5. Висновок

Покриття коду є важливою метрикою для забезпечення якості тестованого програми, особливо якщо мова про проекти зі складною логікою і великим обсягом коду. Аналіз покриття коду виконується за допомогою спеціального інструментарію, який дозволяє простежити в які рядки, гілки і т.д. коду, були входження під час роботи Автотест. Найбільш відомі інструменти для проведення вимірювання покриття коду: AQTime, Bounds Checker, Bullseye Coverage, Coverage Meter, Clover, NCover, IBM Rational PurifyPlus, Intel Compiler, Intel Code Coverage Tool Prototype, JetBrains. За допомогою аналізу покриття коду можна оцінити густину покриття авто-тестами виконуваного коду тестованого програми (можна відповісти на питання який обсяг тестування ми (наші Автотест) виконуємо?). При детальному аналізі результатів покриття коду Автотест можна оцінити покриття окремих компонентів системи (тобто можна відповісти на питання: що і в якому обсязі ми тестуємо ?, в яких місцях потрібно оптимізувати покриття ?, які місця системи не перевіряються тестами? І т. д.). Таким чином, знаючи дану метрику, стане ясно для яких тестових випадків потрібно створити нові тести, або прибрати дублюючі тести. Дані заходи допоможуть збільшити значення метрики Code Coverage, що в свою чергу має підвищити якість коду і якість тестованого програми в цілому. Природно, чим вище показник даної метрики - тим краще, проте вже добре якщо у вас покриті тестами найбільш складні і важливі фрагменти коду.

6. Література:

1. Борзов Ю. В., Уртанс Г. Б., Шимаров В. А. Выбор путей программы

для построения тестов УСиМ. – 2013. – с. 29–36.

2. Брукс Ф. Мифический человеко-месяц или как создаются

программные системы СПб.: Символ-Плюс, 2015. – 304 с.

3. Канер С., Фолк Дж., Тестирование программного

обеспечения К: ДиаСофт, 2013 – 544 с.

4. Котляров В.П. Основы тестирования программного обеспечения

Интернет-университет информационных технологий; 2012.

5. Майерс Г., Баджетт Т., Сандлер К. Искусство тестирования

программ, 3-е издание. – М.: «Диалектика», 2012. – 272 с.

6. Макгрегор Дж, Сайкс Д. Тестирование объектно-ориентированного

программного обеспечения К: Диасофт, 2015. – 432