УДК 681.3

МУЛЬТИВЕРСИОННАЯ МЕТОДОЛОГИЯ СОЗДАНИЯ ВЫСОКОНАДЕЖНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ *

Грибков Ю.А.
Научный руководитель – доцент Царев Р.Ю.
Сибирский федеральный университет

Обязательным компонентом большинства современных систем управления и обработки информации является программное обеспечение (ПО), которое выполняет основные функции обработки данных и выдачи управляющих воздействий. Широкое применение программного обеспечения объясняется тем, что процесс управления может вовлекать сложные расчеты с использованием большого объема информации. При этом необходимым результатом является выдача корректных результатов и обеспечение надежности функционирования ПО систем данного класса.

Надежность – свойство объекта или системы сохранять в установленных пределах значения все параметры, характеризующие способность выполнять требуемые функции в заданных режимах и условиях применения технического обслуживания и транспортирования.

На сегодняшний день единственной альтернативой методам тестирования и доказательства программ, обеспечивающей высокий уровень надежности исполнения программной системы, несмотря на ошибки отдельных программных компонент, является мультиверсионное программное обеспечение.

Для последующего изложения уточним некоторые понятия теории надежности и особенности их использования для изучения характеристик функционирования программ.

Отказ – событие, заключающееся в нарушении работоспособного состояния объекта или системы. Отказ при исполнении программ может проявиться как следствие нарушения кодов записи программ в памяти команд; стирания или искажения данных в оперативной или долговременной памяти; нарушения нормального хода вычислительного процесса.

Также в теории надежности существует понятие сбой, которое трактуется как самоустраняющийся отказ, не требующий внешнего вмешательства для замены отказавших компонент.

Понятие избыточности – это наличие в структуре устройства или системы возможностей сверх тех, которые могли бы обеспечить его нормальное функционирование. Избыточность вводится для повышения надежности работы и для исключения влияния на достоверность передаваемой информации помех и сбоев (в телекоммуникационных устройствах и программных системах). Существует три вида избыточности – временная, информационная и программная.

Временная избыточность – использование некоторой части производительности ЭВМ для контроля исполнения программ и восстановления вычислительного процесса.

Информационная избыточность состоит в дублировании накопленных исходных и промежуточных данных, обрабатываемых программным обеспечением.

Программная избыточность используется для контроля и обеспечения достоверности наиболее важных результатов обработки информации. Ее смысл заключается в

* Исследования выполнены в рамках реализации ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009 – 2013 годы

применении нескольких вариантов программ, различающихся методами решения некоторой задачи или программной реализации одного и того же метода.

Впоследствии, проводимые исследования в области данной методологии выявили три составляющих мультиверсионного программирования:

– процесс исходной спецификации и мультиверсионного программирования (NVP - N-version programming - N-версионное программирование);

– результат (NVS - N-version software - мультиверсионное программное обеспечение);

– внешние средства поддержки исполнения версий ПО (NVX - N-version executive – мультиверсионное исполнение).

Все три элемента NVP объединены целью независимой и конкурентной генерации функционально эквивалентных программных компонент (версий), которые будут исполнены, и будет принято решение о правильности функционирования ПО. Под независимой генерацией программ в данном контексте понимается организация проектирования программной системы так, чтобы каждая из N, вовлеченных в разработку отдельных программных модулей групп, не взаимодействовала с другой по отношению к процессу реализации программной системы.

