Катаєва Є.Ю.,
К.т.н., доцент кафедри програмного забезпечення автоматизованих систем
Черкаського державного технологічного університету
Одокієнко С.М.,
К.т.н., доцент кафедри інформаційно-комп'ютерних
технологій та фундаментальних дисциплін
Київського національного університету технологій та дизайну
Савченко Я.С.,
Магістр кафедри інформаційно-комп'ютерних
Технологій та фундаментальних дисциплін
Київського національного університету технологій та дизайну

АКТУАЛЬНІСТЬ РОЗРОБКИ АВТОТЕСТУ ВИЗНАЧЕННЯ ЯКОСТІ ПРОГРАМНОГО ПРОДУКТУ ЗА ДОПОМОГОЮ МЕТРИК АНАЛІЗУ КОДУ

Успіх будь-якого проекту визначається його здатністю задовольнити потреби споживача, а тому забезпечення високого рівня якості є необхідним завданням будь-якого виробництва, в тому числі програмної інженерії. Недостатня якість створюваного ПО потребує багато IT-організації, до 70% бюджету інформаційної системи резервувати на етап супроводу, при цьому до 60% всіх модифікацій ПО виконується для усунення помилок, а тільки решту 40% - для корекції ПО в рамках бізнес-процесу, вдосконалення тих чи інших показників якості ПЗ, або для запобігання потенційних проблем.

Якість ПЗ - поняття комплексне. Стандарти виділяють якість процесів розробки, внутрішню і зовнішню якість програмного продукту, якість програмного продукту на стадії використання. Для кожного з компонентів якості можна назвати набір метрик, що визначають якість програмного продукту.

Отримана структура називається моделлю якості програмного забезпечення. Метрика програмного забезпечення - це захід, що дозволяє отримати чисельне значення деякої властивості програмного забезпечення або його специфікацій, а також метод її підрахунку. Метрики дозволяють отримати чисельні значення кожної властивості програмного забезпечення або його специфікацій. Особливий інтерес представляють метрики складності програмного забезпечення. Складність є важливим фактором, від якого залежать інші параметри якості ПО, такі як точність, коректність, надійність, зручність супроводу. Існування методів і алгоритмів автоматичного розрахунку метрик складності ПО за допомогою програмних засобів дозволяє отримати комплексний формальний звіт про якість ПЗ за короткий час. Це дозволяє проводити об'єктивний моніторинг рівня якості ПЗ протягом всього життєвого циклу проекту, вносити корективи в план проекту, а також своєчасно приймати рішення про необхідності проведення рефакторингу.

Розмірно-орієнтовані метрики прямо вимірюють програмний продукт і процес його розробки. Найпоширенішою метрикою вихідного коду ПЗ, що відображає розмір програмного проекту, є показник кількості рядків коду (Source Lines Of Code, SLOC).

Досить практичними є метрики програмного забезпечення, запропоновані Томасом Джилбі (Thomas Jilb) заснованих на результатах аналізу текстів програмних продуктів.

Мірою складності розуміння програм на основі вхідних і вихідних даних є метрика Н. Чепіна (Ned Chapin).

В ході аналізу середовищ для написання автотестів, було обрано середовище Selenium IDE. Selenium IDE (інтегроване середовище розробки) - просте у використанні розширення для браузера, яке допомагає розробляти тестові сценарії веб-сторінок. Також було обрано Jenkins - інструмент безперервної інтеграції з відкритим вихідним кодом, написаний на Java.

Тести оцінюються не тільки з точки зору успішності виконання, але і по тестовому покриттю вихідного коду. SonarQube дозволяє аналізувати багатомовні проекти, хоча для кожної мови робиться свій аналіз.

Для реалізації і запуску автотесту було обрано веб-браузер Mozilla Firefox і встановлено розширення Selenium IDE. Відкривши середовище Selenium є можливість відразу створити новий проект і почати записувати дії для виконання.

Було поставлене завдання, розробити програмний продукт розрахунку метрик якості програмного коду, за допомогою інструментів, описаних вище.

Для початку необхідно в режимі запису дій в Jenkins відкрити проект і зібрати нову збірку з останніми змінами в коді, дочекатися поки виконається збірка і перейти по номеру останньої її версії. Потім необхідно відкрити в консоль і перейти за посиланням на SonarQube, де можна відкрити і побачити в деталях результати сканування проекту (Рисунок 1).

