

ИНФОРМАТИКА, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И УПРАВЛЕНИЕ

ВОЗМОЖНЫЙ ПОДХОД К ОЦЕНКЕ НАДЁЖНОСТИ РАЗРАБАТЫВАЕМЫХ ПРОГРАММНЫХ СРЕДСТВ НА РАННИХ ЭТАПАХ ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННО-КОМПЬЮТЕРНЫХ СИСТЕМ

Боровиков Сергей Максимович

*кандидат технических наук, доцент,
Белорусский государственный университет
информатики и радиоэлектроники, город Минск*

Дик Сергей Сергеевич

*аспирант 3-го года обучения,
Белорусский государственный университет
информатики и радиоэлектроники, город Минск*

Лэ Ван Там

*магистрант 2-го года обучения,
Белорусский государственный университет
информатики и радиоэлектроники, город Минск*

Клинов Константин Игоревич

*студент 5-го курса,
Белорусский государственный университет
информатики и радиоэлектроники, город Минск*

Аннотация. Для определения ожидаемой надёжности прикладных программных средств на ранних этапах их разработки (до написания кода компьютерных программ) предлагается подход, при котором используются некоторые параметры будущей программы и данные о влиянии на её надёжность различных факторов. Получена модель для определения ожидаемой надёжности компьютерной программы после окончания предполагаемого её тестирования.

Abstract. To determine the expected reliability of applied software at the early stages of their development (before writing code for computer programs), the authors propose an approach that uses some parameters of a future program and data on the influence of various factors on its reliability. A model has been obtained for determining the expected reliability of a computer program after the end of its intended testing.

Ключевые слова: прикладные программные средства, ранние этапы разработки, прогнозируемая плотность ошибок, ожидаемое число ошибок, интенсивность отказов программного средства.

Keywords: application software, early stages of development, predicted fault density, expected number of errors, software failure rate.

Введение

В общем случае надёжность информационно-компьютерных систем определяется надёжностью следующих трёх функциональных частей:

- комплекса технических средств, включающего компьютеры, датчики, исполнительные устройства и другую аппаратуру;
- программного обеспечения, представляющего собой совокупность компьютерных программ (системных и прикладных);
- оперативного персонала, функции которого выполняет человек-оператор.

Если отсутствует резервирование, то с точки зрения надёжности эти части системы соединены последовательно. Это означает: отказ любой указанной части приводит к отказу системы в целом, что символизируется разрывом цепи на структурной схеме надёжности информационно-компьютерной системы (рисунок 1). Эту схему можно рассматривать как модель надёжности системы. На схеме функциональные части системы обозначены прямоугольниками, в которых указывается примерный вклад соответствующей функциональной части в ненадёжность информационно-компьютерной системы.

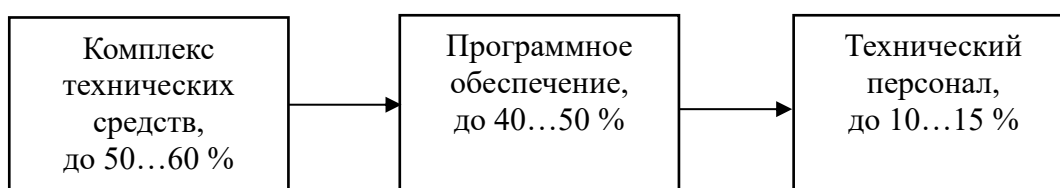


Рисунок 1 – Структурная схема надёжности информационно-компьютерной системы

На этапе проектирования информационно-компьютерной системы важно оценить её ожидаемую надёжность с учётом наличия трёх составных функциональных частей.

Актуальность разработки

Надёжность комплекса технических средств ограничивается ошибками проектирования, производственными дефектами и частотой сбоев (зависит от физических процессов). Методы оценки надёжности технических устройств достаточно хорошо разработаны (см., например [5, 2]). Достигнут также определённый уровень в разработке методов и подходов в оценке надёжности оператора при его работе в составе технических систем (см., например [3]).

