

УДК 004.4

## НАПРАВЛЕНИЕ ПОВЫШЕНИЯ НАДЕЖНОСТИ СПЕЦИАЛЬНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ВСТРАИВАЕМЫХ СИСТЕМ

**Ширяев М.В.**, к.т.н., доцент, E-mail: shirjaev@inbox.ru

**Андреева О.Н.**, к.т.н., доцент, E-mail: andreeva\_olia@mail.ru

МГТУ МИРЭА, Концерн «Моринсис-Агат», Москва, Россия

**Аннотация.** В статье показана необходимость повышения надежности разрабатываемого и модернизируемого специального программного обеспечения встраиваемых систем, функционирующих в режимах реального времени. Предложен способ повышения надежности программного обеспечения путем реализации программной избыточности, более полно удовлетворяющий требованиям, предъявляемым системами реального времени. Представлен анализ влияния способов повышения структурной избыточности на временные показатели функционирования вычислительной системы.

**Ключевые слова:** надежность, программное обеспечение, структурная избыточность, реальное время.

## THE ENHANCEMENT OF RELIABILITY OF THE SPECIAL SOFTWARE FOR EMBEDDED SYSTEMS

**Shiryaev M.V.**, Ph.D., associate Professor, senior researcher, E-mail: shirjaev@inbox.ru

**Andreeva O.N.**, Ph.D., associate Professor, E-mail: andreeva\_olia@mail.ru

MSTU MIREA, Concern «Marines-Agat», Moscow, Russia

**Abstract.** The article shows the need to improve the reliability of the developed and modernized special software embedded systems operate in real time. Provides a method for improving the reliability of software by implementing software redundancy, more fully satisfying the requirements of real-time systems. Presents an analysis of ways to improve the impact of structural redundancy on the temporal parameters of the functioning of a computer system.

**Keywords:** reliability, software, structural redundancy, real-time.

### Введение

Анализ моделей оценки надежности программного обеспечения указывает на то, что единая система показателей оценки надежности специального программного обеспечения встраиваемых систем (СПОВС) отсутствует. Имеющиеся модели оценки надежности СПОВС не учитывают всего многообразия внутренних структур оцениваемых систем (аналитические модели), либо требуют больших временных и стоимостных затрат на проведение процесса сбора статистической информации (статистические, эмпирические модели).

Существующие модели оценки [2] надежности программного обеспечения можно условно разбить на эмпирические, статистические и аналитические (рисунок 1).

