УДК 004.054

С.А. ВИЛКОМИР

Университет Восточной Каролины, США

ПОДХОДЫ К СРАВНЕНИЮ КРИТЕРИЕВ ТЕСТИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Для тестирования программного обеспечения применяются различные стратегии (критерии) тестирования. Наиболее важным свойством таких критериев является их эффективность, т.е. способность обнаруживать дефекты в программном обеспечении. В статье дается обзор современных работ по определению и сравнению эффективности различных критериев тестирования. Показано, что наряду с эффективностью, также важна устойчивость критерия, которая определяется диапазоном распределения эффективности для различных тестовых наборов, удовлетворяющих одному и тому же критерию. Приводятся примеры экспериментальной оценки устойчивости.

Ключевые слова: *программное обеспечение, тестирование, критерии, эффективность, устойчивость.*

Введение

Различные стратегии тестирования программного обеспечения часто формулируются как *критерии тестирования*, т.е. правила, согласно которым генерируются тестовые наборы. Для правильного выбора критерия тестирования в той или иной практической ситуации необходимо использовать методы оценки и сравнения свойств критериев. Одним из важных свойств критериев тестирования является их *эффективность*, т.е. способность обнаруживать дефекты в программном обеспечении. Другим важным и значительно менее изученным свойством является устойчивость критерия, которая определяется диапазоном распределения эффективности для различных тестовых наборов, удовлетворяющих одному и тому же критерию. Данная статья рассматривает эффективность и устойчивость критериев тестирования программного обеспечения и состоит из трех разделов. В разделе 1 рассматривается понятие критерия тестирования и приводятся определения некоторых, как хорошо известных, так и недавно предложенных критериев. В разделе 2 дается обзор современных работ по определению и сравнению эффективности различных критериев тестирования. Раздел 3 посвящен устойчивости критериев тестирования: дается определение устойчивости; показано, почему важно учитывать устойчивость при выборе критерия; приводятся примеры экспериментальной оценки устойчивости. В заключении рассмотрены перспективные направления для дальнейших исследований в данной области.

1. Критерии тестирования программного обеспечения

Критерии тестирования можно рассматривать с различных точек зрения, например, как правила определения момента прекращения тестирования или как меры качества тестирования [1]. В данной статье критерии тестирования понимаются как определения различных стратегий тестирования, т.е. как правила, которым должны соответствовать тестовые наборы. При этом обычно считается достаточным использовать лишь один тестовый набор, удовлетворяющий данному критерию.

Простейшие критерии тестирования изучаются и используются на практике начиная с 70-х годов прошлого столетия, например [2]:

- покрытие ветвей (Decision Coverage – DC) - в ходе тестирования каждое логическое выражение должно принимать и значение «истина», и значение «ложь»;
- покрытие условий (Condition Coverage – CC) в ходе тестирования каждое элементарное логическое условие в логическом выражении должно принимать и значение «истина», и значение «ложь».

Более сложным является критерий модифицированного покрытия условий и ветвей (Modified Condition/Decision Coverage – MC/DC) [3-5], который дополнительно требует, чтобы влияние каждого элементарного условия на значение всего логического выражения было независимо от остальных условий. Как дальнейшее развитие MC/DC, был предложен критерий усиленного покрытия условий и ветвей (Reinforced Condition/Decision Coverage – RC/DC) [6-8]. RC/DC дополнительно требует, чтобы были протестированы ситуации, когда изменения значений элементарных условий не влияют на значение логического выражения.

Все упомянутые критерии принадлежат группе критериев тестирования потоков управления (control-flow criteria). Используются также критерии тестирования потока данных (data-flow), комбинаторные критерии [9] и другие подходы.

2. Эффективность критериев тестирования

Оценка эффективности критериев тестирования рассматривалась в [10-16] и многих других работах. Различные объекты были предметом экспериментальных исследований: большие программные комплексы [17], программы ограниченного (малого) размера [18], отдельные логические выражения [19] и т.п. Наряду с реальными дефектами программного обеспечения рассматривались искуственно сгенирированные дефекты различных типов [20].

Для оценки эффективности критериев тестирования во многих работах использовались E-мера – ожидаемое число обнаруженных дефектов, и P-мера – вероятность обнаружения по крайней мере одного дефекта [21]. Новая мера эффективности, число тестов, необходимых для обнаружения первого дефекта (F-мера), была рассмотрена в [22]. Математический анализ верхней границы эффективности тестирования для всех трех мер предложен в [23]. Авторы определяют теоретический максимум эффективности для целого класса стратегий тестирования и затем сравнивают его с эмперически определяемой эффективностью отдельных стратегий.