В сложных информационно-компьютерных системах, к которым относятся, например, системы медицинского назначения, электронные системы обеспечения безопасности, вклад программного обеспечения в ненадёжность систем может составлять до 40 % и более. Специалистами проделана немалая работа по определению ожидаемой надёжности прикладных компьютерных программ, однако многие разработанные модели и методы носят ограниченный характер. Модели предполагают наличие данных о результатах тестирования разработанной компьютерной программы после устранения в ней ошибок, вызванных нарушениями правил языка программирования.

Проектировщиков информационно-компьютерных систем и разработчиков программного обеспечения для этих систем в большинстве случаев интересует ожидаемый уровень надёжности прикладных компьютерных программ ещё до написания их программного кода. В [1] авторами был предложен метод оценки надёжности прикладных программ на ранних этапах их разработки. Метод основан на использовании статистических моделей ожидаемого числа ошибок в компьютерной программе. Достоверность результатов, которые обеспечивает метод, в значительной степени зависит от используемых коэффициентов, характеризующих опыт и квалификацию программистов. Определение значений этих коэффициентов создаёт проблемы при оценке ожидаемой надёжности компьютерных программ и информационно-компьютерных систем в целом. Это обуславливает необходимость совершенствования подходов и методов оценки надёжности прикладного программного обеспечения на ранних этапах его разработки.

Подход к оценке ожидаемой надёжности программных средств

В соответствии с ГОСТ 27.205-1997 [3] о проектной надёжности разрабатываемой компьютерной программы (программного средства) будем судить по интенсивности проявления скрытых ошибок, оставшихся в компьютерной программе после её разработки и тестирования.

Очевидным является то, что реальный уровень надёжности компьютерной программы зависит от её объёма. Специалисты по программированию пришли к выводу [6, с. 12], что в качестве единицы измерения объёма программ лучше всего использовать количество строк кода (в англоязычном варианте: Lines Of Code – LOC).

Для ориентировочной оценки ожидаемой надёжности прикладных компьютерных программ на ранних этапах их разработки (до написания кода программ) авторами предлагается подход, при котором используются некоторые параметры будущей программы и данные о влиянии на её надёжность различных факторов. Подход основан на следующих основных гипотезах и допущениях:

- интенсивность проявления ошибок в программе (обозначим через λ) прямо пропорциональна числу оставшихся в ней ошибок $N_{\text{ош}}$;
- значения поправочных коэффициентов, учитывающих влияние конкретного фактора (характеристики программы) на интенсивность проявления ошибки λ , прямо пропорциональны трудозатратам написания кода программы;
- поправочные коэффициенты и/или метрики, учитывающие влияние на интенсивность проявления ошибки λ уровня квалификации и опыта программистов, оцениваются экспертным методом.

Основные результаты

С учётом работ [1, 4] и технического отчёта [8] для оценки ожидаемой надёжности прикладной компьютерной программы до выполнения её тестирования может быть использована модель

$$\lambda_0 = K_{\text{пр}} \frac{N_{\text{ош}}}{L} = K_{\text{пр}} F, \quad 1)$$

где λ_0 – ожидаемая начальная интенсивность проявления ошибок компьютерной программы (размерность: 1/ч); $K_{\text{пр}}$ – коэффициент, показывающий, как дефекты проектирования компьютерной программы преобразуются (трансформируются) в интенсивность проявления ошибок при использовании программы по назначению (размерность: строка кода / ошибок в час); $N_{\text{ош}}$ – число оставшихся ошибок в компьютерной программе; L – объём текста кода компьютерной программы в строках кода; F – ожидаемая плотность оставшихся ошибок в компьютерной программе (размерность: ошибок/строка кода).

Коэффициент $K_{\text{пр}}$ зависит от области применения компьютерной программы и выбор его значений может быть сделан, используя данные таблицы 1, приводимой в [8, с. 100-7].

