

УДК 681.3.06

## **Toward the Problem of Estimation of the Complex Software System Reliability**

**Roman Yu. Tsarev\***,  
**Elena N. Shtarik and Andrey V. Shtarik**  
*Siberian Federal University*  
*79 Svobodny, Krasnoyarsk, 660041, Russia*

Received 17.09.2014, received in revised form 21.01.2015, accepted 04.02.2015

---

*Article is devoted to the problem of software reliability estimation. The developed algorithm allows applying universal model for estimation the reliability parameters of software systems with different architectures. The modification of the model for estimation of the reliability of distributed multiversion software systems is given. Based on the proposed models and algorithm the program complex is developed. The paper presents the research of the dependence of various reliability parameters of a software system from the conditional and unconditional probability of failure of individual system components, as well as from the number of components of the software system.*

*Keywords: reliability, estimation, software system, multiversion software, distributed system.*

---

## **К проблеме оценки надежности сложных программных систем**

**Р.Ю. Царев, Е.Н. Штарик, А.В. Штарик**  
*Сибирский федеральный университет*  
*Россия, 660041, Красноярск, пр. Свободный, 79*

---

*Статья посвящена проблеме оценки надежности программного обеспечения. Разработан алгоритм, который позволяет применить универсальную модель при оценке надежности программных систем с различной архитектурой. Представлена модификация модели для оценки надежности распределенных мультиверсионных программных систем. На основе предложенного модельно-алгоритмического аппарата разработан программный комплекс, предназначенный для оценки надежности программных систем. В статье проведено исследование зависимости различных показателей надежности программной системы от условной и безусловной вероятности сбоев отдельных компонентов системы, а также количества компонентов программной системы.*

*Ключевые слова: надежность, оценка, программная система, мультиверсионное программное обеспечение, распределенная система.*

---

© Siberian Federal University. All rights reserved

\* Corresponding author E-mail address: tsarev.sfu@mail.ru

## 1. Введение

Интерес к оценке надежности программного обеспечения (ПО) возник одновременно с появлением самого ПО. За прошедшие десятилетия было создано множество методов, методик и моделей исследования надежности ПО [1]. Однако единого подхода к решению этой проблемы предложено не было и, по-видимому, в ближайшее время не предвидится. Тем не менее при разработке сложных программных систем создатели стараются в той или иной степени получить оценку их надежности. Один из наиболее эффективных подходов заключается в последовательном оценивании надежности программ на каждом этапе их разработки [2]. Основная сложность при использовании статистических методов заключается в отсутствии достаточного количества исходных данных. Динамика выявления ошибок должна тщательно фиксироваться и обрабатываться. Важной проблемой выступает степень детализации элемента расчета надежности [3, 4]. Выявить все траектории выполнения программы при обработке информации (как это порой предлагается), даже для достаточно несложной программы, практически нереально [5, 6]. Исходя из этого детализация элементов расчета надежности (условно называемых программными модулями) должна ограничиваться законченными программными образованиями, которые, взаимодействуя между собой, составляют более сложное объединение (систему), чья надежность нас интересует. При этом допускается, что аппаратное обеспечение, операционная система и среда программирования абсолютно надежны.

В результате анализа работ зарубежных исследователей, таких как А. Авиженис [7], Б. Боэм [8], и трудов отечественных авторов: В.В. Липаева [3, 9], Г.Н. Черкесова [10], И.В. Ковалева [11] и других в области исследования надежности ПО можно выделить три основные группы проблем в данной области:

- отсутствие единой методологии создания высоконадежных программных систем;
- отсутствие единой универсальной методологии тестирования программных систем;
- отсутствие единого подхода к оценке и анализу надежности программных систем.

Одним из путей решения обозначенных проблем является использование моделей оценки параметров надежности программных систем, представленных в данной работе. Обобщенная модульная интегрально-вероятностная модель оценки параметров надежности и ее модификация для мультиверсионных программных систем с распределенной архитектурой адаптированы к современным методам анализа и разработки программного обеспечения. В частности, рассмотрена возможность применения моделей для объектно-ориентированного подхода.

## 2. Обобщенная модульная интегрально-вероятностная модель оценки надежности программных систем

Для оценки параметров надежности программных систем разработана следующая обобщенная модульная интегрально-вероятностная модель оценки надежности. Для данной модели обязательно выполнение условия

$$\sum_{i=1}^F PU_i = 1,$$

где  $F$  – общее число компонентов в архитектуре программной системы;  $PU_i$  – вероятность использования компонента  $i$ ,  $i = 1, \dots, F$ .

