

## ОПРЕДЕЛЕНИЕ ДЛИТЕЛЬНОСТИ ПРОЦЕССА ТЕСТИРОВАНИЯ И ОТЛАДКИ ПРОГРАММ

Коваленко С.М., Платонова О.В., Пиньковский Е.А.

*Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский технологический университет» (МИРЭА), 119454, Россия, г. Москва, проспект Вернадского, 78, e-mail: kovalenko@mirea.ru*

---

**Одним из важных этапов жизненного цикла ПО является стадия тестирования и отладки. Длительность этого этапа является важным параметром в планировании разработки ПО. В настоящей статье рассматривается методика определения длительности этапа тестирования на фазе планирования при помощи модели надежности Джелинского-Моранды и метрики Хостеда.**

---

Ключевые слова: длительность тестирования; модель Джелинского-Моранды; планирование разработки

## DETERMINING TESTING AND DEBUGGING DURATION OF SOFTWARE

Kovalenko S.M., Platonova O.V., Pinkovskij E.A.

*Federal State Educational Institution of Higher Education "Moscow Technological University" (MIREA), 119454, Russia, Moscow, Vernadsky avenue, 78, e-mail: kovalenko@mirea.ru*

---

**One of the most important stages of software life cycle is testing and debugging. One of important parameters of testing is its duration. To determine this parameter we can use the Jelinski-Moranda reliability model to calculate the time between failures (finding errors) and Halstead's measures to predict the total number of failures on the planning phase of software development. The total number of failures is used by Jelinski-Moranda model.**

---

Key words: testing duration, Jelinski-Moranda model; software development planning.

### Введение

Важным параметром в процессе жизненного цикла программного обеспечения (ПО) является длительность процесса тестирования и отладки созданных программ. Процесс отладки программ в случае большой его длительности приводит как к существенным материальным затратам так и к репутационным потерям для программиста или коллектива программистов. Из этого вытекает важность достаточно точного прогнозирования длительности процесса тестирования и отладки программ в составе разрабатываемого ПО.

Для оценки длительности процессов тестирования и отладки программ на этапе формирования технического задания на разработку используются модели, позволяющие прогнозировать характеристики надежности разрабатываемого ПО. Используемые в данном случае модели надежности представляют интерес прежде всего для прогнозирования отказов в процессе эксплуатации и отладки программы. При этом значения параметров моделей определяют в процессе эксплуатации или отладки программы по данным о моментах возникновения отказов. Отсутствие общих справочных данных объясняется тем, что каждый программист является уникальным технологическим объектом по созданию программ, а каждая его программа – эксклюзивное изделие. Основной задачей, решаемой в данной работе, является оценка длительности процессов тестирования и отладки программы на начальных этапах тестирования на основе изучения характеристик самой программы. Рассматриваемая методика применима для вновь создаваемых программ в средах программирования C, C++.

### Определение длительности процесса тестирования и отладки программ

Наиболее известный и применяемый вариант оценки характеристик надежности опирается на модель надежности Джелинского-Моранды, которая будет рассмотрена ниже. Существует целый ряд моделей, описывающих процесс отказов ПО на этапе тестирования [1], например, модели Шумана, Шика-Волвертона и другие, но модель надежности Джелинского-Моранды имеет наиболее разработанный математический аппарат оценки характеристик надежности. Имеющийся в этой модели математический аппарат позволяет на основе длительности процессов тестирования и отладки программы на начальных этапах получать значимые результаты для оценки общей длительности процесса отладки программ.

Рассматриваемая модель основана на следующих допущениях:

1. Время до следующего отказа распределено экспоненциально;
2. Интенсивность отказов программы пропорциональна количеству оставшихся в программе ошибок.

Согласно этим допущениям, вероятность безотказной работы программ как функция времени равна:

$$P(t_i) = e^{-\lambda_i t_i}, \quad (1)$$

$$\text{где } \lambda_i = C \cdot (N - (i - 1)) \quad (2)$$

$$C = \frac{\sum_{i=1}^K \left( \frac{1}{N + 1 - i} \right)}{\sum_{i=1}^K \Delta t_i}, \quad (3)$$

где  $K$  - число экспериментально полученных интервалов между отказами,

$\Delta t$  – экспериментально полученные значения интервалов времени между отказами,

$N$  – значение общего числа ожидаемых ошибок.

Данная методика работает для  $K \geq 2$ , то есть надо иметь хотя бы два экспериментально полученных интервала между моментами возникновения ошибок.

Существующие методики оценки длительности процесса тестирования, например, модель Джелинского-Моранды [1] обладают существенным недостатком, связанным с тем, что ожидаемое число ошибок в модели  $N$  выбирается произвольно. Но именно ожидаемое число ошибок прежде всего влияет на длительность процесса тестирования программы. Общая длительность процесса тестирования в среднем может быть оценена как:

$$T = \sum_{i=1}^N \Delta t_i$$

Значение общего числа ожидаемых ошибок  $N$  может быть получено, например, исходя из статистики, имеющейся в организации (фирме), занимающейся разработкой ПО. В фирме Microsoft имеются данные о том, что в среднем написанная программа имеет 5 ошибок на 1000 строк кода, тогда как отлаженная программа может иметь в среднем 0,3 ошибки на 1000 строк кода. Недостатком такого подхода является то, что значение  $N$  от фирмы к фирме может меняться, а кроме того, для оценки  $N$  необходимо написать программу, то есть знать число строк кода в ней.

