

УДК 004.519.217

Д.А. Маевский, канд. техн. наук, доц.,
С.А. Яремчук, инженер,
Одес. нац. политехн. ун-т

МЕТОД ВЕРОЯТНОСТНОЙ ОЦЕНКИ КОЛИЧЕСТВА ДЕФЕКТОВ В ПРОГРАММНЫХ МОДУЛЯХ

Д.А. Маєвський, С.О.Яремчук. Метод імовірнісної оцінки кількості дефектів в програмних модулях. Запропонована комплексна метрика складності програмних модулів, з використанням якої створено метод імовірнісної оцінки кількості дефектів. Проведена верифікація методу показала незначні відхилення оцінок прогнозних та фактичних показників, що є свідченням достатньо високої точності методу.

Ключові слова: логістична оцінка, імовірнісна оцінка, кількість дефектів, програмні модулі, метрики складності.

Д.А. Маевский, С.А. Яремчук. Метод вероятностной оценки количества дефектов в программных модулях. Предложена комплексная метрика сложности программных модулей, с использованием которой создан метод вероятностной оценки количества дефектов. Проведенная верификация метода показала незначительные отклонения оценок прогнозных и фактических показателей, что свидетельствует о достаточно высокой точности метода.

Ключевые слова: логистическая оценка, вероятностная оценка, количество дефектов, программные модули, метрики сложности.

D.A. Maevsky, S.A. Yaremchuk. Method of probabilistic estimation defects quantity in program modules. The complex metrics of complexity of program modules is offered. With use of this metrics the method of probabilistic estimation defects quantity in program modules is created. Method verification is executed. Verification has shown insignificant deviations of estimations of settlement and actual indicators. It speaks about enough big accuracy of this method.

Keywords: logistical estimation, probabilistic estimation, quantity of defects, program modules, complexity metrics.

Дефекты программного обеспечения (ПО) являются основной причиной снижения его надежности. По данным компаний-разработчиков около пятидесяти процентов средств разработки ПО расходуется на тестирование. Избежать этих громадных капиталовложений, составляющих миллиарды долларов в год по всему миру, невозможно, поскольку тестирование — важное, а зачастую единственное, средство контроля качества и надежности ПО [1]. Повышение эффективности капиталовложений в тестирование предполагает обнаружение максимального числа дефектов ПО в кратчайшие сроки при минимальных усилиях тестируемых. Для этого до начала тестирования необходимо знание количества дефектов в тестируемом ПО, и их местоположения. Поэтому разработка методов точной прогнозной оценки количества дефектов в модулях ПО является актуальной научно-технической задачей.

Модульное тестирование является первоначальным, наиболее важным и трудоемким этапом тестирования [2]. А поскольку оптимизация модульного тестирования — значительный фактор экономии средств разработки ПО, то и оценку дефектов целесообразно рассматривать на уровне модуля. Стандарт [3] определяет модуль (или компонент) ПО, как отдельную дискретную идентифицируемую структурную единицу разрабатываемого ПО.

В настоящей работе, в соответствии с установившимся подходом [1, 4, 5], считается, что при неизменных условиях и технологии разработки, а также при неизменной квалификации разработчика, основным фактором возникновения дефектов является сложность ПО.

Для учета программной сложности разработано множество различных метрик, основанных на статическом анализе исходного кода. В [4] проведен анализ различных метрик кода с целью их использования для прогноза количества дефектов. В работе [5] выполнено сравнение методов прогнозирования оценки дефектности модулей. Перечисленным методам свойственны недостаточная точность, невозможность прогноза количества дефектов в модуле, сложность и ограниченность применения. Необходимость преодоления указанных проблем определяет *цель настоящего исследования* — разработку достаточно точного и менее сложного метода прогнозирования оценки количества дефектов в программных модулях с использованием метрик сложности.

Для достижения указанной цели в работе решены следующие задачи:

- на базе известных метрик сложности сформирована комплексная метрика;
- разработан метод прогнозирования вероятностной оценки количества дефектов в модулях ПО с использованием комплексной метрики;
- выполнена верификация предложенного метода с использованием экспериментальных данных программных проектов с открытым исходным кодом.