Таблица 1

**Коэффициент преобразования плотности ошибок в интенсивность их проявления ($K_{пр}$),
базовая плотность ошибок проектирования (A)**

Область применения компьютерной программы	Коэффициент преобразования $K_{пр}$ (строка кода/ ошибок в час)	Значение A (ошибок/ строка кода)
1. Электронные системы, используемые в авиации	6,28	0,0128
2. Системы мониторинга и обеспечения безопасности	1,2	0,0092
3. Телекоммуникационные системы, мобильные электронные устройства	13,8	0,0078
4. Управление производственными процессами	3,8	0,0018
5. Автоматизированные системы управления (в различных областях и сферах деятельности людей)	23	0,0085
6. Инструменты разработки программного обеспечения, моделирование, испытательные стенды, обучение	16,9	0,0123
Среднее	10,6	0,0087

Прогнозируемую плотность ошибок F выражения (1) в соответствии с моделью RL-92-52 [8, с. 100-3], основанной на использовании метрик программного обеспечения, можно определить как произведение

$$F = A \cdot D \cdot S, \quad (2)$$

где A – средняя или базовая плотность ошибок для компьютерных программ, используемых в данной отрасли, области применения (см. таблицу 1); D – метрика, учитывающая производственную среду разработки программного обеспечения: особенность организации, разрабатывающей компьютерную программу, характеристику группы программистов, их квалификацию; S – метрика, учитывающая характеристики компьютерной программы.

Метрика D выражения (2), по мнению авторов, может быть представлена в виде произведения

$$D = K_{орг} \cdot K_{кв.прог}, \quad (3)$$

где $K_{орг}$ – коэффициент, характеризующий особенность организации и группы программистов, разрабатывающей компьютерную программу; $K_{кв.прог}$ – коэффициент, учитывающий квалификацию и опыт программистов.

В качестве рекомендуемых значений коэффициента $K_{орг}$ могут быть взяты данные, приводимые в техническом отчёте [8, с. 102-3, В-8] о метрике, учитывающей среду разработки программного обеспечения (таблица 2). В случае характеристики группы программистов, заметно отличающейся от указанных в таблице 2, можно дать экспертную оценку значению этого коэффициента, принимая во внимание условие, указанное в документе RL-92-52 [8, с. 102-3]:

$$0,5 \leq K_{орг} \leq 2.$$

Таблица 2

Рекомендуемые значения коэффициента $K_{орг}$

Характеристика группы программистов, разрабатывающей компьютерную программу	Значение $K_{орг}$
1. Группа программистов работает в организации, обеспечивает её потребности в программном обеспечении и отвечает за программу	0,76
2. Группа программистов имеет опыт работы по разработке программ, но не связана с пользователем программы	1,00
3. Группа программистов обладает опытом работы с компьютером, но может быть не знакома с программным обеспечением, для которого разрабатывается компьютерная программа. Программа должна использоваться во взаимосвязанном комплексе технических средств, программного обеспечения и действий оператора	1,30

Значения коэффициента $K_{\text{кв.прог}}$, полученные методом экспертной оценки на кафедре проектирования информационно-компьютерных систем Белорусского государственного университета информатики и радиоэлектроники, приведены в таблице 3.

Таблица 3

Значения коэффициента $K_{\text{кв.прог}}$	
Квалификация и опыт программиста	Значение $K_{\text{кв.прог}}$ *
1. Студент, освоивший программирование на уровне программы учебной дисциплины высшего технического учебного заведения	2,0
2. Младший программист (<i>Junior Development</i>)	1,3
3. Программист (<i>Middle Development</i>)	1,0
4. Ведущий программист (<i>Senior Development</i>)	0,7
Примечание: * – получено методом экспертной оценки	

Для определения метрики S выражения (3) предлагается модель вида

$$S = K_{\text{слож}} \cdot K_{\text{С.Р}} \cdot K_{\text{нов}} \cdot K_{\text{мод}}, \quad (4)$$

где $K_{\text{слож}}$ – коэффициент, учитывающий категорию сложности компьютерной программы; $K_{\text{С.Р}}$ – коэффициент, учитывающий используемые средства разработки компьютерной программы; $K_{\text{нов}}$ – коэффициент, учитывающий степень новизны компьютерной программы; $K_{\text{мод}}$ – коэффициент, характеризующий степень использования стандартных модулей в компьютерной программе.