Среднее время простоя программного обеспечения вычисляется как

$$\begin{aligned}
 TR = & \sum_{i=1}^F [PU_i \times PF_i \times [TA_i + TC_i + TE_i + \\
 & + \sum_{j=1, j \neq i}^F [PL_{ji} \times [TA_j + TC_j + TE_j + \\
 & + \sum_{l, l \in D} [PL_{lj} \times (TA_l + TC_l + TE_l)]]] + \\
 & + \sum_{k, k \in D} PL_{ki} \times [TA_k + TC_k + TE_k + \\
 & + \sum_{m=1, m \neq j}^F [PL_{mk} \times [TA_m + TC_m + TE_m + \\
 & + \sum_{l, l \in D} PL_{lm} \times (TA_l + TC_l + TE_l)]]]]],
 \end{aligned} \tag{1}$$

где  $M$  – число уровней архитектуры программной системы;  $PU_i$  – вероятность использования компонента  $i$ ,  $i = 1, \dots, F$ ;  $PF_i$  – вероятность сбоя в компоненте  $i$ ,  $i = 1, \dots, F$ ;  $PL_{ij}$  – условная вероятность сбоя в компоненте  $i$  при появлении сбоя в компоненте  $j$ ,  $i = 1, \dots, F$ ,  $j = 1, \dots, F$ ;  $TA_i$  – относительное время доступа к компоненту  $i$ ,  $i = 1, \dots, F$ ;  $TC_i$  – относительное время анализа сбоя в компоненте  $i$ ,  $i = 1, \dots, F$ ;  $D_{mj}$  – непересекающиеся множества компонент  $j$  на уровне  $m$ ,  $m = 1, \dots, M$ ,  $j = 1, \dots, F$ ;  $TE_i$  – относительное время устранения сбоя в компоненте  $i$ ,  $i = 1, \dots, F$ .

Среднее время появления сбоя вычисляется как

$$\begin{aligned}
 MTTF = & \sum_{i=1}^F [PU_i \times (1 - PF_i) \times [TU_i + \sum_{j=1, j \neq i}^F [(1 - PL_{ji}) \times [TU_j + \\
 & + \sum_{l, l \in D} [(1 - PL_{li}) \times TU_l + \sum_{k, k \in D} [(1 - PL_{ki}) \times [TU_k + \\
 & + \sum_{m=1, m \neq i}^F [(1 - PL_{mk}) \times [TU_m + \sum_{l, l \in D} [(1 - PL_{lm}) \times TU_l]]]]]]]]],
 \end{aligned} \tag{2}$$

где  $TU_i$  – относительное время использования компонента  $i$ ,  $i = 1, \dots, F$ .

Коэффициент готовности программной системы вычисляется как

$$S = MTTF / (MTTF + TR).$$

Вероятность безотказной работы программного обеспечения вычисляется как

$$R_S = \sum_{i=1}^F PU_i \times R_i, \text{ где } R_i = 1 - \prod_{k \in Z_i} PF_{ik}, \tag{3}$$

где  $R_i$  – вероятность безотказной работы компонента  $i$ ,  $i = 1, \dots, F$ ;  $Z_i$  – множество версий компонента  $i$ ,  $i = 1, \dots, F$ .

Стоимость разработки программного обеспечения вычисляется как

$$C_s = \sum_{i=1}^F \sum_{j \in Z} C_j,$$

где  $C_i$  – стоимость разработки компонента  $i$ ,  $i = 1, \dots, F$ .

### 3. Алгоритм применения обобщенной модульной интегрально-вероятностной модели оценки надежности программных систем

Шаг 1. Разделить оцениваемую программную систему на модули, определить границы модулей, их характеристики и порядок взаимодействия.

Шаг 2. Определить количество уровней в архитектуре. Если архитектура многоуровневая, перейти к шагу 4, иначе – шаг 3.

Шаг 3. Исключить из модели  $D_{mj}$  в формулах (1) и (2). Затем перейти к шагу 4.

Шаг 4. Определить количество версий. Если архитектура мультиверсионная, перейти к шагу 6, иначе – шаг 5.

Шаг 5. Исключить из модели  $Z_i$  в формуле (3). Затем перейти к шагу 6.



Рис. 1. Блок-схема алгоритма применения обобщенной модульной интегрально-вероятностной модели оценки надежности программных систем

Шаг 6. Определить присутствует ли возможность устранения сбоев. Если нет, перейти к шагу 8, иначе – шаг 7.

Шаг 7. Исключить из модели  $TC_i$ ,  $TE_i$  в формуле (1). Затем перейти к шагу 8.

Шаг 8. Получить итоговые выражения  $R$ ,  $TR$  и  $MTTF$ , решая формулы (1)-(3).

Блок-схема алгоритма применения обобщенной модульной интегрально-вероятностной модели оценки надежности программных систем представлена на рис. 1.

#### **4. Модульная интегрально-вероятностная модель оценки надежности распределенных мультиверсионных программных систем**

Сложность использования обобщенной модульной интегрально-вероятностной модели на фазе дизайна архитектуры программной системы состоит в том, что далеко не всегда известны все необходимые параметры [12]. Если надежность компонента заранее неизвестна, то ее можно оценить только на фазе кодирования. Более точную информацию о надежности можно получить на стадии тестирования компонентов. Вероятность использования компонента и вероятность сбоя в компоненте можно получить только после тестирования программной системы. Для распределенных мультиверсионных программных систем такие параметры, как время доступа, анализа и восстановления компонент, можно оценить также только после тестирования, что не исключает формирования структуры архитектуры будущей программной системы на концептуальной фазе. Для программных систем, реализуемых на основе объектно-ориентированного подхода, можно построить дерево иерархии классов и методов классов. В общем случае на этой стадии необходимо задать параметры, которые требуется оценить на последующих фазах.