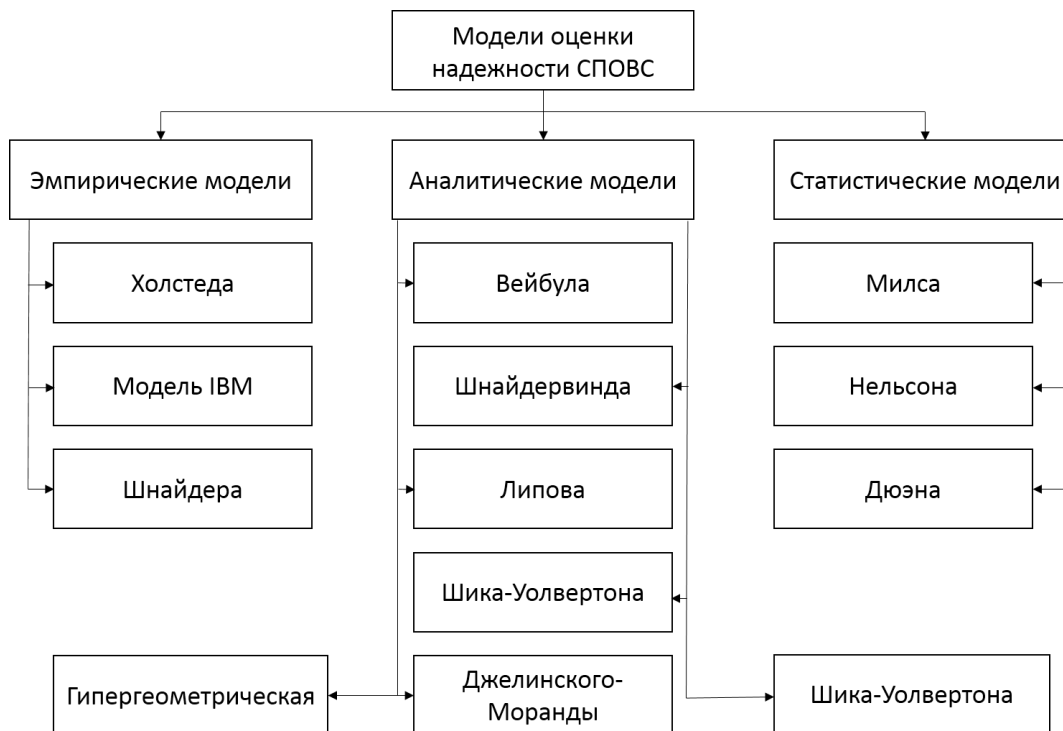


Рисунок 1 – Классификация моделей оценки надежности программного обеспечения.

### Повышения надежности специального программного обеспечения встраиваемых систем

Повышение надежности специального программного обеспечения встраиваемых систем возможно различными методами. Однако, использование различных методов осложнено спецификой функционирования вычислительных систем реального времени, а так же малой эффективностью от их применения. Сравнение преимуществ и недостатков различных методов повышения надежности представлены в таблице 1.

Методы повышения надежности СПОВС	Оперативный контроль состояния вычислительного процесса	Оперативное восстановление системы после сбоя	Потребность в ресурсах ВС на этапе эксплуатации	В настоящий момент используется
Методы контроля программ, данных и вычислительного процесса	+	-	+	+
Методы программного	+	+	-	-

восстановления				
Методы испытаний на надежность	-	-	+	+
Методы обеспечения надежности при сопровождении	-	-	+	+
Методы использования избыточности	+	+	-	-

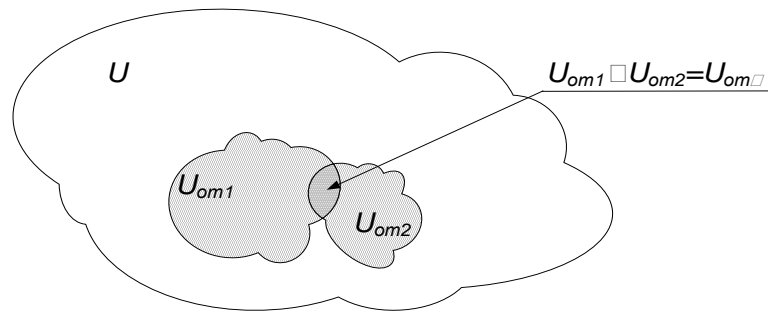
**Таблица 1. Сравнительные характеристики различных методов повышения надежности специального программного обеспечения встраиваемых систем**

Из данного анализа следует, что из методов повышения надежности программного обеспечения, таким качеством, как «Оперативное восстановление системы после отказа» [3] обладают только методы программного восстановления и методы использования избыточности.

Методы программного восстановления основаны на создании в процессе функционирования «контрольных точек», и в случае отказа система переводится в состояние, соответствующее последней работоспособной «точке». Это не всегда может привести к устранению отказа при дальнейшем функционировании системы. Отказ программного обеспечения может возникнуть при повторном поступлении набора входных данных, вызвавших отказ перед восстановлением системы.

Методы использования избыточности (программной, информационной) основаны на получении достоверного результата функционирования программного обеспечения за счет использования ресурсов вычислительной системы. Это и достоверность хранения и передачи информации (способ повышения надежности СПОВС путем добавления информационной избыточности) и достоверность получения результатов функционирования программных модулей (способ повышения надежности СПОВС путем добавления программной избыточности).

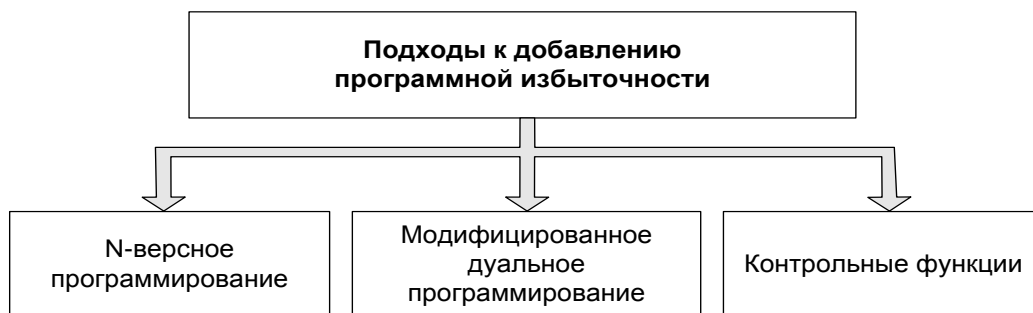
Способ повышения надежности СПОВС путем добавления программной избыточности основан на использовании различных по содержанию вариантов программных модулей, выполняющих одинаковые функции, но отличающиеся друг от друга алгоритмами функционирования, либо программной реализацией одного и того же алгоритма. Это приводит к смещению множества входных данных  $U_{om}$ , вызывающих отказ при функционировании программного модуля в общем множестве входных данных  $U$ . При одновременном использовании нескольких программных модулей, множество входных данных, вызывающих при функционировании отказ данной группы, можно представить пересечением всех множеств  $U_{omi}$ , где  $i$  – текущий модуль (рисунок 2.).



