

Интернет-журнал «Наукovedение» ISSN 2223-5167 <http://naukovedenie.ru/>

Том 7, №5 (2015) <http://naukovedenie.ru/index.php?p=vol7-5>

URL статьи: <http://naukovedenie.ru/PDF/75TVN515.pdf>

DOI: 10.15862/75TVN515 (<http://dx.doi.org/10.15862/75TVN515>)

**УДК 004.052**

**Степович-Цветкова Галина Сергеевна**

ФГБОУ ВПО «Ивановский государственный университет»

Россия, Иваново<sup>1</sup>

Доцент кафедры «Прикладной математики и компьютерных наук»

Ученый секретарь

Кандидат экономических наук

E-mail: [sgs-2608@yandex.ru](mailto:sgs-2608@yandex.ru)

РИНЦ: [http://elibrary.ru/author\\_profile.asp?id=663596](http://elibrary.ru/author_profile.asp?id=663596)

## **Повышение надежности программного обеспечения: системно-функциональный подход**

---

<sup>1</sup> 153025, г. Иваново, ул. Ермака, д. 39

**Аннотация.** В данной статье рассматривается поэтапное применение функционального и системного подходов для выявления необходимых мер по повышению надежности программного обеспечения. В результате применения функционального подхода надежность программы складывается из совокупного достижения надежностей каждого функционального блока, обладающего своим особым набором возможных отказов и сбоев. При этом функциональным блоком является структурированная совокупность логически связанных команд, направленных на обеспечение выполнения строго определенного круга задач одного характера. Кроме того, предлагается методика оценки надежности программного обеспечения, основанная на выделении функциональных блоков. В отличие от существующих моделей оценки надежности программного обеспечения оценка производится не на этапе тестирования программного продукта, а оценивается конечный результат. Последующее применение системного подхода позволяет выявить особенности, вытекающие из взаимосвязи между составляющими элементами программ и определяющие новые способы управления надежностью программных продуктов. Предлагается адаптация основных системных принципов к вопросу повышения надежности программного обеспечения, обосновывается полезный эффект от их применения. Совместное применение функционального и системного подходов в вопросе выявления способов повышения надежности позволяет учесть, как внутренние особенности структурных элементов программы, так и взаимосвязи между ними.

**Ключевые слова:** надежность программного обеспечения; оценка надежности; обеспечение надежности; повышение надежности; программное обеспечение; функциональный подход; функциональный блок; системный подход; системные принципы; системно-функциональный подход.

**Ссылка для цитирования этой статьи:**

Степович-Цветкова Г.С. Повышение надежности программного обеспечения: системно-функциональный подход // Интернет-журнал «НАУКОВЕДЕНИЕ» Том 7, №5 (2015) <http://naukovedenie.ru/PDF/75TVN515.pdf> (доступ свободный). Загл. с экрана. Яз. рус., англ. DOI: 10.15862/75TVN515

Наблюдающаяся в наше время повсеместная компьютеризация приводит к тому, что компьютерные технологии проникают практически во все сферы деятельности человека, в том числе они все больше применяются в медицине, космонавтике и других областях, где даже небольшое отклонение в ходе работы программы может стоить человеческой жизни. Поэтому для принятия решения о возможности эксплуатации разработанной системы программное обеспечение должно быть достаточно надежным. При этом надежность является важнейшей характеристикой качества программного обеспечения (ГОСТ 28195-89 Оценка качества программных средств).

Теория надежности первоначально зародилась применительно к техническим наукам, и понятие надежности характеризовалось как свойство объекта выполнять заданные функции, сохраняя во времени значения установленных эксплуатационных показателей в заданных пределах. Кроме центрального понятия надежности теория технической надежности оперирует такими ключевыми понятиями как отказ, безотказность и сбой (ГОСТ 27002-89 Надежность в технике. Основные понятия. Термины и определения).