Две различных меры эффективности критериев тестирования были предложены в [24] для критериев, основанных на субдоменах (subdomain-based), т.е. делящих область значений входных параметров на подобласти, из каждой из которых требуется выбрать по одному тестовому набору. При тестировании логических выражений, когда область входных значений не делится на подобласти, данные меры эффективности совпадают друг с другом. Для критерия тестирования C и содержащей дефект реализации логичесого выражения P, эти меры равны m/d, где d есть число всех наборов тестирования, удовлетворяющих критерию C и m есть число наборов тестировани, удовлетворяющих C и обнаруживающих дефект в P.

Заметим, что невозможно исследовать эффективность критериев тестирования относительно любых возможных дефектов без каких-либо ограничений или допущений относительно дефектов. Действительно, допустим изучается эффективность тестирования реализации логического выражения с фиксированным числом элементарных условий. Если считать, что любая дефектная реализация этого логического выражения возможна и равновероятна, то никакой подход (критерий) не может дать преимущества в тестировании. Данная ситуация аналогична игре в рулетку, где невозможно придумать стратегию выбора ставок, повышающую шансы на выигрыш. В такой ситуации эффективность критерия зависит только от числа тестов в одном тестовом наборе (чем больше тестов, тем эффективней) и критерии с одинаковым размером тестовых наборов являются равно-эффективными.

На практике, не все дефектные реализации являются равновероятными, однако априорная информация о типах и вероятностях дефектов обычно отсутствует. Существует два направления эмпирического изучения эффективности критериев тестирования:

– рассматривать применение различных критериев к реальному программному обеспечению с реальными известными (ранее обнаруженными) дефектами;

– рассматривать искусственно внесенные дефекты различных типов (мутационное тестирование).

Однако, оба подхода имеют свои недостатки. Результаты исследования эффективности для реальных программных продуктов могут существенно отличаться для различных программ. В свою очередь, при исследовании эффективности относительно конкретных типов дефектов отсутствует обоснованная информация о том, какого типа дефекты будут встречаться при практичесом тестировании.

Следующие типы дефектов изучались при иследовани эффективности тестирования логических формул [19, 25] :

– VNF (Variable Negation Fault) – одно вхождение какой-либо логической переменной заменяется ее отрицанием.

– ENF (Expression Negation Fault) –логическое выражение заменяется его отрицанием.

– VRF (Variable Reference Fault) –вхождение одной логической переменной заменяется на другую переменную или константу.

– ORF (Operator Reference Fault) – один логический оператор заменяется на другой.

– ASF (Associative Shif Fault) – изменение порядка действий в формуле.

Эффетивность различных критериев тестирования для указанных и других типов дефектов изучалась многими исследователями как теоретически, так и экспериментально.

Аналитически наиболее полно изучено отношение «включения» (subsumption) между критериями [26, 27]. Говорят, что критерий C_1 включает критерий C_2, если каждый тестовый набор, удовлетворяющий C_1 также удовлетворяет C_2. Однако, как показано в [28], тот факт, что C_1 включает C_2, не гарантирует, что критерий C_1 более эффективен, чем критерий C_2.

Наиболее часто и полно экспериментально изучались и сранивались между собой метод случайного тестирования, критерии потоков данных и потоков управления [11, 12, 17, 18]. Среди последних работ по экспериментальному изучению критериев тестирования можно выделить [29], где сравнивает-

ся эффективность CC, DC и MC/DC. При этом, чтобы компенсировать влияние размеров тестовых наборов, для CC и DC добавлялись отдельные тесты и затем сравнивалась эффективность тестовых наборов одинакового размера. Результаты этих исследований показали, что все данные критерии превосходят по эффективности случайное тестирование и что MC/DC значительно более эффективен, чем остальные критерии даже после компенсации размеров тестовых наборов.

3. Устойчивость критериев тестирования

3.1. Определение устойчивости

Практическое применение какого-либо критерия тестирования подразумевает, что достаточно использовать любой единственный тестовый набор, который удовлетворяет данному критерию. Для предсказания эффективности одного специфического тестового набора недостаточно знать среднюю эффективность критерия. Необходимо также знать диапазон распределения эффективности для всех наборов, удовлетворяющих данному критерию.