С точки зрения объектно-ориентированного подхода вычислительный процесс – это последовательная совокупность вызовов методов классов [13]. Для такого варианта количество уровней архитектуры равно одному. Такие параметры, как время доступа, анализа и восстановления, характерны для распределенной мультиверсионной программной системы.

Рассмотрим подробнее модификацию обобщенной модульной интегрально-вероятностной модели для случая объектно-ориентированной распределенной мультиверсионной программной системы.

Для объектно-ориентированного подхода архитектура программной системы – совокупность иерархий классов. Каждый класс – совокупность свойств (переменных) и методов (функций) объекта. Процесс – совокупность переходов от метода одного класса к методу другого класса [6].

Для данной модели обязательно выполнение условия

$$\sum_{i=1}^F PU_i = 1,$$

где  $F$  – общее число компонентов (классов) в архитектуре программной системы;  $PU_i$  – вероятность использования компонента  $i$ ,  $i = 1, \dots, F$ .

Вероятность безотказной работы мультиверсионного компонента зависит от вероятности безотказной работы каждой версии и метакласса, реализующего механизм мультиверсионности:

$$R_i = (1 - \prod_{k \in Z_i} PF_{ik}) R_{mul},$$

где  $R_i$  – вероятность безотказной работы компонента  $i$ ,  $i = 1, \dots, F$ ;  $Z_i$  – множество версий компонента  $i$ ,  $i = 1, \dots, F$ ;  $R_{mul}$  – вероятность безотказной работы метакласса, реализующего механизм мультиверсионности. Отметим, что метакласс не должен рассматриваться как компонент архитектуры и должен быть исключен из расчетов  $TR$ ,  $MTTF$  и  $R_s$ .

Среднее время простоя программной системы вычисляется как

$$TR = \sum_{i=1}^F [PU_i \times PF_i \times [(TA_i + TC_i + TE_i) + \sum_{j=1, j \neq i}^F [PL_{ji} [(TA_j + TC_j + TE_j)]]]].$$

Среднее время появления сбоя вычисляется как

$$MTTF = \sum_{i=1}^F [PU_i \times (1 - PF_i) \times [TU_i + \sum_{j=1, j \neq i}^F [(1 - PL_{ji}) \times TU_j]].$$

Коэффициент готовности программной системы вычисляется как

$$S = MTTF / (MTTF + TR).$$

Вероятность безотказной работы программной системы вычисляется как

$$R_s = \sum_{i=1}^F PU_i \times R_i, \text{ где } R_i = 1 - \prod_{k \in Z_i} PF_{ik}.$$

Так как предлагаемый подход не учитывает условную вероятность сбоя в компонентах, то в реализации модели также была использована следующая его модификация:

$$R_s = \sum_{i=1}^F PU_i \times R_i \times \prod_{j=1, j \neq i}^F [PU_j \times (1 - R_j \times PL_{ji})].$$

Рассмотрим программную реализацию комплекса оценки надежности распределенных мультиверсионных программных систем на основе представленной модели.

## 5. Комплекс оценки надежности распределенных мультиверсионных программных систем

Модульная интегрально-вероятностная модель оценки надежности распределенных мультиверсионных программных систем реализована в виде программного комплекса на языке C# [14].

Эксплуатационное назначение комплекса:

- обеспечение пользователя информацией о показателях надежности проектируемой программной системы;
- определение степени правдоподобности модульной интегрально-вероятностной модели оценки показателей надежности программной системы в сравнении с реальной системой.

Основные выполняемые функции:

- определение показателей надежности проектируемой программной системы посредством модульной интегрально-вероятностной модели;
- определение показателей надежности проектируемой программной системы посредством оценки поведения ее симулятора;
- визуализация поведения компонентов симулятора программной системы во времени.

Множество функций комплекса образует структуру из пяти блоков:

- предварительная обработка данных – обеспечивает ввод и представление данных в удобной для пользователя форме;
- модульная интегрально-вероятностная модель – позволяет определять показатели надежности программной системы;
- симулятор – имитирует поведение проектируемой программной системы согласно с параметрами, полученными из блока предварительной обработки данных, в течение заданного числа тактов;
- мониторинг симулятора – предназначен для сбора статистики работы симулятора и определения показателей надежности программной системы по собранным данным;
- вывод – предназначен для вывода результатов работы системы (рис. 2).

Подсистема «Блок предварительной обработки данных» предназначена для решения следующих задач:

- редактирование данных;
- проверка корректности внесенных данных.

Статистические данные о составе программной системы вносятся в таблицу через буфер обмена или непосредственно пользователем.

Визуализация массивов количественных показателей программной системы и ее компонентов осуществляется в виде таблицы. При наличии ошибки в редактируемых данных пользователь будет оповещен о ней по средствам сообщения «Ошибка».