Названные характеристики могут быть в той или иной степени трансформированы применительно к теории надежности компьютерных программ, в которой понятие надежности выражается в наборе атрибутов, относящихся к способности программного обеспечения (ПО) сохранять свой уровень качества функционирования при установленных условиях за установленный период времени (ГОСТ Р ИСО/МЭК 9126-93 Информационная технология. Оценка программной продукции. Характеристики качества и руководства по их применению) [10-12].

Для обеспечения и оценки надежности программных продуктов существуют различные методы, соответствующие разработанным моделям надежности программного обеспечения, таким как модель Миллса, модель роста надежности, модель Шумана и другие. Ряд методов основаны на оценке числа оставшихся в программе ошибок по результатам ее тестирования. Однако, количество имеющихся ошибок не является прямым идентификатором надежности, поскольку к неверной работе программы может привести как одна, так и несколько ошибок, при этом они могут компенсировать друг друга, кроме того, некоторые ошибки влекут за собой неправильную работу программы при любом ее запуске, а другие только при определенном наборе исходных данных, который может быть трудно выявляемым, тем самым такие ошибки по-разному влияют на надежность всей программы [14].

Большинство разработанных методов предусмотрено, главным образом, для оценки надежности компьютерной программы в целом [6, 13, 16]. Однако, ошибки в программе, влекущие снижение ее надежности, как правило, локализованы в той или иной части, поэтому, на наш взгляд, целесообразно произвести выделение в программе отдельных функциональных блоков и оценку надежности осуществлять по блокам. Тогда надежность программы в целом будет определяться совокупной надежностью ее частей, которые назовем функциональными блоками. Под функциональным блоком будем понимать структурированную совокупность логически связанных команд, направленных на обеспечение выполнения строго определенного круга задач одного характера. Для многих программных продуктов характерно присутствие, например, таких функциональных блоков, как блок ввода исходных данных, блок вывода результирующей информации, блок объявления и определения пользовательских типов данных, блок обработки информации, подпрограммы, процедуры и функции и прочие.

Кроме того, большинство существующих моделей предполагают осуществление оценки надежности программного обеспечения на стадии тестирования программного

продукта [6-10], хотя, на наш взгляд, тестирование является неотъемлемой частью процесса разработки программного продукта, а в оценке надежности нуждается конечный результат.

Поэтому представляется, что надежность программы должна складываться из совокупного достижения надежностей каждого функционального блока. При этом каждый блок помимо общих для всех вероятных отказов, таких как синтаксические ошибки и опечатки, характеризуется своим особым набором возможных отказов и сбоев.

К примеру, блок ввода исходных данных снижает надежность программы в целом при отсутствии проверки введенных данных на корректность, то есть возможными отказами данного функционального блока являются выход данных из допустимого диапазона, а также неверный формат данных. Блок подключения внешних модулей и библиотек может характеризоваться следующими отказами: отсутствие в системе необходимой библиотеки и несовместимость подключаемых библиотек. Функциональный блок, содержащий обработку исходных данных, может содержать ряд неверно построенных логических решений реализуемого алгоритма, таких как неправильно заданные условия выхода из цикла, неправильная индексация цикла, неверно построенное логическое условие, приводящее к неправильному переходу при разветвлении алгоритма, и другие. Аналогично могут быть выделены вероятные отказы по всем функциональным блокам.

Стоит отметить, что обеспечение и оценка надежности – это самостоятельные категории, требующие отдельной проработки. Вопрос о том, что первично, дискуссионный, но, несомненно, и обеспечение, и оценка надежности программных средств взаимосвязаны и взаимозависимы, поскольку для обеспечения надежности необходимо оценить достигнутый уровень, а для того чтобы произвести оценку надежности, ее необходимо сначала обеспечить.

Обеспечение некоторого уровня надежности программ возможно в результате применения ряда мер, направленных на повышение надежности, наиболее общие из которых приведены в таблице (составлена автором).