Ниже в качестве примера выбора поправочных коэффициентов модели (4) показано, как выбирать значения коэффициента $K_{\text{слож}}$, учитывающего категорию сложности компьютерной программы, и коэффициента $K_{\text{мод}}$, учитывающего степень использования стандартных модулей в разрабатываемой компьютерной программе. Для выбора значений этих коэффициентов использованы данные, приводимые в документе [7] с учётом гипотезы о том, что число возможных ошибок в компьютерной программе прямо пропорционально затратам труда на её разработку.

В таблице 4 приводится описание основных характеристик, которые рекомендуется принимать во внимание при оценке степени сложности разрабатываемой компьютерной программы.

Таблица 5 показывает, как оценивать степень сложности разрабатываемой компьютерной программы, используя характеристики сложности, указанные в таблице 4, и выбирать на основе этого значения коэффициента $K_{\text{слож}}$.

Таблица 4

Характеристики сложности компьютерной программы	
Номер	Описание характеристики
1	Наличие сложного интеллектуального языкового интерфейса с пользователем
2	Обеспечение телекоммуникационной обработки данных и управление удалёнными объектами
3	Обеспечение существенного распараллеливания вычислений
4	Криптография и другие методы защиты информации
5	Реализация особо сложных инженерных и научных расчётов
6	Моделирование объектов и процессов
7	Обеспечение настройки компьютерной программы на изменения структур вход-выходных данных
8	Обеспечение переносимости компьютерной программы
9	Интерактивный доступ
10	Машинная графика, многомашинные комплексы
11	Нестандартная конфигурация технических средств
12	Функционирование компьютерной программы в расширенной операционной среде (связь с другими компьютерными программами)
13	Обеспечение хранения, ведения и поиска данных в сложных структурах
14	Режим работы в реальном времени

Таблица 5

Значение коэффициента $K_{\text{слож}}$		
Степень сложности компьютерной программы	Описание сложности компьютерной программы с учётом наличия характеристик, приведённых в таблице 4	Значение $K_{\text{слож}}$
1. Особо сложные	Более 7 характеристик	1,47
2. Сложные	5 или 6 характеристик	1,39
3. Заметной сложности	3 или 4 характеристики	1,26
4. Умеренной сложности	1 или 2 характеристики	1,15
5. Низкой сложности	Отсутствуют характеристики, перечисленными в таблице 4	1

В таблице 6 указываются значения коэффициента $K_{\text{мод}}$, учитывающего степень использования стандартных модулей в разрабатываемой компьютерной программе.

Таблица 6

Выбор коэффициента $K_{\text{мод}}$	
Степень охвата реализуемых функций стандартными модулями в разрабатываемой компьютерной программе	Значение $K_{\text{мод}}$
1. От 60% и выше	0,55
2. От 40 до 60%	0,65
3. От 20 до 40%	0,77
4. До 20%	0,9
5. Не используются стандартные модули	1

Используя документ [7], аналогичным способом могут быть получены значения коэффициентов $K_{\text{с.р}}$ и $K_{\text{нов}}$, которые принимают во внимание используемые средства разработки компьютерной программы ($K_{\text{с.р}}$) и степень новизны разрабатываемой компьютерной программы ($K_{\text{нов}}$).

Рассмотренный подход позволяет определить ожидаемую надёжность компьютерной программы до начала выполнения её тестирования: значение λ_0 модели (1).