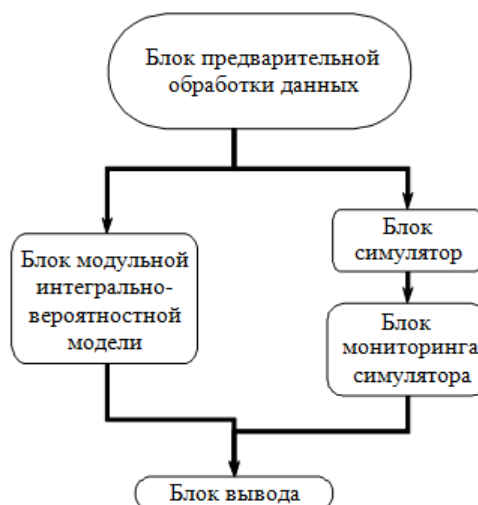


Рис. 2. Структура комплекса оценки надежности распределенных мультиверсионных программных систем

Подсистема «Блок модульной интегрально-вероятностной модели» является базовой в структуре комплекса и служит для определения показателей надежности проектируемой системы посредством использования модульной интегрально-вероятностной модели оценки надежности распределенных мультиверсионных программных систем.

Входными данными блока модульной интегрально-вероятностной модели служит результат работы подсистемы «Блок предварительной обработки данных».

Подсистема «Блок симулятора» базовая в структуре комплекса и служит для имитации работы проектируемой программной системы в соответствии с данными, полученными от подсистемы «Блок предварительной обработки данных». Имитация выполнения программной системы продолжается в течение заданного пользователем промежутка времени. В ходе работы данного блока предполагается, что каждый компонент программной системы запускается на исполнение с вероятностью  $PU_i$  на время, равное  $TU_i$ . При этом с вероятностью  $PF_i$  в процессе исполнения компонента произойдет ошибка, время которой равно сумме  $TA_i$  (время доступа к компоненту  $i$ ),  $TC_i$  (время анализа сбоя в компоненте  $i$ ) и  $TE_i$  (время устранения сбоя в компоненте  $i$ ). Также при определении наличия ошибки в компоненте принимается во внимание вероятность  $PL_{ij}$  (вероятность сбоя в компоненте  $i$  при сбое в компоненте  $j$ ).

Подсистема «Блок мониторинга симулятора» предназначена для сбора статистической информации о работе симулятора и определения на основе собранной статистики показателей надежности программной системы.

Подсистема «Блок вывода» предназначена для вывода результатов работы комплекса. Она отображает пользователю график работы симулятора и результаты работы блока мониторинга симулятора и блока модульной интегрально-вероятностной модели.

## **6. Исследование модульной интегрально-вероятностной модели оценки надежности распределенных мультиверсионных программных систем**

Исследование модульной интегрально-вероятностной модели оценки надежности программных систем включает рассмотрение поведения модели в зависимости от количества компонентов программной системы, от условной и безусловной вероятности возникновения сбоя в компонентах, а также зависимость показателей надежности программной системы от временных характеристик компонентов.

Положим, что программная система состоит из однотипных компонентов со следующими характеристиками: вероятность использования  $PU_i = 1$ , безусловная вероятность сбоя  $PF_i = 0,1$ , условная вероятность сбоя  $PL_{ij} = 0$  для всех  $j$ , время доступа  $TA_i = 5$  тактов, время анализа  $TC_i = 7$  тактов, устранения сбоя  $TE_i = 10$  тактов, среднее время использования компонента  $TU_i = 30$  тактов. Время имитации 1200 тактов.

Исследование зависимости надежности программной системы от числа компонентов выявило различный характер поведения показателей надежности модульной интегрально-вероятностной модели оценки надежности программной системы от числа компонент. Так, зависимость среднего времени простоя и вероятности безотказной работы программной системы от количества компонент  $F$  имеет линейную форму (рис. 3 и 4). В то же время зависимость роста среднего времени появления сбоя от числа компонент программной системы имеет нелинейную форму (рис. 5).



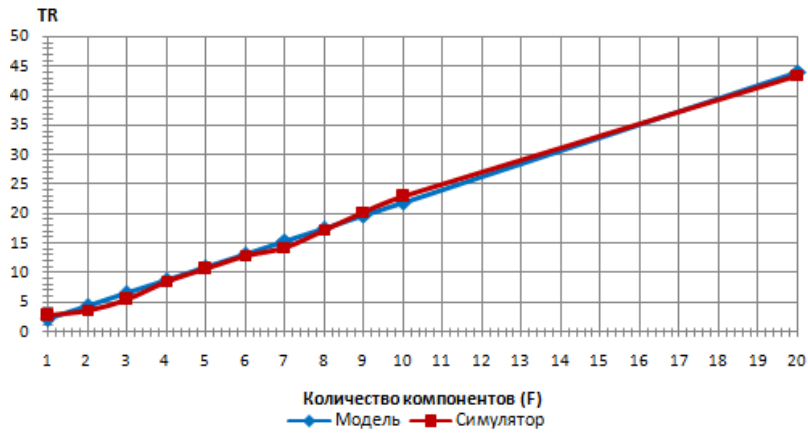


Рис. 3. Зависимость значения среднего времени простоя программной системы от числа компонентов