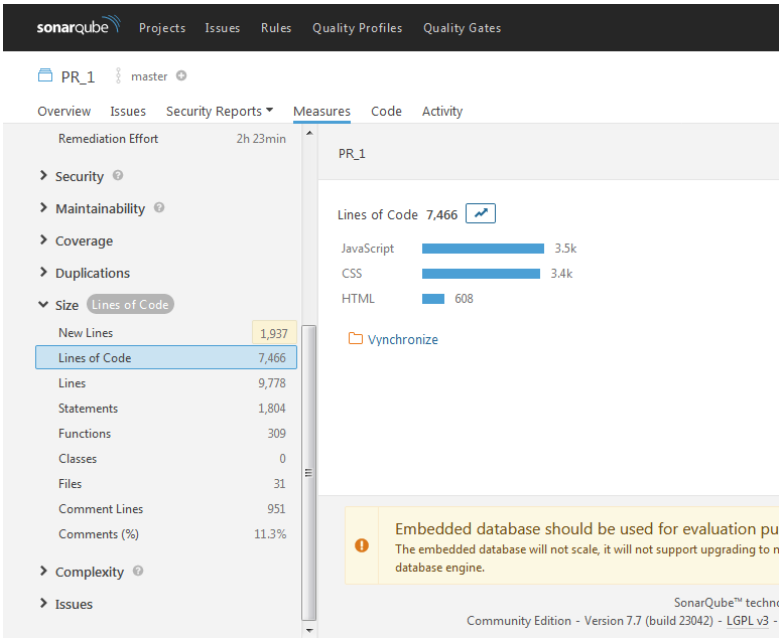

Рисунок 1 – **Вікно з результатами сканування проекту в SonarQube**

Тепер необхідно скопіювати необхідні дані, для запису їх в код автотесту. Для цього потрібно увімкнути інспектор в інтерфейсі вікна Selenium і вибрати дані які потрібно зберегти, для їх подальшої обробки. Після вибору даних на сайті через інспектор, в поточний крок в коді автотесту записується елемент коду сайту в якому знаходяться необхідні дані і при кожному наступному запуску автотесту на даному кроці будуть копіюватися нові дані які будуть відображатися в даному елементі коду сайту.

Після збереження в Selenium необхідних даних потрібно задати перехід на сторінку, на якій знаходиться форма для вводу збережених даних, щоб провести обчислення над цими даними за формулою відповідно до обраної метрики. В поля для вводу потрібно ввести раніше збережені дані із SonarQube, такі як:

- Продуктивність;
- Якість;
- Питома вартість;
- Документованість.

Для обчислення даних за результатами аналізу коду проекту в SonarQube було обрано SLOC-метрику.

Отримані дані з SonarQube можна обчислити за формулою SLOC-метрики. Ці дані, які потрібно записати в змінні в Selenium IDE, щоб їх можна було підставити в форму для обчислення даних. Передачу даних в поля для вводу можна зробити ідентифікувавши кожне поле ідентифікатором і передавати відповідні дані у відповідні поля. Щоб запустити функцію яка буде проводити обчислення даних з полів на формі, потрібно створити кнопку для обчислення і назначити на неї подію, яка буде запускати функцію обчислення кожного разу після натиснення на кнопку.

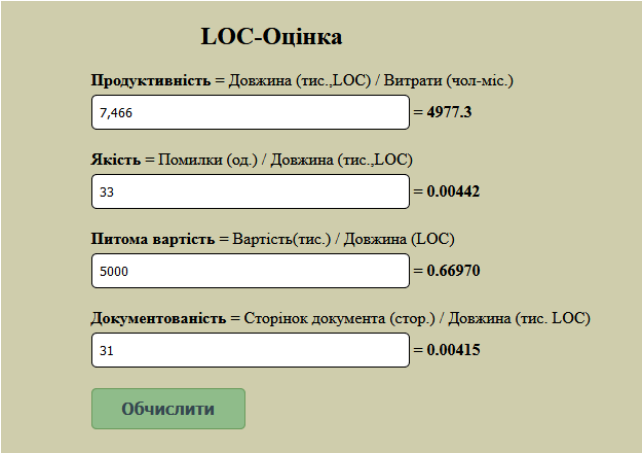

Рисунок 2 – **Результати обчислень за формулою SLOC-метрики**

Отже можна зробити висновок, що за допомогою розробленої системи можна буде обраховувати метрики, які будуть представляти якість програмного коду.

Література

1. Системный контекст программного обеспеченияСистемный контекст программного обеспечения Рівен доступу - https://stepik.org/lesson/106620/step/1?unit=81144

2. Принципі тестирования – Режим доступу - https://qalight.com.ua/baza-znaniy/pochemu-testirovanie-neobhodimo/

3. Автоматизация тестирования методом программніх приложений https://www.dissercat.com/content/avtomatizatsiya-testirovaniya-programmnykh-prilozhenii-metodom-klyuchevykh-sostoyanii

Кусий Я.М.,
к.т.н., доцент кафедри технології машинобудування
Національного університету «Львівська політехніка»
Личак О.В.,
к.т.н., с.н.с. відділ методів і засобів відбору та обробки діагностичних сигналів № 9
Фізико-механічного інституту ім. Г.В.Карпенка НАН України, м. Львів
Топільницький В.Г.,
к.т.н., доцент кафедри проектування та експлуатації машин
Національного університету «Львівська політехніка»

УДОСКОНАЛЕННЯ СТРУКТУРНОЇ МОДЕЛІ ЖИТТЄВОГО ЦИКЛУ ВИРОБУ ІЗ ВРАХУВАННЯМ ТЕХНОЛОГІЧНОГО УСПАДКОВУВАННЯ ПАРАМЕТРІВ МАТЕРІАЛУ

Забезпечення експлуатаційних характеристик виробів із дотриманням призначених конструктором параметрів точності та якості поверхневих шарів їх виконавчих поверхонь є пріоритетним завданням сучасних технологій машинобудівного виробництва. Експлуатаційні умови роботи деталей машин характеризуються, як правило, нестаціонарними багатопараметричними термосиловими та хімічними впливами, що спричинюють деградацію відправних властивостей міцності елементів конструкцій та, як наслідок, вичерпання ресурсу матеріалу виробів [1-4].

Технологічне успадковування при виготовленні деталей машин розглядається як сукупність складних фізичних явищ перенесення залежних один від одного параметрів якості виробу від попередніх до наступних технологічних операцій. Технологічне успадковування властивостей/параметрів доцільно