Сложность ПО многомерна. Ее количественное измерение возможно с использованием некоторого множества метрик сложности, вычисляемых для каждого модуля ПО в отдельности. Проведенные авторами исследования различных метрик объектно-ориентированного кода десяти программных проектов из депозитария экспериментальных данных [6] показали различную степень связи между значениями отдельных метрик и количеством дефектов. Коэффициент корреляции R различных метрик изменялся в максимально возможном интервале $-1 \leq R \leq 1$. При этом в модулях при одинаковом количестве дефектов значения одних метрик увеличивались, значения других уменьшались, значения третьих оставались неизменными. Во избежание привлечения сложного математического аппарата для учета многомерных зависимостей между значениями метрик и количеством дефектов для каждого модуля предлагается вычисление комплексной метрики (КМ) в виде выражения:

$$KM = \sum_{k=1}^n Z_k R_k \quad (1)$$

где Z_k — значение k -той метрики;

R_k — коэффициент корреляции между значениями k -той метрики и количеством дефектов модулей ПО;

n — количество метрик сложности кода с положительными R .

С целью получения максимальных коэффициентов корреляции (R_{KM}) из расчета по формуле (1) последовательно и итеративно исключались метрики, понижающие его значение. В результате для каждого из десяти программных проектов была получена КМ с $R_{KM} \geq \max R_k \forall k$.

В основу предложенного метода положены следующие допущения:

- количество дефектов в модулях программного проекта обусловлено сложностью этих модулей, которая, в свою очередь, может быть количественно выражена значением КМ;
- пространством элементарных событий для расчета вероятностных характеристик являются факты обнаружения в модуле одного, двух, трех и так далее дефектов, или обнаружения бездефектного модуля;
- вероятности наличия определенного количества дефектов в модулях одинаковой сложности одинаковы в пределах одного программного проекта.

Метод предполагает деление всего множества модулей ПО на две части. Первая часть модулей (назовем ее базовой) используется для расчета вероятностей дефектов в модулях различной сложности, в зависимости от значений КМ. Для второй части модулей (назовем ее прогнозная) на основе полученных вероятностей выполняется прогнозная оценка количества дефектов.

Алгоритм метода состоит из последовательности 8 шагов:

- 1) рассчитать значения стандартных метрик для всех модулей проекта;
- 2) рассчитать по формуле (1) комплексную сложность каждого модуля – значение КМ;
- 3) весь числовой диапазон КМ разделить на статистически значимые интервалы (не менее 20 модулей в интервале) с различными количествами дефектов;
- 4) выполнить тестирование базового подмножества модулей и определить количество обнаруженных дефектов для каждого модуля;
- 5) на основе данных п. 4 рассчитать вероятности отсутствия или наличия дефектов в модулях, путем деления количества модулей в данном интервале КМ, содержащих один, два и т.д. дефектов на общее количество модулей в этом интервале;
- 6) на основе данных п. 4 рассчитать вероятности наличия различного числа дефектов в дефектных модулях, отношением суммарного количества дефектов в интервале КМ, к количеству дефектов в модулях;
- 7) подсчитать количество модулей в прогнозной части проекта для интервалов КМ, и на основе данных п. 5 получить прогнозное количество модулей с различным количеством дефектов в интервалах КМ;
- 8) используя полученные в п.7 результаты, рассчитать количество дефектов в модулях прогнозной части проекта.

В результате использования описанного метода будут получены вероятностные количественные оценки количества дефектов в каждом модуле прогнозной части, и общее количество дефектов проекта.

Для верификации метода использовались экспериментальные данные, опубликованные в депозитарии данных [6], в виде значений метрик и количества дефектов для каждого модуля программных проектов с открытым исходным кодом. Были выбраны десять проектов, соответствующих различным предметным областям. Выборка является репрезентативной по различным аспектам. Проекты существенно отличаются количеством строк исходного кода, количеством модулей, дефектов, языками программирования. Количество дефектов на тысячу строк кода в разных проектах колеблется от 1 до 8. Информация о проектах представлена в первой части таблицы 1.

После получения прогнозных вероятностных количественных оценок подсчитывалось фактическое количество дефектных модулей и количество дефектов прогнозной части проекта, вычислялись отклонения прогнозных значений от фактических по каждому проекту. Отклонение 1 рассчитывалось, как отношение разницы прогнозного и фактического количества дефектных модулей, к общему количеству модулей. Отклонения 2, 3, 4, 5 рассчитывались как отношение разницы прогнозного и фактического количества модулей с одним, двумя, тремя и четырьмя дефектами соответственно к общему количеству дефектных модулей. Отклонение 6 рассчитывалось, как отношение разницы прогнозного и фактического количества дефектов, к общему количеству дефектов. Полученные отклонения оценок представлены во второй части таблицы 1.

Приведем пример использования предложенного метода для проекта номер 6 из таблицы 1. В базовую часть проекта было включено 370 модулей, в общей сложности содержащих 159 дефектов. Результаты выполнения п. 1..4 алгоритма показаны в таблицах 2 и 3. В таблице 3 приведены детальные данные о дефектах в модулях базовой части проекта. Рассчитанные в п. 6 алгоритма вероятности наличия различного числа дефектов в дефектных модулях представлены в таблице 4. В таблице 5 приведены рассчитанные в соответствии с п. 7, 8 вероятности появления модулей с различным числом дефектов в прогнозной части проекта.

