

## **ОБ ОЦЕНКЕ НАДЁЖНОСТИ ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НА РАННИХ ЭТАПАХ ЕГО ПРОЕКТИРОВАНИЯ**

В.Т. Лэ, С.С. Дик

Научный руководитель – Боровиков С.М.

канд. техн. наук, доцент

### **Белорусский государственный университет информатики и радиоэлектроники**

В настоящее время актуальной является проблема исследования надежности программного обеспечения (ПО)[1-2]. В рамках данной проблемы можно выделить ряд частных задач, таких как:

- определение основных факторов, влияющих на надежность ПО;
- разработка методов оценки надежности ПО;
- разработка методов, обеспечивающих достижение заданного уровня надежности ПО.

Под надежностью ПО понимается его способность безотказно выполнять определенные функции при заданных условиях в течение заданного периода времени с достаточно большой вероятностью.

Функциональная надежность ПО – комплексное свойство. В зависимости от назначения и условий применения может включать безотказность, работоспособность, безопасность и защищенность.

Оценка надежности ПО, как и любого другого объекта, начинается с определения критериев, по которым будет оцениваться надежность[3,4]. При анализе надежности ПО используются традиционные для технических систем критерии надежности: вероятность безотказной работы за заданное время ( $t_3$ ) –  $P(t_3)$ , вероятность отказа  $Q(t_3)$ , средняя наработка до отказа (*Mean Time To Failure*)  $T_{ср}$ , средняя наработка на отказ (*Mean Time Between Failures*)  $T_0$ , интенсивность отказов  $\lambda(t)$ , коэффициент готовности  $K_T$ , вероятность восстановления  $v(t_3)$  и среднее время восстановления  $T_B$ .

Для количественной оценки показателей надежности программного обеспечения используют модели надежности, под которыми понимаются математические модели, построенные для оценки зависимости надежности от заранее известных или определённых в ходе выполнения задания параметров. Эти модели можно разбить на две основные группы: эмпирические и аналитические. Аналитические модели разделяют на статические и динамические[5].

Известные методы оценки надёжности прикладного ПО исходят из того, что имеются определённые данные о тестировании уже написанного ПО с использованием кода программ и устранены ошибки, обусловленные нарушением правил языка программирования. Однако в большинстве случаев проектировщиков информационно-компьютерных систем и разработчиков программного обеспечения для анализа информации с помощью этих систем интересует ожидаемый уровень надёжности прикладного ПО ещё до написания программного кода их программ. Возникает вопрос, как спрогнозировать ожидаемый уровень надёжности ПО на этом этапе.

В настоящее время к простой статистической модели относят феноменологическую модель фирмы TRW [6,7]. Используя эту модель, можно на этапе разработки технического задания на ПО оценить количество

ожидаемых в нём ошибок  $N$  до начала этапа тестирования. Феноменологическая модель представляет собой линейную модель оценки показателя сложности ПО по 5-ти эмпирическим характеристикам программ:

1. Логической сложности  $L_{tot}$ , измеряется числом логических операторов (разветвлений, циклов, условных переходов).
2. Сложности взаимосвязей  $C_{inf}$ , измеряется числом прикладных и системных интерфейсов рассматриваемого модуля, т.е. числом вызываемых из него прикладных и системных программ.
3. Сложности вычислений  $C_c$ , измеряется числом операторов, содержащих арифметические операции.
4. Сложности ввода-вывода  $C_{i/o}$ , измеряется числом операций ввода-вывода.
5. Понятности  $U_{read}$ , измеряется числом комментариев.

Показатель общей сложности определяют по формуле

$$C = L_{tot} + 0,1 \cdot C_{inf} + 0,2 \cdot C_c + 0,4 \cdot C_{i/o} + (-0,1) \cdot U_{read}. \quad (1)$$

Коэффициенты 1; 0,1; 0,2; 0,4 и (-0,1) задают относительные веса соответствующих показателей в аддитивной модели общей сложности. Для расчета числа ошибок  $N$  предлагается использовать следующую многофакторную модель:

$$N = L_{tot} \cdot k_1 + 0,1 \cdot C_{inf} \cdot k_2 + 0,2 \cdot C_c \cdot k_3 + 0,4 \cdot C_{i/o} \cdot k_4 + (-0,1) \cdot U_{read} \cdot k_5, \quad (2)$$

где  $k_i$  – коэффициент корреляции числа ошибок с  $i$ -м показателем сложности. Значения коэффициентов  $k_i$  можно найти методом наименьших квадратов, используя статистические данные.

#### *Библиографический список*

1. Программное обеспечение – источник всех проблем. [Электронный ресурс]. – 2019. – Режим доступа: <http://www.williamspublishing.com/PDF/5-8459-0785-3/part1.pdf>
2. Технология программирования. Надёжность программного обеспечения. [Электронный ресурс]. – 2019. – Режим доступа : [http://www.tehprog.ru/index.php\\_page=lecture13.html](http://www.tehprog.ru/index.php_page=lecture13.html)
3. Формальные методы обеспечения качества ПО. [Электронный ресурс]. – 2019. – Режим доступа : <http://kspt.icc.spbstu.ru/media/files/2010/course/softwarequality/lec2.pdf>
4. Методы обеспечения аппаратно-программной надёжности вычислительных систем. Д.т.н., проф. Чуканов В.О., к.т.н., доц. Гуров В.В. [Электронный ресурс]. – 2019. – Режим доступа : [http://www.mcst.ru/files/5357ec/dd0cd8/50af39/000000/seminar\\_metody\\_obespecheniya\\_apparatno-programmnoy\\_nadezhnosti\\_vychislitelnyh\\_sistem.pdf](http://www.mcst.ru/files/5357ec/dd0cd8/50af39/000000/seminar_metody_obespecheniya_apparatno-programmnoy_nadezhnosti_vychislitelnyh_sistem.pdf)
5. Лэ В.Т. Классификация моделей надёжности прикладного программного обеспечения/ В.Т. Лэ, С.С. Дик // 55-я юбилейная научная конференция аспирантов, магистрантов и студентов учреждения образования БГУИР, 22–26 апреля 2019 г., БГУИР, Минск, Беларусь: тезисы докладов. – Мн. – 2019. – С. 167–168.
6. Липаев, В. В. Надёжность программных средств / В.В. Липаев. – М.: СИНТЕГ, 1998.
7. Тейер Т. Надёжность программного обеспечения. / Т. Тейер, М. Липов, Э. Нельсон. – М.: Мир, 1981.