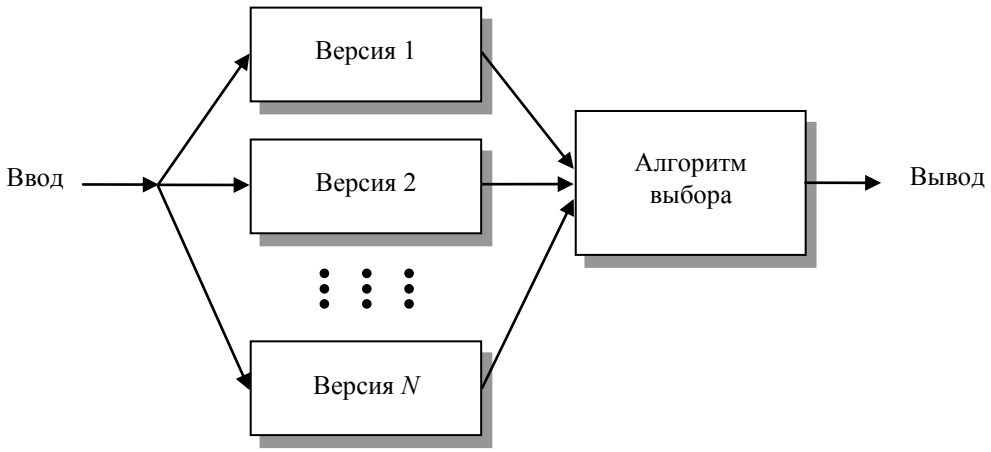

Рис. 1. Структура мультиверсионного или *N*-вариантного ПО

Существует четыре модели формирования мультиверсионного ПО:

– однофункциональное мультиверсионное ПО без избыточности;

– однофункциональное мультиверсионное ПО с избыточностью;

– многофункциональное мультиверсионное ПО без избыточности;

– многофункциональное мультиверсионное ПО с избыточностью.

Перед тем как перейти непосредственно к одной из математических моделей формирования мультиверсионного ПО, введем следующие условные обозначения:

n – число модулей мультиверсионного ПО;

m_i – число версий, i-го модуля, $i = 1, ..., n$;

R – вероятность безотказной работы мультиверсионного ПО;

R_i – вероятность безотказной работы i-го модуля;

R_{ij} – вероятность безотказной работы j-ой версии i-го модуля;

Z_{ij} – булева переменная, равная 1, если j-я версия выбрана для i-го модуля, иначе – 0;

C_{ij} – стоимость разработки и сопровождения j-й версии i-го модуля;

B – ограничение по стоимости создаваемой мультиверсионного ПО.

Рассмотрим однофункциональное мультиверсионное программное обеспечение с избыточностью, модель SR (Рис. 2).

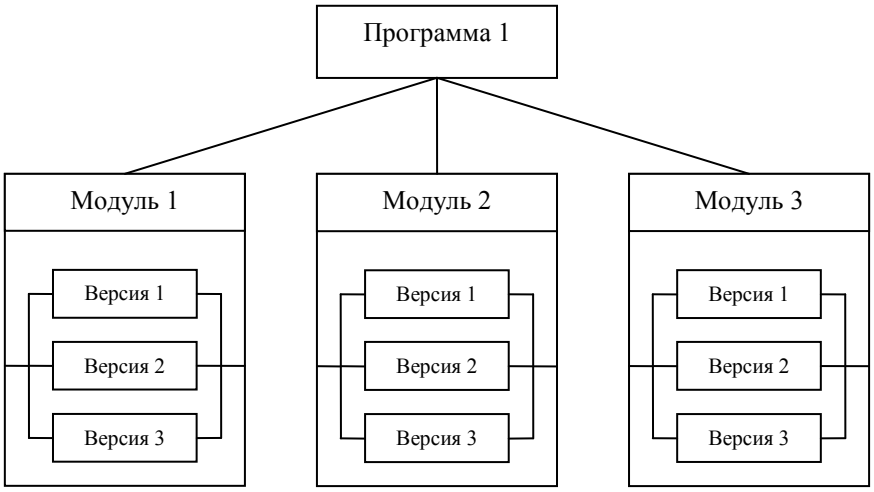

Рис. 2. Однофункциональное мультиверсионное ПО с избыточностью

Цель модели SR – определить оптимальный состав мультиверсионного ПО с учетом избыточности версий, максимизируя надежность мультиверсионного программного обеспечения, не выходя за рамки стоимостных ограничений:

$$\max R = \max_{Z_{ij}} \left\{ \prod_{i=1}^{n} \left(1 - \prod_{j=1}^{m_i} (1 - R_{ij})^{Z_{ij}} \right) \right\}$$

в соответствии с ограничениями

$$Z_{ij} = 0, 1; \quad i = 1, \ldots, n; \quad j = 1, \ldots, m_i,$$

где

$$\sum_{j=1}^{m_i} Z_{ij} \geq 2, \; i = \overline{1,n}, \tag{1}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{m_i} Z_{ij} C_{ij} \leq B.$$

Надежность i-го модуля увеличивается за счет того, что по крайней мере одна из m_i версий модуля выполняется правильно, а ограничение (1) гарантирует, что для каждого i-го модуля выбрано не менее двух версий.

Рассмотрим на примере увеличение надежности программного модуля за счет использования избыточных версий. Пусть модуль имеет три версии (рис. 3).

