перелік сформульованих вимог поступово розростається, потребуючи при цьому все більше ресурсів, часу, коштів бюджету.	Перелік сформульованих вимог поступово розростається, потребуючи при цьому все більше ресурсів, часу, коштів бюджету.
Двозначність сформульованих вимог	Вимоги сформульовані таким чином, що у різних людей, які ознайомилися з переліком вимог, будуть різні очікування стосовно роботи ПЗ.	Якщо тестувальник зрозуміє вимоги так, як і розробник, деякі дефекти, важливі для користувача, можуть бути не виправлені. В іншому випадку будуть потрібні додаткові ресурси на серйозні виправлення.
«Позолота» продукту	Розробники модифікують ПЗ, додаючи до нього деякі функції, які відсутні у вимогах, бо вважають, що так краще.	Тестувальник витрачає більше запланованого часу на тестування нових непотрібних функцій ПЗ.
Мінімальна специфікація	Розроблено специфікацію, яка містить не конкретні чіткі вимоги, а лише їх скорочені нариси, за якими не можна чітко встановити очікуваного режиму	Тестувальник змушений консультуватися у розробника стосовно призначення тих чи інших функцій.
Недбале планування	Розподіл часу та інші форми планування відбуваються завчасно, до аналізу вимог.	Розподіл часу та інші форми планування відбуваються завчасно, до аналізу вимог.

Джерело: розроблено автором на основі [2-3]

Такий вид опису поля є більш зручним, передбачає можливість відзначення залежностей між окремими полями, легкого розширення за умови зміни програми тощо.

Таким чином, існує безпосередній зв'язок між якістю сформульованих вихідних вимог та ефективністю процесу тестування. Наслідки таких ризиків формулювання вимог, як недостатня залученість користувачів, поступове збільшення обсягу вимог, двозначність сформульованих вимог, «позолота» продукту, мінімальна специфікація, недбале планування, досить серйозні для процесу тестування. Саме тому запропоновано здійснювати формалізацію вимог – передавати її зміст в чіткій конкретній оброблюваній формі. Існують різні підходи до формалізації вимог, кожен з підходів має свої переваги та недоліки, але є застосовним у деяких практичних задачах.

Перелік посилань

1. Глосарій Міжнародної комісії з кваліфікацій тестування програмного забезпечення ISTQB. URL: <http://glossary.istqb.org/> (дата звернення: 06.12.2021).

2. Флейта Ю.В., Штанько О.В. Вплив формалізації вимог на процес тестування програмного забезпечення. *Інфокомунікації - сучасність та майбутнє* : зб. матеріалів X Міжнар. наук.-практ. конф. (Одеса, 16-19 листопада). Одеса. 2020. С. 426-428.
3. Бахвалова З.А., Камышова Е.А. Роль формализации требований в тестировании программного обеспечения. *Информационные и математические технологии в науке и управлении*. 2020. № 1(17). С. 120-129.
4. Липко Ю.Ю. Алгоритм формализации требований при разработке информационных систем. *Известия Южного федерального университета. Технические науки*. 2014. № 6(155). С. 153-158.

Нагорний П.В., студент групи ПІ-181

Науковий керівник: Богдан І.В., канд. техн. наук

Національний університет «Чернігівська політехніка», irakirienko@gmail.com

СУЧАСНІ ЗАСОБИ АВТОМАТИЗАЦІЇ ТЕСТУВАННЯ

Одна з найважливіших властивостей програмного забезпечення (ПЗ) в сучасних умовах – якість, яка визначається як відповідність ПЗ очікуваному режиму роботи. Забезпечення належної якості складає окрему складову розробки ПЗ. Основний механізм забезпечення якості – пошук та виправлення наявних проблем. Пошук проблем здійснюється шляхом тестування. Тому дослідження та покращення ефективності тестування є актуальним для сучасної сфери ІТ.

Один із сучасних трендів тестування – автоматизація. Дійсно, важко віднайти хоча б одну сферу діяльності людини, яка за останній час не характеризувалася хоча б певним рівнем діджиталізації та автоматизації. Тим паче, сфера, яка сама пов'язана з цифровими технологіями. Автоматизація дозволяє скоротити, вдосконалити процес тестування. В той же час автоматизація є досить ресурсоємкою, саме тому для малих проєктів досить часто зупиняються на ручному чи змішаному тестуванні. Однак для великих проєктів автоматизація тестування є незамінною [1].

На ринку ПЗ на даний момент представлено значну кількість різноманітних засобів автоматизації тестування. Кожен з таких засобів вирізняється своїми особливостями, перевагами та недоліками. В [2] виділяють такі найефективніші сучасні засоби автоматизації тестування: Selenium, Katalon Studio, UFT One, TestComplete, SoapUI, Postman. За даними [2-4] систематизуємо особливості кожного із засобів та наведемо в таблиці 1 порівняння зазначених засобів за їх основними сутнісними характеристиками.

Кожен з розглянутих вище інструментів має свої переваги та недоліки. Перерахуємо головні серед них [3-4].

Основними перевагами Selenium є підтримка великої кількості платформ, мов програмування, браузерів, та безкоштовний доступ. До основних недоліків Selenium можна віднести необхідність значного досвіду програмування для ефективного використання та тестування лише веб-застосунків.

Основними перевагами Katalon Studio є легкість інсталяції та використання, а також підтримку тестування значної кількості типів ПЗ. До основних недоліків Katalon Studio можна віднести підтримку малої кількості мов програмування, відсутність можливості паралельного запуску тестів, відсутність підтримки деталізованих звітів.

Основними перевагами UFT One є спрощений запис скриптів, оновлення інтерфейсу з мінімальними змінами скриптів. Основними недоліками UFT One є підтримка тільки VBScript та Windows, платний доступ.

Основними перевагами TestComplete є підтримка великої кількості мов програмування, тестування різнотипних ПЗ. До основних недоліків TestComplete можна віднести велику вартість, можливість роботи тільки на платформі Windows.