**Таблица**

**Возможные отказы и меры повышения надежности по блокам**

<b>Функциональный блок ПО</b>	<b>Возможные отказы</b>	<b>Меры повышения надежности</b>
Подключение библиотек/модулей	– отсутствие в системе необходимой библиотеки; – несовместимость подключаемых библиотек; – синтаксические ошибки.	Загрузка необходимых внешних ресурсов в программное окружение при первом обращении. Использование отладочного компилятора.
Ввод исходных данных	– выход из допустимого диапазона данных; – неверный формат данных; – синтаксические ошибки.	Проверка исходных данных на допустимость (должны быть выделены все группы допустимых/недопустимых данных). Проверка формата данных, при необходимости преобразование данных к необходимому формату. Использование отладочного компилятора.

<p>Функциональные блоки, содержащие обработку исходных данных</p>	<p>– логические ошибки: отсутствует инициализация переменных; неверно заданы условия выхода из цикла; неверно индексирован цикл; неправильно задан переход по ветвям алгоритма; не учтена возможность появления отрицательных значений переменных, малых и больших величин и пр.;                  – синтаксические ошибки.</p>	<p>Построение логических цепочек и тестовых заданий для каждой ветви алгоритма; аккуратный и логичный стиль написания, как можно меньшее использование программистских трюков; анализ кода с помощью специальных анализаторов; пошаговая трассировка и пр.  Использование отладочного компилятора.</p>
<p>Вывод результата</p>	<p>– отсутствие необходимого устройства вывода;                  – неверный формат выводимых данных;                  – синтаксические ошибки.</p>	<p>Резервное копирование результата в создаваемый файл.                  Проверка формата данных, при необходимости преобразование данных к необходимому формату.</p>

Для оценки надежности программного обеспечения может быть использована следующая формула:  $N = \frac{1}{n} \sum_{i=1}^n N_i$ , где  $n$  – количество выделенных функциональных блоков

в программе,  $N_i$  – показатель надежности функционирования  $i$ -ого блока, а  $N$  – интегральный показатель надежности программного обеспечения. При этом оценка  $N_i$  должна производиться исходя из вероятностей возникновения каждого выделенного вида отказа в

блоке:  $N_i = \frac{1}{m_i} \sum_{j=1}^{m_i} p_j^i$ ,  $i = 1, \dots, n$ , где  $m_i$  – количество возможных отказов в  $i$ -ом блоке,  $p_j^i$  – вероятность возникновения  $j$ -ого вида отказа в  $i$ -ом блоке.

Отметим, что рассмотрение надежности программного продукта с точки зрения функционального подхода, тем не менее, не лишено ряда недостатков. Так, количество, и степень тяжести последствий от возникновения отказов и сбоев того или иного функционального блока могут быть разными. Но данное упущение можно устранить, если в общем случае функциональные блоки в рассматриваемой формуле при расчете интегрального показателя надежности будут иметь разные весовые характеристики.

Кроме того, локализация возникающих проблем программного обеспечения не всегда может производить положительный эффект, поскольку в таком случае не учитывается взаимодействие между функциональными блоками, от которого может зависеть результативность применения тех или иных мер повышения надежности программного обеспечения. К примеру, наличие глобальных и внешних переменных приводит к необходимости исследования изменений таких переменных во всей программе, а не только в выделенном блоке, так как переприсваивание значения переменной в одной части программы будет непосредственно влиять на возможность возникновения отказа в другой. Данный недостаток снимается путем применения системного подхода, для которого свойственно рассмотрение всех частей единого целого во взаимодействии и изучение взаимосвязей между составляющими элементами.

Программное обеспечение, созданное для достижения определенной цели, состоит из совокупности множества элементов и подсистем, что позволяет квалифицировать его как систему, рассмотрение внутренних и внешних связей которой возможно с использованием

основных принципов системного подхода. Под системой в широком смысле этого слова понимается набор элементов, составляющих единое целое, причем в совокупности целое наделено признаками, отсутствующими у отдельных его элементов [10, 15]. При данном подходе программная система представляет собой такую систему, которая является не просто сложением отдельных частей, а обладает дополнительными качествами, которые отсутствуют у составляющих ее элементов, но возникают в результате их взаимодействия. Действительно, компьютерная программа, состоящая из различных функциональных блоков и подпрограмм, лишь в целом способна выполнить поставленную задачу, при этом каждый составляющий ее как систему элемент, хотя и реализует определенные функции и даже, вероятно, может эксплуатироваться отдельно, является лишь частью единого целого и приносит свой особенный полезный эффект в работу всей системы.