Другой подход для оценки числа ожидаемых ошибок  $N$  предлагается в работе М. Холстеда [2]. В ней предлагается проводить оценку длины кода программы и её объема на основе определения словаря программы, который включает число простых (или отдельных) операторов  $b$ , появляющихся в данной реализации программы, и числа простых (или отдельных) операндов  $d$ , также появляющихся в данной реализации программы. В соответствии с метриками Холстеда длина  $S$  и объем  $V$  программы определяются в соответствии с выражениями:

$$S = b \cdot \log_2 b + d \cdot \log_2 d,$$

$$V = S \cdot \log_2 (b + d)$$

Число ошибок  $N$ , согласно подходу Холстеда, может быть определено из выражения:

$$N = \frac{V}{E_K}, \text{ где } E_K \text{ составляет } 3000.$$

Развитие методики проводится в работах [3-5]. В этих работах проводится оценка возможности использования метрик М. Холстеда для современных языков программирования. Использование стандартных библиотек программ снижает оценку числа ожидаемых ошибок в пределах 10-20%.

### Пример расчета

Пусть  $b = 32$ ,  $d = 128$ , что характерно для средних по величине программ объемом до 1000 строк кода, написанных на языках Си, Си++.

Тогда  $S = 160 + 896 = 1054$ ,  $V = 7378$ .

Получаем  $N = 2,5$ . Округлив, получаем  $N = 3$ .

Допустим, что ошибки были обнаружены на этапе тестирования через интервалы времени  $X_i$ . Полученные данные сведены в следующую таблицу:

| i | $X_i$ (дн.) |
|---|-------------|
| 1 | 5           |
| 2 | 7           |

После возникновения каждой из ошибок можно с некоторой точностью прогнозировать длительность процесса тестирования и максимальное время до следующей ошибки.

Так, например, после возникновения второй ошибки имеем следующее:

$$K = 2, \Delta t_1 = 5, \Delta t_2 = 7.$$

По формуле (3) получаем  $C = \frac{5}{72}$ . Тогда по формуле (2)  $\lambda_3 = \frac{5}{72}(3-2) = \frac{5}{72}$  и  $T_3 = \frac{1}{\lambda} = \frac{72}{5} = 14.4$

дней – среднее время до возникновения следующей ошибки.

Теперь можно прогнозировать общую длительность процесса тестирования. Она составит  $5 + 7 + 14.4 = 26.4$  дня.

### Заключение

Предлагаемая методика оценки длительности процесса тестирования дополняет методику Джелинского-Моранды, вводя признанную схему оценки числа ожидаемых ошибок. Достоинство данного подхода состоит в том, что оценку значения  $N$  можно сделать в процессе планирования разработки и в процессе разработки ПО еще до того момента, когда написан программный код. При этом можно начать оценивать длительность процесса тестирования уже после обнаружения второй ошибки. Чем дольше уже длится процесс тестирования, тем более точную оценку длительности тестирования можно получить.

### Список литературы

---

1. Иьуду К.А. Надежность, контроль и диагностики вычислительных машин и систем. – М.: Высшая школа, 1989. – 216с.
2. Холстед М.Х. Начала науки о программах / Пер. с англ. Яз. – М.: Финансы и статистика, 1981. – 128с.
3. Антонов А.Г. Выбор конфигурации аппаратной платформы для выполнения алгоритмов заданной сложности.- Компьюлог, №2 (56), 2003, с.8-13.
4. Jones, Capers. "Software Metrics: Good, Bad, and Missing." Computer 27, 9 (September 1994): 98-100.
5. Коваленко С.М., Петухов Г.П. Моделирование программной нагрузки вычислительных систем реального времени.- В кн.: Теоретические вопросы ВТ и программного обеспечения. Межвузовский сб. научных трудов.- М., МИРЭА, 2006, с.2-6.

### References

---

1. Lyudu K.A. Nadezhnost, kontrol i diagnostiki vychislitelnyh mashin i sistem [Reliability, control and diagnostics of computers and systems], Moscow, Vysshaya Shkola, 1989. 216p.
2. Maurice H. Halstead. Elements of Software Science (Operating and programming systems series). Elsevier Science Inc., New York, NY, USA, 1977.127p.
3. Antonov A.G. Vyborkonfiguracii apparatnoj platformy dlya vypolneniya algoritmov zadannoj slozhnosti [The choice of configuration of the hardware platform to execute the algorithms of a given complexity]. Kompyulog, 2003, no. 2 (56), pp.8-13.
4. Jones Capers Software Metrics: Good, Bad, and Missing. Computer27,1994, no. 9, pp. 98-100.
5. Kovalenko S.M.,Petuhov G.P. Modelirovanie programmnoj nagruzki vychislitelnyh sistem realnogo vremeni [Modeling a software load of the computing real-time systems], Teoreticheskie voprosy VT i programmnoho obespecheniya. Mezhvuzovskij sb. nauchnyh trudov, Moscow, MIREA, 2006, pp.2-6.