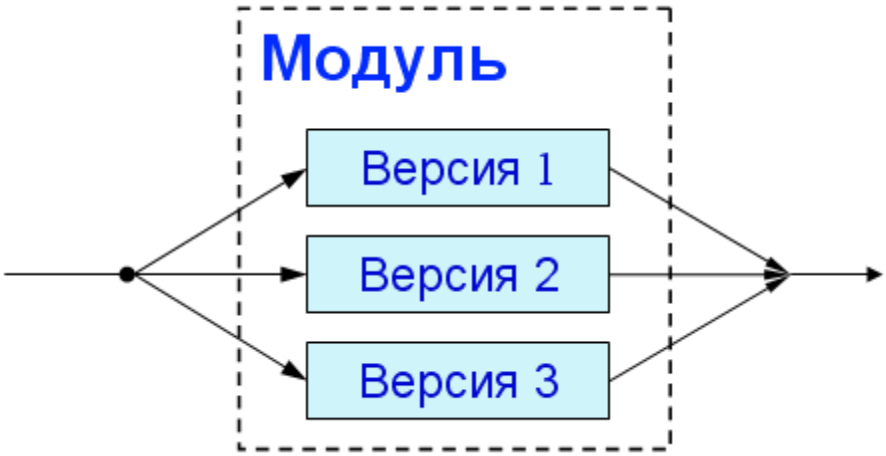

Рис. 3. Программный модуль с тремя версиями

Из табл. 1 видно, что все версии имеют низкие значения такого показателя надежности, как вероятность безотказной работы.

Таблица 1

Вероятность безотказной работы версий модуля

№ версии	Вероятность безотказной работы
1	0,98
2	0,96
3	0,97

Низкий уровень надежности отдельных версий неприемлем для полноценной эксплуатации модуля и, соответственно, всего ПО в целом. Однако благодаря применению мультиверсионного программирования, значение данного показателя существенно увеличивается. Согласно формуле, его надежность равна:

$$R_{\text{модуля}} = 1 - \prod_{i=1}^{3}(1 - R_i) = 0.999976.$$

Сравнительная диаграмма вероятности безотказной работы отдельных версий и самого мультиверсионного модуля представлена на рис. 4.

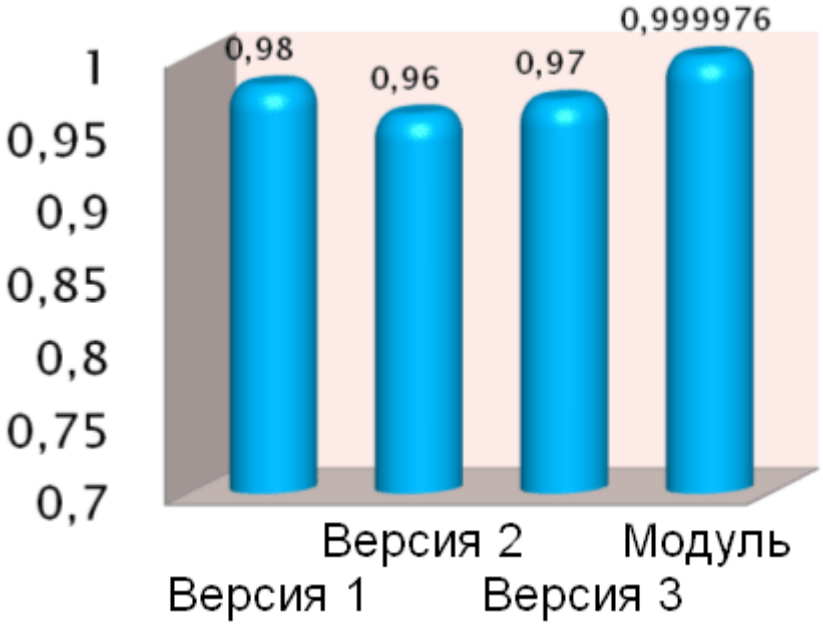

Рис. 4. Надежность версий и мультиверсионного модуля

Вышеприведенный пример показывает, как даже из ненадежных компонентов может быть сформировано высоконадежное программное обеспечение, соответствующее современным требованиям по обеспечению высокого качества программного продукта.

Данная методология гарантирует, что ошибки одной из версий программных модулей не приведут к нарушению процесса работы программного обеспечения, для которого характерны жесткие требования по надежности.