**Рисунок 2. Множество входных данных, вызывающих отказ при функционировании системы программных модулей**

Следовательно, если при использовании одного программного модуля был зафиксирован отказ, но входные данные, вызвавшие данную исключительную ситуацию, не входят во множество  $U_{om\Sigma}$ , использование другого программного модуля позволит это компенсировать.

Существует несколько подходов к добавлению программной избыточности (рисунок 3.).



**Рисунок 3. Подходы к добавлению программной избыточности**

Проведя обобщение всех существующих подходов к добавлению программной избыточности можно выделить ряд характеристик, позволяющих обобщить данный способ повышения надежности СПОВС. К данным характеристикам относятся:

- методика локализации последствий программных ошибок;
- количество дублирующих программных модулей  $N$ ;
- коэффициент избыточности СПОВС  $K_{изб}$ ;
- методика использования резервных программных модулей.

Все известные методики локализации последствий программных ошибок можно структурировать на следующие группы:

*проверка результатов функционирования программного модуля по мажоритарному принципу, используется при наличии нескольких вариантов одного программного модуля (нечетного количества);*

*проверка результатов работы двух программных модулей, с учетом их погрешности результатов.*

Если при одинаковых исходных данных результаты программ отличаются на величину большую, чем допустимая погрешность, делается вывод об отказе основной программы и принимается в качестве верного – результат резервной программы. В итоге средняя погрешность работы двух программ несколько увеличивается, но вероятность отказа уменьшается. Обозначим погрешность основной программы через  $\delta_1$ , а погрешность резервной – через  $\delta_2$ . Пусть вероятности их отказа соответственно  $p_1$  и  $p_2$ . При этом  $p_1 > p_2$ ,  $\delta_1 < \delta_2$ . Тогда можно показать, что средняя погрешность не отказавшей системы из двух программ есть  $\delta_c = (1 - p_1) \delta_1 + p_1 \delta_2$  при вероятности отказа системы  $p_c = p_2$ . В случае, когда имеется только основная программа, погрешность результата равна  $\delta_1$ , а вероятность отказа –  $p_1$ . Решающим органом выбора результата для системы из двух программ является простейшая программа, сравнивающая результаты  $y_1$  и  $y_2$  обеих программ и выдающая в качестве выходного результата величину:

$$y = \begin{cases} y_1, \text{ если } |y_1 - y_2| \leq \delta_1 + \delta_2; \\ y_2, \text{ если } |y_1 - y_2| > \delta_1 + \delta_2. \end{cases} \quad (1)$$

*использование для проверки результатов работы программного модуля взаимосвязанных функций, находящихся с проверяемой в контрольной взаимосвязи.* Этот способ дополнительно определяет значение другой функции, находящейся с основной, вычисляемой, в так называемых контрольных соотношениях. Эти соотношения позволяют не только обнаружить отказ одной из программ, но также восстановить искаженный результат отказавшей программы на основе результата безошибочно работающей программы. Простейшим примером данного способа контрольных соотношений является вычисление функций  $\sin(x)$  и  $\cos(x)$  по отдельным программам. В этом случае обнаружение и исправление ошибок ведется известными методами.

Данная характеристика является основной. Согласно этих методик, имеется несколько подходов к добавлению программной избыточности.

Использование данных методик в различных способах повышения программной избыточности напрямую влияет на основные показатели функционирования СПОВС:

вероятность безотказной работы  $p_{\delta o}$  и средний прирост времени реализации функции  $\Delta t_f$ .