Данное свойство было названо *устойчивостью* критерия тестирования [30], подразумевая под этим способность каждого тестового набора, удовлетворяющего критерию, иметь эффективность, близкую к средней эффективности критерия. Для критерия с *высокой устойчивостью*, эффективность отдельных тестовых наборов различается незначительно и близка к средней эффективности. Для критерия с *низкой устойчивостью* эффективность отдельных тестовых наборов может значительно различаться. Поэтому в последнем случае высокая средняя эффективность не гарантирует такой же высокой эффективности выбранного тестового набора.

Например, пусть эффективность некого гипотетического критерия для определенного типа отказов равна 0.5 и при этом эффективность отдельных тестовых наборов имеет равномерное распределение. В таком случае невозможно предсказать реальную эффективность тестирования, которая может с равной вероятностью быть как высокой, так и низкой. Поэтому для практического тестирования целесообразно использовать критерии не только с высокой эффективностью, но и с высокой устойчивостью.

Одной из возможных мер устойчивости является среднеквадратическое отклонение $T(C, F, S)$. Если $T(C_1, F, S) < T(C_2, F, S)$, то критерий C_1 имеет более высокую устойчивость, чем критерий C_2.

3.2. Экспериментальная оценка устойчивости

Рассмотрим приведенный в [30] пример оценки устойчивости двух критериев, CC и RC/DC, для логического выражения *s*, содержащего восемь элементарных логических условий, обозначенных буквами от A до H:

$$\neg(A\wedge B)\wedge(D\wedge\neg E\wedge\neg F\vee\neg D\wedge E\wedge\neg F\vee\neg D\wedge\neg E\wedge\neg F)\wedge$$
$$((A\wedge C\wedge(D\vee E)\wedge H\vee A\wedge(D\vee E)\wedge\neg H)\vee B\wedge(E\vee F)). \quad (1)$$

Данное выражение рассматривалось в [14, 19] как часть спецификаций Traffic Alert and Collision Avoidance System, TCAS II [31].

Эффективность и устойчивость данных критериев были изучены в [V] для отказов типа *Operator Reference Faults* (ORF) [19, 25], когда один логический оператор ошибочно заменяется другим оператором, в данном случае оператор «∧» заменяется оператором «∨» и наоборот. Значительное количество тестовых наборов согласно CC и RC/DC было применено для тестирования множества логических выражений, включающих все возможные логические ошибки типа ORF. Эксперимент показал следующую среднюю эффективность критериев:

$$E(CC,ORF, s) = 0{,}34;\ E(RC/DC,ORF, s) = 0{,}92.$$

Эффективность RC/DC значительно выше, чем эффективность CC. Данный факт частично объясняется тем, что число отдельных тестов в тестовом наборе для RC/DC больше, чем для CC. Более интересен результат оценки устойчивости данных критериев:

$$T(CC,ORF, s) = 0{,}17;\ T(RC/DC,ORF, s) = 0{,}05.$$

Как видно, устойчивость RC/DC более чем в три раза выше, чем устойчивость CC. Учитывая высокую среднюю эффективность RC/DC, это означает, что мы также можем быть уверены в высокой эффективности одного случайно взятого (согласно RC/DC) тестового набора. Данный факт подтверждает целесообразность использования RC/DC для практического тестирования.

Заключение

В статье рассмотрены различные аспекты эффективности критериев программного обеспечения и дается обзор современных результатов исследований в данной области. Рассмотрено понятие устойчивости критерия тестирования, которое характеризует способность каждого отдельного тестового набора, удовлетворяющего данному критерию, обеспечивать эффективность, близкую к средней

эффективности критерия.

Таким образом, высокая устойчивость гарантирует стабильную и предсказуемую эффективность в ходе тестирования.

Несмотря на то, что в области эффективности критериев тестирования проводились интенсивные теоретические и экспериментальные исследования, многие вопросы еще ждут своего изучения.

К перспективным направлениям дальнейших исследований относятся:

– Эффективность новых критериев тестирования, например, RC/DC критерия.

– Эффективность известных критериев тестирования для новых (нетрадиционных) типов дефектов и/или областей применения. Например, представляет интерес изучение эффективности комбинаторных критериев (таких, как pair-wise критерий) для тестирования логических выражений.

– Устойчивость критериев тестирования. Применение экспериментальных методов исследования может быть особенно полезным в данном направлении.

– Выработка практических рекомендаций по выбору критериев тестирования для различных типов программного обеспечения, в первую очередь для программного обеспечения, важного для безопасности.

Литература

1. Zhu H. Software unit test coverage and adequacy / H. Zhu, P.A. Hall, H. R. May // ACM Computing Surveys. – 1997. – Vol. 29 (4). – P. 336–427.