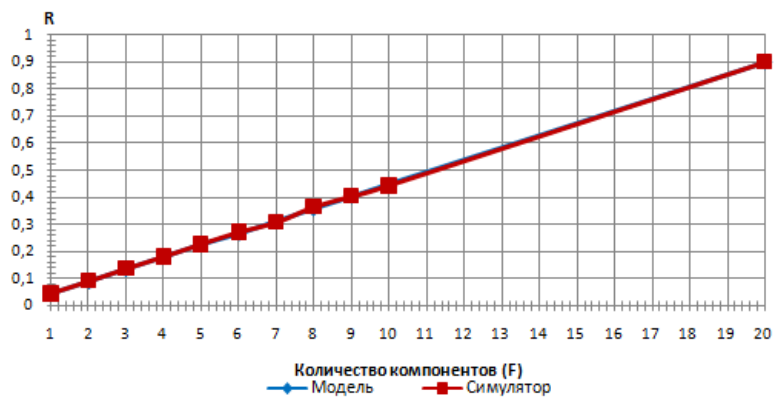


Рис. 4. Зависимость значения вероятности безотказной работы от числа компонентов программной системы

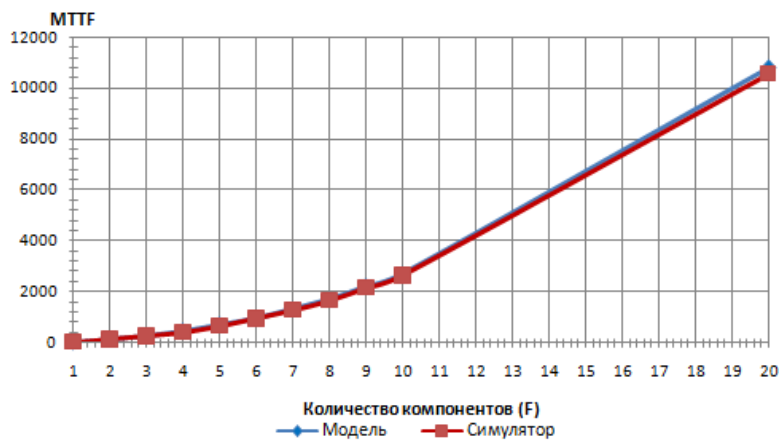


Рис. 5. Зависимость значения среднего времени появления сбоя от числа компонентов программной системы

При анализе зависимости показателей надежности программной системы с фиксированным числом компонентов  $F = 10$  от величины безусловной вероятности сбоя в ее компонентах выявлен линейный рост среднего времени простоя (рис. 6), линейное уменьшение среднего времени появления сбоя и вероятности безотказной работы программной системы (рис. 7 и 8).

При исследовании зависимости показателей надежности программной системы с фиксированным числом компонентов  $F = 10$  от величины условной вероятности сбоя в компоненте отмечено линейное уменьшение среднего времени появления сбоя программной системы при увеличении безусловной вероятности сбоя в компоненте (рис. 9).

При увеличении безусловной вероятности сбоя в компонентах программной системы происходит линейное увеличение среднего времени простоя системы (рис. 10). При этом отсутствует зависимость значения вероятности безотказной работы программной системы от величины условной вероятности сбоя в компоненте (рис. 11).

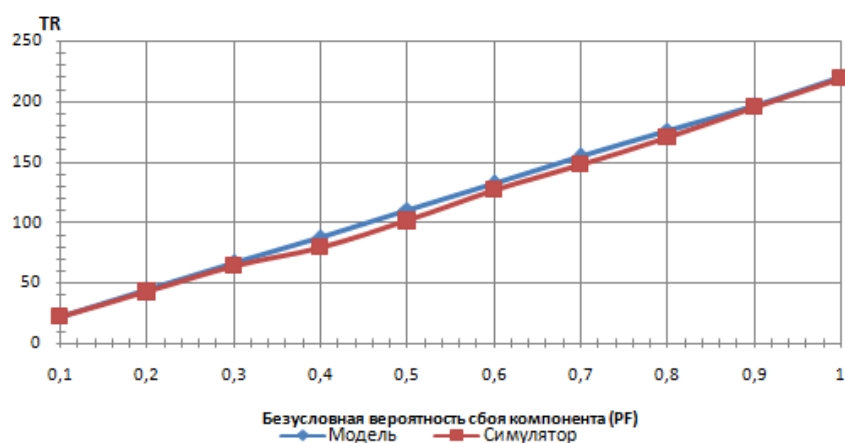


Рис. 6. Зависимость значения среднего времени простоя программной системы от величины безусловной вероятности сбоя в ее компонентах