Количество дублирующих программных модулей  $N$  указывает на то, сколько версий одного алгоритма используется для локализации последствий программной ошибки, либо, сколько программных модулей находится в контрольной взаимосвязи с проверяемым модулем.

Данный параметр напрямую воздействует на два ресурсных показателя функционирования СПОВС: объем занимаемого программой адресного пространства ОЗУ и ПЗУ ( $V_{ПЗУ}$ ,  $V_{ОЗУ}$ ) и средний прирост времени реализации функции  $\Delta t_f$ .

Коэффициент избыточности СПОВС представляет собой отношение количества зарезервированных программных модулей к общему количеству модулей, содержащихся в СПОВС:

$$K_{изб} = \frac{N_o}{N_k}, \quad (2)$$

где:  $N_o$  – количество задублированных программных модулей СПОВС;

$N_k$  – количество программных модулей СПОВС.

Численно данный показатель указывает долю программных модулей СПОВС, которые были дублированы. Этот параметр комплекса с количеством используемых дублирующих программных модулей также воздействует на объем занимаемого программой адресного пространства ОЗУ и ПЗУ ( $V_{ПЗУ}$ ,  $V_{ОЗУ}$ ).

Известны две методики использования резервных программных модулей [1]: замена основного программного модуля после фиксации сбоя в его работе;

использование результатов функционирования резервного программного модуля, без замены основного модуля при последующем вызове функции.

Иначе, данный параметр указывает, какой программный модуль будет использоваться после сбоя в качестве основного – основной или резервный (один из резервных).

При учете того, что источник входных данных не удовлетворяет условиям максимальной энтропии, можно говорить об условных вероятностях поступления пакетов входных данных, в связи с чем предпочтительнее может быть вторая методика.

Использование любого из имеющихся подходов к добавлению программной избыточности требует определенных ресурсных затрат, в первую очередь это затраты временного ресурса. Это связано с необходимостью при каждом использовании программного модуля запуска резервных с целью сравнения полученных результатов с

использованием специальных алгоритмов. Данные алгоритмы определяются используемой методикой локализации последствий программной ошибки.

Соответственно увеличивается объем используемого адресного пространства ПЗУ, за счет необходимости хранения резервных программных модулей.

Увеличение объема используемого ОЗУ происходит по следующим причинам: необходимость разграничения областей ОЗУ для основного и резервных модулей для устранения их взаимного влияния; необходимость хранения в ОЗУ результатов функционирования основного и резервных модулей для обнаружения факта отказа программного модуля.

### **Заключение**

Анализ моделей оценки качества программного обеспечения позволяет сделать вывод об отсутствии единой системы показателей оценки надежности СПОВС. Имеющиеся модели оценки надежности специального программного обеспечения не учитывают всего многообразия внутренних структур оцениваемых систем (аналитические модели), либо требуют больших временных и стоимостных затрат на проведение процесса сбора статистической информации (статистические, эмпирические модели). Таким образом, повышение надежности программного обеспечения путем добавления программной избыточности позволяет компенсировать отказы отдельных программных модулей за счет использования резервных. Однако большая часть времени, требуемого на реализацию СПОВС при использовании различных подходов к добавлению программной избыточности, затрачивается на обнаружение факта программной ошибки. Это связано с необходимостью при каждом использовании программного модуля проводить запуск резервных, с целью сравнения полученных результатов с использованием специальных алгоритмов. Данные алгоритмы определяются использованием той или иной методикой локализации последствий программной ошибки. Данная характеристика способа повышения надежности СПОВС путем добавления программной избыточности является основной, позволяющей выделить различные подходы к его использованию.

### **Список литературы**

1. Гецци К. и др. Основы инженерии программного обеспечения / Пер. с англ. СПб.: БХВ-Петербург, 2005.
2. Орлов С.А. Технология разработки программного обеспечения: Учеб. пособие. СПб.: Питер, 2003.

3. Ширяев М.В. Методика выбора программных модулей, рекомендованных для реализации программной избыточности. // Вычислительные системы реального времени и цифровые устройства. Научно-технический рецензируемый сборник / под ред. Е.С. Новикова – М.: Изд-во «Концерн «Моринформсистема-Агат», 2010. – С.110-116.

4. Кроль В.М., Красников А.К., Андреева О.Н. Методика построения зрительного интерфейса человеко-машинной системы управления объектом в режиме реального времени. //Наукоёмкие технологии. – 2013. №2. – С. 48-50.