

Модели оценки надежности электронных обучающих пособий в образовании

Кулягин Виталий Александрович

Сибирский Федеральный Университет

В докладе показаны некоторые особенности использования математического аппарата прикладной теории надежности технических систем для определения показателей надежности программного обеспечения (ПО) в образовании. Предлагается на практике измерить надежность программного продукта «PSN», предназначенного для целей обучения.

Введение

В настоящее время актуальной является проблема исследования надежности программного обеспечения (ПО). В рамках данной проблемы можно выделить более мелких 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.

Основным средством определения показателей надежности являются *модели надежности*, под которыми понимают математическую модель, построенную для оценки зависимости надежности от заранее известных или оцененных в ходе создания ПО параметров.

1 Модели надежности программного обеспечения

Различают модели ПО статические и динамические. Статические модели принципиально отличаются от динамических прежде всего тем, что в них появление отказов не связывают со временем появления ошибок в процессе тестирования, а учитывают только зависимость количества ошибок от числа тестовых прогонов (по области ошибок) или зависимость количества ошибок от характеристики входных данных (по области данных). В динамических же моделях поведение ПО (появление отказов) рассматривается во времени.

Экспоненциальная модель

Рассмотрим одну из распространённых динамических моделей надежности ПО – экспоненциальная модель или модель Шумана [2].

В это методе вводится ряд допущений и условий.

Условия сводятся к следующему:

1. Предполагается, что в начальный момент компоновки программных средств системы в них имеются небольшие ошибки (E – количество ошибок). С этого времени отсчитывается время отладки τ , которое включает затраты времени на выявление ошибок с помощью тестов, на контрольные проверки и т.д. При этом время исправного функционирования системы не учитывается. В течение времени τ устанавливается $\varepsilon_0(\tau)$ ошибок в расчете на одну команду машинного языка. Т.о. удельное число ошибок на одну машинную команду, остающихся в системе после времени τ работы равно

$$\varepsilon_\tau(\tau) = \frac{E}{I} - \varepsilon_0(\tau)$$

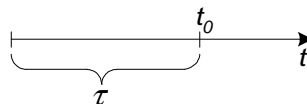
I – общее число машинных команд.

2. Предполагается, что значение функции частоты или интенсивности отказов $z(t)$ пропорциональна числу ошибок, оставшихся в ПО после израсходования на отладку времени τ , то есть

$$z(t) = C \varepsilon_\tau(\tau)$$

C – коэффициент пропорциональности.

Тогда, если время работы системы t отсчитывается от момента времени t_0 , а τ остается фиксированным ($\tau = \text{const}$), то имеем показательное распределение, где случайная величина - время между двумя ближайшими отказами системы. Функция надежности или вероятность безотказной работы на интервале времени от 0 до t есть



$$R(t, \tau) = \exp\{-C \varepsilon_\tau(\tau)t\} = \exp\left\{-C \left[\frac{E}{I} - \varepsilon_0(\tau) \right] t\right\}$$

Для нахождения C и E используются принцип максимального правдоподобия (пропорция).

2 Расчёт надёжности программного продукта «PSN»

В данной статье предлагается фиксировать ошибки и их параметры на стадии эксплуатации, т.е. во время работы пользователей. Была проведена эксплуатация ПО, в результате которой зафиксировано и исправлено 6 ошибок, использована экспоненциальная модель для оценки надежности.

Итак, метод тестирования ПО - «Черный ящик».

Стратегия тестирования - использование случайных входных значений, фиксирование наличия, время появления и тип ошибки.

В ходе тестировочных испытаний было выявлены следующие виды ошибок:

-функциональные (часть функций программы не работает) – 1 ошибка

-неправильный вывод – 3 ошибок

-недостатки в оформлении, представлении чисел – 2 ошибки

В среднем 1 прогон программы пользователем занимает 20 секунд.

Количество прогонов / интервалы времени между ближайшими ошибками (с):

4 / 80, 2 / 40, 4 / 80, 7 / 140, 5 / 100, 28 / 560

Общее время тестирования – 1000 с, общее кол-во прогонов программы – 50.

Используя модель Шумана и применяя метод максимального правдоподобия, получим оценки c и N :

Частота возникновения ошибок прямо пропорциональна количеству оставшихся ошибок в системе:

$$\lambda(t) = c * (E - E_0(t))$$

$$\text{Для } t_1 = 80 \text{ с: } 1 / 80 = c * (N - 0)$$

$$\text{Для } t_2 = 920 \text{ с: } 1/1000 = c * (N - 6)$$

$$E = 6,522, c = 0,0019$$

Рассчитаем надежность системы – в данной модели это вероятность безотказной работы на интервале $(0, t)$, примем $t = 1800$ с

$$R = \exp(-c * (E - E_0(t)) * t)$$

$$R = 0,1687$$

Используя данную модель, мы получили достаточно адекватную оценку надежности рассматриваемого ПО, что говорит о возможности применения данной модели в дальнейших исследованиях.

Литература

1 Майерс, Г. «Надежность ПО», М.: Мир, 1981.

2 Половко, А.М. Основы теории надёжности / А.М. Половко, С.В. Гуров – 2-е изд., перераб. и доп. – Спб.: БХВ-Петербург, 2006