Таблиця 1

Характеристики проектів і відхилення оцінок для них

Характеристики програмних проектів				Процент відхилень					
№ проекту	Кол-во модулів	Кол-во строк	Кол-во дефектів	1	2	3	4	5	6
1	231	55 538	341	9,18	10,28	9,04	11,58	6,69	5,78
2	882	411 737	618	0,63	3,06	0,53	1,13	0,08	2,31
3	379	119 701	496	2,33	5,53	2,92	1,17	0,80	9,46
4	438	129 307	475	5,33	2,35	5,95	6,77	0,77	10,22
5	351	113 247	184	4,07	12,76	2,49	6,03	4,42	11,96
6	645	208 654	338	0,18	3,74	4,03	4,11	1,54	8,13
7	608	66 302	522	0,99	7,79	5,97	2,59	9,16	5,87
8	856	287 317	114	0,66	13,00	2,05	0,35	3,55	0,50
9	964	113 026	472	4,51	14,14	4,01	4,23	2,78	11,78
10	944	112 926	394	4,29	18,24	3,29	2,21	2,07	12,31
Середні відхилення				3,22	9,09	4,03	4,02	3,19	7,83

Таблиця 2

Дані про модулі базової частини проекту

Інтервали КМ	Кол-во модулів без деф.	Кількість дефектних модулів					Всього модулів	Процент дефектних модулів
		Общее	с 1 деф.	с 2 деф.	с 3 деф.	с 4 і більше деф.		
5	67	2	2				69	19
10	54	2	2				56	15
15	46	4	3		1		50	14
20	31	8	7	1			39	11
30	53	11	8	3			64	17
50	28	24	15	2	3	4	52	14
100	14	21	6	5	5	5	35	9
200		5		1		4	5	1
Всього	293	77	43	12	9	13	370	100
Проц.	79	21	12	3	2	2	100	

Таблиця 3

Дані про дефекти базової частини проекту

Інтервали КМ	Общее количество дефектов				Всього дефектов	Процент дефектов
	в 1-о деф. модулях	в 2-х деф. модулях	в 3-х деф. модулях	в 4-х деф. модулях		
5	2				2	1
10	2				2	1
15	3		3		6	4
20	7	2			9	5
30	8	6			14	9
50	15	4	9	18	46	29
100	6	10	15	24	55	35
200		2		23	25	16
Всього	43	24	27	65	159	100
Проц.	27	15	17	41	100	

Таблиця 4

Значения вероятностей отсутствия/наличия дефектов в модулях

Интервалы КМ	Вероятность отсутствия деф. в мод.	Вероятность наличия деф. в мод.	Вероятность наличия дефектов в модулях			
			1 деф.	2 деф.	3 деф.	4 деф.
5	0,97	0,03	0,03			
10	0,96	0,04	0,04			
15	0,92	0,08	0,06		0,02	
20	0,80	0,20	0,17	0,03		
30	0,83	0,17	0,13	0,04		
50	0,54	0,46	0,28	0,04	0,06	0,08
100	0,40	0,60	0,18	0,14	0,14	0,14
200		1		0,20		0,80

Таблиця 5

Значения вероятностей появления модулей с заданным числом дефектов для прогнозной части проекта

Интервалы КМ	Вероятности наличия модулей			
	с 1 деф.	с 2 деф.	с 3 деф.	с 4 деф.
5	1,00			
10	1,00			
15	0,50		0,50	
20	0,78	0,22		
30	0,57	0,43		
50	0,33	0,09	0,20	0,39
100	0,11	0,18	0,27	0,44
200		0,08		0,92

Результаты сравнения прогнозного и фактического количества дефектных модулей исследуемых проектов представлены на рис. 1.

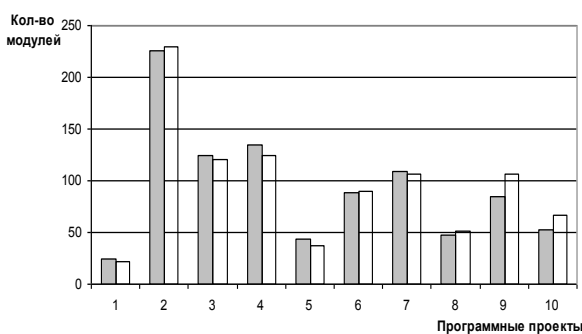


Рис. 1. Фактическое (темные столбцы) и прогножное (светлые столбцы) количество модулей с дефектами по проектам