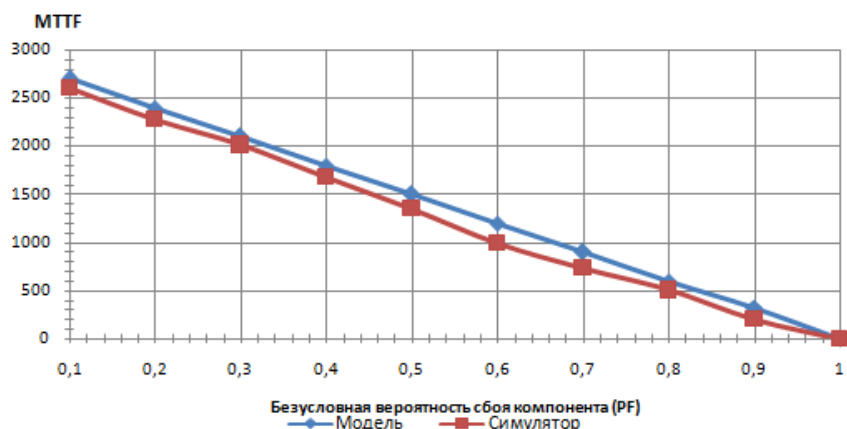


Рис. 7. Зависимость значения среднего времени появления сбоя от величины безусловной вероятности сбоя в компонентах программной системы

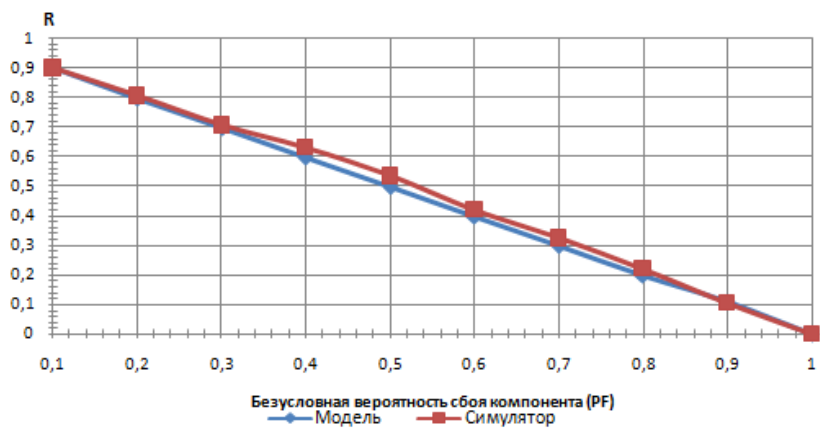


Рис. 8. Зависимость значения вероятности безотказной работы программной системы от величины безусловной вероятности сбоя в ее компонентах

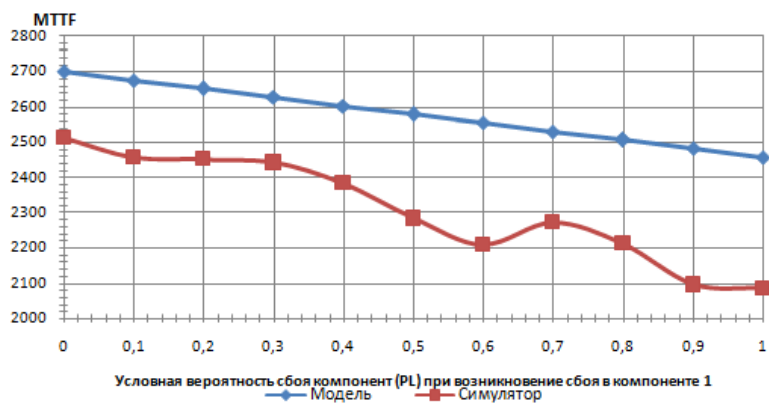


Рис. 9. Зависимость значения среднего времени появления сбоя от величины условной вероятности сбоя в компоненте при возникновении сбоя в компоненте 1

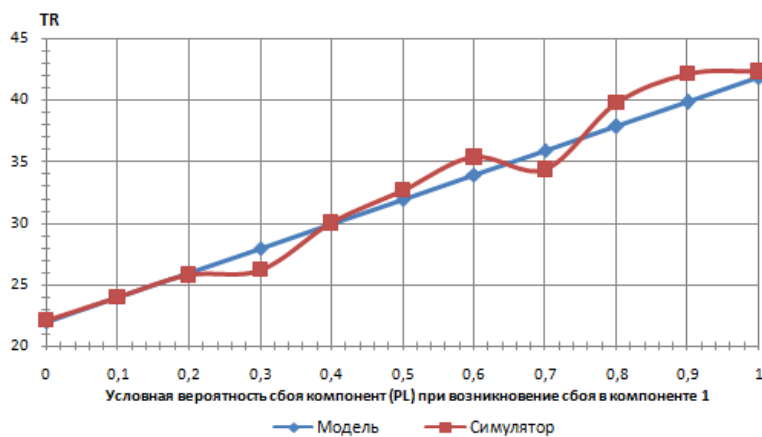


Рис. 10. Зависимость значения среднего времени простоя программной системы от величины условной вероятности сбоя в компоненте при возникновении сбоя в компоненте 1

Поскольку такой показатель надежности, как вероятность безотказной работы, не учитывает условную вероятность сбоя в компонентах, то для увеличения качества прогноза воспользуемся его модифицированной оценкой (10). Результат представлен на рис. 12.