2. Myers, G. The Art of Software Testing / G. Myers. – Wiley-Interscience, 1979.

3. Chilenski, J. Applicability of Modified Condition/Decision Coverage to software testing / J. Chilenski, S. Miller // Software Engineering Journal. – 1994. – Vol. 9 (5). – P. 193–200.

4. RTCA/DO-178B. Software Considerations in Airborne Systems and Equipment Certification / RTCA, Washington D.C., USA. – 1992.

5. Vilkomir, S.A. Formalization of software testing criteria using the Z notation / S.A. Vilkomir, J.P. Bowen // Proceedings of 25th IEEE Annual International Computer Software and Applications Conference (COMPSAC 01), Chicago, Illinois, USA, October 8–12, 2001, IEEE Computer Society Press. – P. 351–356.

6. Vilkomir, S.A. Reinforced Condition/Decision Coverage (RC/DC): A new criterion for software testing / S.A. Vilkomir, J.P. Bowen // In Bert, D., Bowen, J.P., Henson, M.C., Robinson, K. (eds.), ZB2002: Formal Specification and Development in Z and B, Proceedings of 2nd International Conference of B and Z Users, Grenoble, France, January 23–25, 2002, Springer-Verlag, LNCS 2272. – P. 295–313.

7. Vilkomir, S.A. From MC/DC to RC/DC: Formalization and Analysis of Control-Flow Testing Criteria / S.A. Vilkomir, J.P. Bowen // Formal Aspects of Computing. – 2006. – Vol. 18 (1). – P. 42-62.

8. Vilkomir, S.A. Using MC/DC and RC/DC criteria for specification-based testing of safety-critical software / Radio-electronic and Computer Systems. – 2006. – Vol. 6 (18). – P. 130-135.

9. Grindal M. Combination testing strategies: A survey / M. Grindal, J. Offutt, S.F. Andler // Software Testing, Verification, and Reliability. – 2005. – Vol. 15 (3). – P. 167-199.

10. Frankl, P. A formal analysis of the faultdetecting ability of testing methods / P. Frankl, E. Weyuker // IEEE Transactions on Software Engineering. – 1993. – Vol. 19 (3). – P. 202–213.

11. Hutchins, M. Experiments on the effectiveness of dataflow- and control-flow based test adequacy criteria / M. Hutchins, H. Foster, T. Goradia, T. Ostrand // Proceedings of 16th International Conference on Software Engineering (ICSE-16), 1994. – P. 191–200.

12. Ntafos, S.C. On comparisons of random, partition, and proportional partition testing / S.C. Ntafos // IEEE Transactions on Software Engineering. – 2001. – Vol. 27 (10). – P. 949–960.

13. Offutt, A.J. Criteria for generating specification-based tests / A.J. Offutt, Y. Xiong, S. Liu // Proceedings of 5th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS'99), Las Vegas, Nevada, USA, October 18–21, 1999, IEEE Computer Society Press. – P. 119–129.

14. Paradkar, A. Test generation for Boolean expressions / A. Paradkar, K. Tai // Proceedings of 6th International Symposium on Software Reliability Engineering (ISSRE'95), Toulouse, France, 1995. – P. 106–115.

15. Weyuker, E. Thinking formally about testing without a formal specification / E. Weyuker // Proceedings of Formal Approaches to Testing of Software (FATES'02), A Satellite Workshop of CONCUR'02, Brno, Czech Republic, August 24, 2002. – P. 1–10.

16. Wong, W. Test set size minimization and fault detection effectiveness: A case study in a space application / W. Wong, J. Horgan, A. Mathur, A. Pasquini // Proceedings of 21st Annual International Computer Software and Applications Conference (COMPSAC 97), Washington, DC, USA, August 13–15, 1997, IEEE Computer Society Press. – P. 522–528.

17. Frankl, P. Further empirical studies of test effectiveness / P. Frankl, O. Iakounenko // Proceedings of ACM SIGSOFT 6th International Symposium on Foundations of Software Engineering, November 1998, Vol. 23, No. 6. – P. 153–162.

18. Frankl, P. An experimental comparison of the effectiveness of branch testing and data flow testing / P. Frankl, S. Weiss // IEEE Transactions on Software Engineering. – 1993. – Vol. 19 (8). – P. 774–787.

19. Weyuker, E. Automatically generating test data from a Boolean specification / E. Weyuker, T. Goradia, A. Singh // IEEE Transactions on Software Engineering. – 1994. – Vol. 20 (5). – P. 353–363.