Из рисунка видно, что метод с высокой точностью позволяет прогнозировать число модулей с заданным количеством дефектов в прогнозной части проекта. Из данных в таблицах 2...5 можно сделать выводы относительно организации процесса тестирования. Из таблиц 4 и 5 можно судить о вероятности наличия дефектов модуля любой сложности, значении критической сложности модуля, вероятности появления определенного количества дефектов в любом дефектном модуле. Эти показатели позволят тестировщикам обоснованно отбирать модули для тестирования, планировать очередность тестирования и тестовые усилия

при работе с этими модулями. Сравнение количества выявленных в процессе тестирования дефектов с их прогнозной оценкой позволит судить о степени завершенности процессов тестирования. Установлено, что точность прогноза в большой степени зависит от статистической представительности дефектных модулей в каждом метрическом интервале. Чем больше модулей с различным количеством дефектов в каждом интервале, тем точнее определяются вероятности, и точнее количественная прогнозная оценка. На точность оценки влияет также R_{KM} . Чем выше его значение, тем точнее оценка.

Для учета многомерной сложности ПО, на основе стандартных метрик, предложена

комплексная метрика сложности программных модулей. С использованием комплексной метрики *предложен метод* вероятностной оценки количества дефектов в программных модулях. Проведенная верификация метода показала, что средние отклонения оценок количества дефектных модулей составили 3,22 %; оценок количества модулей с 1, 2, 3, 4 дефектами 5,08 %; оценок общего количества дефектов 7,83 %. Незначительные (до 10 %) отклонения свидетельствует о том, что предложенный метод показал высокую прогнозную способность и точность количественной оценки дефектных модулей и дефектов в ПО.

В отличие от методов, использующих байесовские сети доверия, данный метод не нуждается в экспертных оценках вероятностей, и привлечении специалистов высокой квалификации. Метод отличается от методов на основе машинного обучения простотой математического аппарата, и отсутствием необходимости применения сложных и дорогостоящих программных средств. Метод не зависит от языка программирования, поскольку использует различные стандартные метрики. Информация, полученная данным методом, может быть использована для принятия различных управленческих решений в процессе разработки ПО. Перечисленные обстоятельства делают метод доступным для широкого использования специалистами софтверных компаний.

Литература

1. Макконнелл, С. Совершенный код. Мастер-класс / С. Макконел. — СПб. : Питер, 2005. — 896 с.
2. Благодатских, В.А. Стандартизация разработки программных средств / В.А. Благодатских, В.А. Волынин, К.Ф. Посакалов // М.: Финансы и статистика, 2003. — 284 с.
3. IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology.
4. Khoshgoftaar, T.M. Predicting software development errors using complexity metrics. / T.M. Khoshgoftaar, J.C. Munson // IEEE of Selected Areas in Communications. — 1990. — Vol. 8, № 2. — P. 253 — 261.
5. Bellini, P. Comparing fault-proneness estimation models. /P. Bellini, I. Bruno, P. Nesi, D. Rogai // In: Proc. of 10th IEEE International Conference on Engineering of Complex Computer Systems. — 2005. — P. 205 — 214.
6. Menzies, T. The PROMISE Repository of empirical software engineering data [Electronic resource] / [T. Menzies, B. Caglayan, E. Kocaguneli and others] // Access mode: <http://promisedata.googlecode.com>. — 01.10.2012.

References

1. McConnell, S. Sovershennyj kod. Master-klass [Code complete. Master class] / S. McConnell. — St.-Petersburg, 2005. — 896 p.
2. Blagodatsky, V.A. Standartizacija razrabotki programnyh sredstv [Standardization of construction of software] / V.A. Blagodatsky, V.A. Volinin, K.F. Poskakov . — Moscow, 2003. — 284 p.
3. IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology.
4. Khoshgoftaar, T.M. Predicting software development errors using complexity metrics. / T.M. Khoshgoftaar, J.C. Munson // IEEE of Selected Areas in Communications. — 1990. — Vol. 8. — # 2. — pp. 253 — 261.
5. Bellini, P. Comparing fault-proneness estimation models. /P. Bellini, I. Bruno, P. Nesi, D. Rogai // In: Proc. of 10th IEEE International Conference on Engineering of Complex Computer Systems. — 2005. — pp. 205 — 214.
6. Menzies, T. The PROMISE Repository of empirical software engineering data [Electronic resource] / [T. Menzies, B. Caglayan, E. Kocaguneli and others] // Access mode: <http://promisedata.googlecode.com>. — 01.10.2012.

Рецензент д-р техн. наук, проф. Одес. нац. политехн. ун-та Дрозд А.В.

Поступила в редакцию 24 декабря 2012 г.