На рис. 13 приведена зависимость среднего времени использования компонента от числа компонентов, входящих в состав программной системы, со средним временем между сбоями, равным 675 тактам.

На рис. 14 представлена зависимость среднего времени восстановления компонента от числа компонентов программной системы со средним временем восстановления системы, равным 11 тактам.

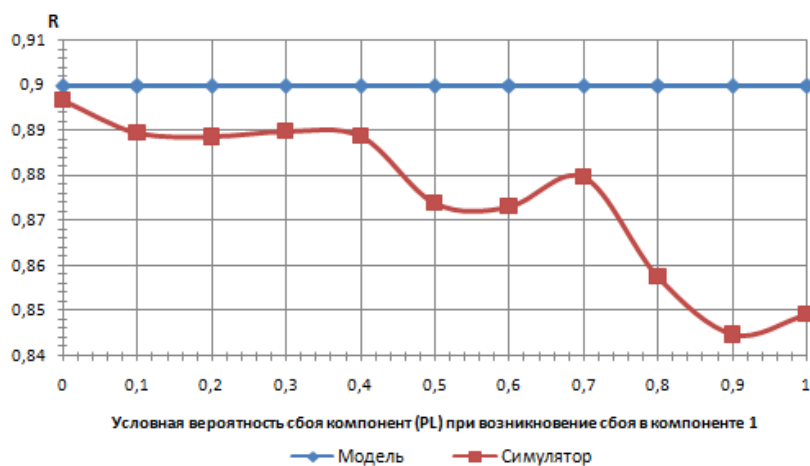


Рис. 11. Зависимость значения надежности программной системы от величины условной вероятности сбоя в компоненте при возникновении сбоя в компоненте 1

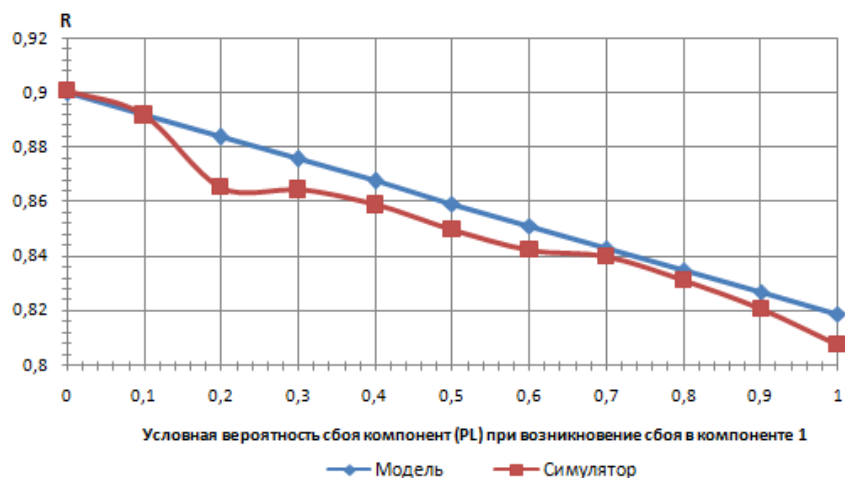


Рис. 12. Зависимость значения вероятности безотказной работы программной системы от величины условной вероятности сбоя в компоненте при возникновении сбоя в компоненте 1

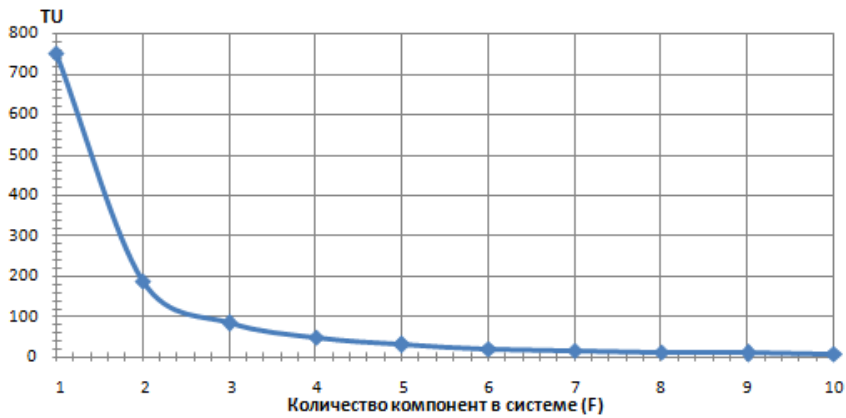


Рис. 13. Зависимость среднего времени использования компонента от числа компонент программной системы