20. Vouk, M. Empirical studies of predicate-based software testing / M. Vouk, K.C. Tai, A. Paradkar //

Proceedings of 5th International Symposium on Software Reliability Engineering, 1994. – P. 55–64.

21. Chen, T.Y. On the statistical properties of testing effectiveness measures / T.Y. Chen, F. Kuo, R.J. Merkel // Journal of Systems and Software. – 2006. – Vol. 79 (5). – P. 591-601.

22. Chen, T.Y. Adaptive random testing / T.Y. Chen, H. Leung, I. Mak // Proceedings of the 9th Asian Computing Science Conference (ASIAN 2004), LNCS Vol. 3321, Springer, 2004. – P. 320–329.

23. Chen, T.Y. An upper bound on software testing effectiveness / T.Y. Chen, R. Merkel // ACM Transactions on Software Engineering and Methodology (TOSEM). – 2008. – Vol. 17 (3). – P. 1-27.

24. Weyuker, E. Can we measure software testing effectiveness? / E. Weyuker // Proceedings of 1st International Software Metrics Symposium, Baltimore, USA, May 21–22, 1993. – P. 100–107.

25. Kuhn, D. Fault classes and error detection capability of specification-based testing / D. Kuhn // ACM Transactions on Software Engineering and Methodology. – 1999. – Vol. 8 (4). – P. 411–424.

26. Ntafos, S.C. A comparison of some structural testing strategies / S.C. Ntafos // IEEE Transactions on Software Engineering. – 1988. – Vol. 14 (6). – P. 868-874.

27. Rapps S. Selecting Software Test Data Using Data Flow Information / S. Rapps, E. Weyuker // IEEE Transactions on Software Engineering. – 1985. – Vol. 11 (4). – P. 367-375.

28. Frankl, P.G. Assessing the fault-detecting ability of testing methods / P.G. Frankl, E. Weyuker // ACM SIGSOFT Software Engineering Notes. – 1991. – Vol. 16 (5). – P. 77-91.

29. Lau M. On Comparing Testing Criteria for Logical Decisions / M. Lau, Y. Yu // Proceedings of the 14th Ada-Europe International Conference, Brest, France, June 8-12, 2009. LNCS, Volume 5570. – P. 44-58.

30. Vilkomir S.A. Tolerance of Control-Flow Testing Criteria / S. A. Vilkomir, K. Kapoor, J. P. Bowen // Proceedings of 27th IEEE Annual International Computer Software and Applications Conference (COMPSAC 2003), Dallas, Texas, USA, 3-6 November 2003. IEEE Computer Society Press, 2003. – P. 182-187.

31. Leveson, N.G. Requirements specification for process-control systems / N.G. Leveson, M.P.E. Heimdahl, H. Hildreth, J.D. Reese // IEEE Transactions on Software Engineering. – 1994. – Vol. 20 (9). – P. 684–707.

Поступила в редакцию 18.01.2010

Рецензент: д-р техн. наук, проф. В.С. Харченко, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков.

ПІДХІДИ ДО ПОРІВНЯННЯ КРИТЕРІЇВ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕНИЯ

С.А. Вілкомір

Для тестування програмного забезпечення застосовуються різноманітні стратегії (критерії) тестування. Найбільш важливою властивістю таких критеріїв є їх ефективність, тобто спроможність виявляти дефекти у програмному забезпеченні. В статті надається огляд сьогочасних робіт по визначенню і порівнянню ефективності і також різноманітних критеріїв тестування. Показано, що поряд з ефективністю, також важливою є стійкість критерію, яка визначається діапазоном розподілення ефективності для різних тестових наборів, задовольняючих одному и тому ж критерію. Наводяться приклади експериментальної оцінки стійкості.

Ключові слова: програмне забезпечення, тестування, критерії, ефективність, стійкість

APPROACHES TO COMPARISON OF SOFTWARE TESTING CRITERIA

S.A. Vilkomir

Various testing strategies (criteria) are used for software testing. The most important feature of such criteria is effectiveness, i.e., ability to detect faults in a software program. The paper gives a survey of recent results on evaluation and comparison of effectiveness of various testing criteria. It is shown that, along with effectiveness, tolerance of criteria is also important. Tolerance is determined by scope of effectiveness distribution for test sets, which satisfy the same criterion. Examples of experimental evaluation of tolerance are provided.

Key words: software, testing, criteria, effectiveness, tolerance.

Sergiy Vilkomir – PhD, Department of Computer Science, East Carolina University, NC, USA e-mail: vilkomirs@ecu.edu.