Процедура тестирования компьютерной программы позволяет выявить часть имеющихся в программе ошибок и тем самым уменьшить интенсивность проявления ошибок на этапе эксплуатации программы. Для определения ожидаемой надёжности компьютерной программы после выполнения её тестирования будем предполагать, что процедура тестирования выполняется общепринятыми методами, и при этом её суммарная продолжительность составляет не менее 40 процентов времени, отводимого на разработку компьютерной программы. Согласно работе [8, с. 300-3], экспериментальные данные показали, что интенсивность проявления ошибок компьютерной программы за счёт выполнения тестирования уменьшается примерно в семь раз относительно начальной интенсивности проявления ошибок λ_0 , получаемой по модели (1). Следовательно, для определения ожидаемой надёжности компьютерной программы после окончания её тестирования, то есть на начальном этапе эксплуатации программы может быть использована модель

$$\lambda_{\text{эсп}} = 0,14K_{\text{тр}} \cdot A \cdot D \cdot S, \quad (5)$$

где $\lambda_{\text{эсп}}$ – интенсивность проявления ошибок на начальном этапе эксплуатации компьютерной программы, прошедшей тестирование; A – базовая плотность ошибок для компьютерных программ, используемых в данной отрасли; D и S – величины, определяемые по выражениям (3) и (4); множитель 0,14 – коэффициент, учитывающий уменьшение интенсивности проявления ошибок в семь раз за счёт процедуры тестирования компьютерной программы ($1/7 \approx 0,14$).

Модель вида (5) получена по данным исследования программных средств, выполняемых на компьютерах с быстродействием процессора 10 миллионов операций в секунду (10 Million Instructions Per Second) [8, с. 300-3]. Если компьютер, используемый в составе проектируемой системы, имеет другое быстродействие, то модель (5) должна быть скорректирована с учётом этого быстродействия. Решение такой задачи является предметом отдельного рассмотрения и в данной статье не обсуждается.

Выводы

Предложенный подход даёт возможность приближённо оценить ожидаемую надёжность разрабатываемого программного средства. Но даже такой ориентировочный расчёт полезен, так как позволяет на раннем этапе разработки программного средства получить представление о его надёжности и сориентироваться в целесообразности проектирования и/или эффективности применения информационно-компьютерной системы, использующей это программное средство. При необходимости могут быть разработаны организационно-технические мероприятия по обеспечению требуемого уровня надёжности программного средства и информационно-компьютерной системы в целом.

Список литературы

1. Боровиков С. М., Лэ Ван Там, Дик С. С. Возможный подход к оценке надёжности прикладных программных средств для технологий Big Data // BIG DATA и анализ высокого уровня : сб. материалов V Междунар. науч.-практ. конф. (Республика Беларусь, Минск, 13–14 марта 2019 года). В 2 ч. Ч. 2. Минск : БГУИР, 2019. С. 77-83.
2. Боровиков С. М., Цырельчук И. Н., Троян Ф. Д. Расчёт показателей надёжности радиоэлектронных средств. Минск: БГУИР, 2010. 68 с.
3. ГОСТ 27.205-1997. Надёжность в технике. Проектная оценка надёжности сложных систем с учётом технического и программного обеспечения и оперативного персонала. Основные положения. Минск : Госстандарт Республики Беларусь, 2005. 22 с.
4. Модели оценки надёжности программных средств [Электронный ресурс]. Режим доступа : <https://helpiks.org/4-73503.html> (дата обращения: 15.02.2020).
5. Надёжность технических систем: справочник / Под ред. И. А. Ушакова. М.: Радио и связь, 1985. 608 с.
6. Программное обеспечение – источник всех проблем. [Электронный ресурс]. URL: <http://www.williamspublishing.com/PDF/5-8459-0785-3/part1.pdf> (дата обращения: 15.02.2020).
7. Укрупнённые нормы затрат труда на разработку программного обеспечения. Постановление Министерства труда и социальной защиты Республики Беларусь 27.06.2007 № 91 [Электронный ресурс]. Режим доступа : https://belzakon.net/Законодательство/Постановление_Министерства_труда_и_социальной_защиты_РБ/2007/71737 (дата обращения: 15.02.2020).
8. Software reliability, measurement and testing guidebook for software reliability measurement and testing: RL-TR-92-52, Vol II (of two) Final technical report April 1992/ Science Applications International Corp. (SAIC). Rome Laboratory Air Force Systems Command Griffiss Air Force Base NY 13441-5700.