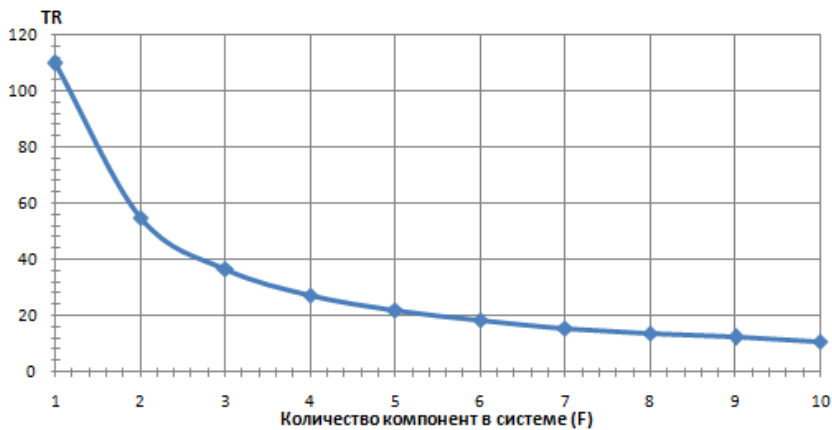


Рис. 14. Зависимость среднего времени восстановления после сбоя компонента от числа компонент программной системы

При исследовании выявлена обратная экспоненциальная зависимость среднего времени восстановления компонента от числа компонентов программной системы.

Исследование показало высокую точность прогноза значений показателей надежности модульной интегрально-вероятностной модели оценки надежности программной системы. Представленная модификация расчёта надежности для модульной интегрально-вероятностной модели позволяет расширить область применения модели и улучшить качество прогнозирования надежности программных систем.

## 7. Заключение

Представленная обобщенная модульная интегрально-вероятностная модель оценки надежности программных систем позволяет решать задачи оценки показателей надежности систем различной архитектуры: многоуровневой, мультиверсионной, распределенной, объектно-

ориентированной. Предложен алгоритм применения разработанной модели для оценки надежности программных систем с требуемой архитектурой. Подробно рассмотрена модификация обобщенной модульной интегрально-вероятностной модели для распределенных мультиверсионных программных систем.

Разработанный комплекс на базе представленной модификации обобщенной модульной интегрально-вероятностной модели для распределенных мультиверсионных программных систем обеспечивает решение следующих задач: предоставление пользователю информации о показателях надежности проектируемой программной системы и определение степени адекватности модульной интегрально-вероятностной модели показателей надежности по отношению к реальной программной системе.

Исследование подтвердило высокую эффективность модульной интегрально-вероятностной модели оценки надежности программных систем. Выявлена нелинейная зависимость между величинами среднего времени использования компонента, среднего времени восстановления после сбоя компонента и количеством компонентов программной системы, а также характер поведения модели при изменении величин условной и безусловной вероятности сбоя в компонентах программной системы. Экспериментально выявлено, что среднее время появления сбоя в программной системе и среднее время простоя линейно зависят от безусловной вероятности сбоя в компонентах системы.

#### Список литературы

- [1] *Dubrova E.* Fault tolerant design: an introduction. London: Kluwer Academic Publishers, 2005. 164 p.
- [2] *Reis G.A., August D.I., Mukherjee S.S., Cohn R.* Software fault detection using dynamic instrumentation. Proceedings of the Fourth Annual Boston Area Architecture Workshop, Boston, 2006.
- [3] *Лунаев В.В.* Тестирование компонентов и комплексов программ. М.: Синтег, 2010. 400 с.
- [4] *Царев Р.Ю., Штарик Е.Н., Штарик А.В., Завьялова О.И.* // Приборы и системы. Управление, контроль, диагностика. 2012. № 6. С. 29–33.
- [5] *Durairaj G., Koren I. and Krishna C.M.* // Simulation. 2001. Vol. 76. № 3. P. 172-183.
- [6] *Царев Р.Ю., Штарик А.В., Штарик Е.Н.* Мультиверсионное программное обеспечение. Алгоритмы голосования и оценка надежности: монография. Красноярск: Сибирский федеральный университет, 2013. 118 с.
- [7] *Avizienis A.* // Proceedings of the 10th IEEE Pacific Rim International Symposium on Dependable Computing, 2004.
- [8] *Boehm B.* Software Risk Management, IEEE Computer Society Press, 1989.
- [9] *Лунаев В.В.* Сопровождение и управление конфигурацией сложных программных средств. М.: Синтег, 2006. 372 с.
- [10] *Черкесов Г.Н.* Надежность аппаратно-программных комплексов. СПб.: Питер, 2005. 479.
- [11] *Ковалев И.В., Царев Р.Ю., Русаков М.А., Слободин М.Ю.* // Проблемы машиностроения и автоматизации. 2005. № 2. С. 30-35.

- [12] Царев Р.Ю., Штарик А.В., Штарик Е.Н. и др. // Вестник СибГАУ. 2012. № 4. С. 64-70.
- [13] Царев Р.Ю., Капулин Д.В., Штарик А.В., Штарик Е.Н. Синтез и управление развитием кластерных структур АСУ космических систем // Вестник СибГАУ. 2012. № 2. С. 80-84.
- [14] Torgersen, Mads. New features in C# 4.0. Microsoft. 2008.