

Фактор социального сопротивления связан с непониманием прямых выгод и преимуществ, создаваемых информатизацией общественной жизни, с нежеланием, недоумением и непониманием необходимости изменять свои жизненные стереотипы, с традиционным человеческим консерватизмом.

Таким образом, исследование в области подготовки пожилых людей в условиях развития информационного общества имеет первостепенное значение для их социальной адаптации, включения в процесс создания цифровой экономики в Республике Беларусь.

ЛИТЕРАТУРА

1. Гринберг А.С., Король И.А., Информационный менеджмент / А.С. Гринберг, И.А.Король - Минск.: Академия управления при Президенте Республики Беларусь – 2006.- 479с.

В.Т.ЛЭ¹, С.С.ДИК¹, С.М.БОРОВИКОВ¹

**МЕТОД ОЦЕНКИ НАДЁЖНОСТИ ПРИКЛАДНЫХ ПРОГРАММНЫХ СРЕДСТВ
НА РАННИХ ЭТАПАХ ИХ РАЗРАБОТКИ**

¹Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники», г. Минск, Республика Беларусь

В сложных информационно-компьютерных системах, к которым относятся системы телекоммуникаций, вклад программного обеспечения в ненадёжность систем составляет до 40 процентов. По мнению некоторых авторов этот вклад может превышать вклад, вносимый техническими средствами (компьютерами), поскольку входные данные могут быть сложнее и их формат всё время меняется [1]. Надёжность аппаратуры ограничивается ошибками проектирования, производственными дефектами и частотой сбоев (зависит от физических процессов). По своей природе программное обеспечение сложнее технических средств систем. Объём программных средств (далее – ПС) для современных информационно-компьютерных систем оценивается в $10^6 \dots 10^8$ и более команд или информационных слов.

Современное программное обеспечение довольно сложное, и есть все предпосылки считать, что оно станет ещё сложнее в ближайшем будущем. Например, в 1983 году программа Microsoft Word состояла только из 27 000 строк кода, но, согласно данным Nathan Myhrvold [1], к 1995 году эта программа увеличилась уже до 2 млн. строк кода! Программисты потратили годы на то, чтобы придумать единицы измерения для программного обеспечения. Но только одна единица измерения позволяет установить соотношение с числом ошибок – количество строк кода (LOC). И действительно, в некоторых кругах специалистов по программированию число строк кода стало единственным приемлемым средством измерения объёма программных средств [1].

Количество ошибок на тысячу строк кода (KLOC) изменяется для каждой конкретной программы. Достоверное значение варьируется от 5 до 50 ошибок на 1000 строк кода [2]. В программе, которая прошла тестирование только на предмет работоспособности функциональных возможностей, что справедливо для большей части коммерческого программного обеспечения, присутствует намного больше ошибок – около 50 ошибок на 1000 строк кода. Большая часть программ попадает в последнюю категорию. Чтобы оценить всю сложность современного программного обеспечения, рекомендуется проанализировать информацию, приведённую в таблице 1.

Таблица 1 – Объём программного обеспечения

Программное средство (объект)	Космическая станция	Космический корабль	Boeing 777	Windows NT5	Linux	Windows 95	Windows XP
Количество строк кода, млн	40	10	7	35	1,5	5	40

Применительно к надёжности программных средств ошибка – это погрешность или искажение кода программы, неумышленно внесённое в неё в процессе разработки, которые при определённом наборе входных данных в ходе функционирования этой программы могут вызвать отказ или снижение эффективности функционирования ПС.

Надёжность ПС определяется качеством отладки программы, глубиной её тестирования. Целью тестирования является не тотальное обнаружение всех ошибок, что принципиально невозможно, а выявление наибольшего количества наиболее критичных ошибок.

При отладке ПС происходит локализация и устранение синтаксических ошибок и явных ошибок кодирования. В процессе же тестирования проверяется работоспособность программы, не содержащей явных ошибок. С помощью тестирования должно быть выявлено как можно больше смысловых ошибок с учётом возможного формата изменения исходных данных. Тестирование требует значительного времени, и даже после его завершения некоторые ошибки в ПС остаются необнаруженными.

Известные методы оценки надёжности прикладных ПС исходят из того, что имеются определённые данные о тестировании уже написанного ПС с использованием кода и устранены ошибки языка программирования, т.е. выполнена отладка программы. Однако в большинстве случаев проектировщиков информационно-компьютерных систем и разработчиков программного обеспечения для анализа с помощью этих систем интересует ожидаемый уровень надёжности прикладных ПС ещё до написания их программного кода. Возникает вопрос, как спрогнозировать ожидаемый уровень надёжности ПС на этом этапе.

На сегодняшний день модель Холстеда является единственной, используя которую можно еще на этапе разработки технического задания на ПС оценить количество ожидаемых в нём ошибок $N_{\text{ош}}$ до начала этапа тестирования. При использовании этой модели пользуются такими понятиями как [2]

- количество различных операций;
- количество различных переменных и констант.
- количество входных и выходных переменных;
- общее количество повторов $n_{\text{пк}}$;
- потенциальный объём программы (в битах);
- длина самой программы;
- объём программы (измеряемый в битах).

Определив число ошибок в программе $N_{\text{ош}}$, далее можно воспользоваться гипотезой о том, что интенсивность проявления ошибки ПС прямо пропорциональна количеству оставшихся ошибок в программном средстве $N_{\text{ош}}$ [3]:

$$\lambda_{\text{эсп}} = (N_{\text{ош}} - N_{\text{тест}})C, \quad (1)$$

где $\lambda_{\text{эсп}}$ – интенсивность проявления ошибок (интенсивность отказов), соответствующая начальному этапу эксплуатации программного средства; $N_{\text{тест}}$ – прогнозное число обнаруживаемых и исправляемых ошибок при тестировании; C – коэффициент пропорциональности.

Величины $N_{\text{тест}}$ и C зависят от длительности процедуры тестирования и квалификации тестировщиков. Для оценки (прогнозирования) их значений пригодны статистические модели, рассмотренные в [3], принимающие во внимание количество строк кода (LOC) программы.

ЛИТЕРАТУРА

1. Программное обеспечение – источник всех проблем. [Электронный ресурс]. – 2018. – Режим доступа: <http://www.williamspublishing.com/PDF/5-8459-0785-3/part1.pdf>
2. Методы обеспечения аппаратно-программной надёжности вычислительных систем. Д.т.н., проф. Чуканов В.О., к.т.н., доц. Гуров В.В. [Электронный ресурс]. – 2018. – Режим доступа : http://www.mcst.ru/files/5357ec/dd0cd8/50af39/000000/seminar_metody_obespecheniya_apparatno-programmnoy_nadezhnosti_vychislitelnyh_sistem.pdf
3. Боровиков, С.М. Прогнозирование ожидаемой надёжности прикладных программных средств с использованием статистических моделей их безотказности / С.М. Боровиков, С.С. Дик // BIG DATA and Advanced Analytics: сб. материалов IV Междунар. науч.-практ. конф. (Минск, Республика Беларусь, 3–4 мая 2018 года) / редкол. : М.П. Батура [и др.]. – Минск : БГУИР, 2018. – С. 348–354.