Поэтому видится целесообразным рассмотрение вопроса повышения надежности программного обеспечения с точки зрения основных системных принципов, таких как принципа целостности, необходимого разнообразия, гибкости, внешнего дополнения и целеустремленности [1, 5, 7].

Согласно принципу целостности с ростом величины и сложности системы увеличивается вероятность того, что система не будет наследовать свойства своих составляющих элементов, а приобретет новые особенные свойства. Поэтому применение каких-либо способов повышения надежности программных продуктов только в одной функциональной части программы может не принести существенных результатов. Таким образом, различные способы повышения надежности необходимо использовать в комплексе, при совместном применении мер по управлению надежностью в различных взаимосвязанных частях программы может быть достигнут оптимальный результат. Так, например, выделение области памяти с некоторым резервом только в какой-либо одной подпрограмме может не дать реального эффекта, поскольку при необходимости использования данного резерва могут потребоваться соответствующие ресурсы в других частях программы. В то время как стратегия выделения резервной памяти, разработанная с системных позиций, может рационализировать ее использование.

Принцип необходимого разнообразия применительно к повышению надежности программных продуктов состоит в поиске таких средств и технологий программирования, которые обладали бы достаточными возможностями для реализации поставленных задач, представляя многообразие способов для их решения, и не принося чрезмерных издержек.

В соответствии с принципом гибкости программная система должна иметь возможность адаптироваться к каким-либо возникшим изменениям ее параметров, например, при изменении используемых программой системных библиотек может возникнуть сбой в программе, которого возможно избежать при заблаговременной загрузке необходимых внешних ресурсов в программное окружение.

Согласно принципу внешнего дополнения, системе требуется наличие так называемого «черного ящика» – определенных резервов, компенсирующих неучтенные воздействия внешней и внутренней среды [4]. Действительно, при разработке сложных алгоритмов даже самая тщательная проработка всех нюансов алгоритма не гарантирует абсолютно безотказную работу создаваемой программы, поскольку практически невозможно предвидеть все до единых возможных факторов, влияющие на работу программы в процессе ее эксплуатации. Неучтенные факторы могут резко снизить качество программного продукта, воздействие этих факторов возможно компенсировать, например, посредством обработки исключений.

Принцип целеобусловленности определяется наличием четко поставленной цели, достижение или недостижение которой позволяет контролировать правильность

функционирования созданного программного продукта. Детальное описание целевых функций существенно упрощает процесс построения оптимального алгоритмического решения и выбора технологий программирования, способствуя повышению надежности программного обеспечения.

Надежность компьютерных программ требует детального исследования, которому способствует применение в комплексе системного и функционального подходов, при этом функциональный подход позволяет определить надежность программы с учетом специфики каждого ее функционального блока, в свою очередь применение системного подхода способствует выявлению особенностей, возникающих из взаимодействия составляющих элементов программных систем. Поэтапное применение функционального и системного подходов позволяет определить новые возможные способы повышения надежности в конкретных условиях.

## ЛИТЕРАТУРА

1. O'Connor J., McDermott I. The Art of Systems Thinking: Essential Skills for Creativity and Problem Solving. Thorsons, 1997. 288 p.
2. Pan J. Software Reliability // Carnegie Mellon University. Dependable Embedded Systems. 1999. URL: [users.ece.cmu.edu/~koopman/des\\_s99/sw\\_reliability/](http://users.ece.cmu.edu/~koopman/des_s99/sw_reliability/).
3. Trachtenberg M. Discovering How to Ensure Software Reliability // RCA Engineer. Jan/Feb. 1982. Pp. 53-57.
4. Бир С.Т. Кибернетика и менеджмент. Перевод с англ. В.Я. Алтаева / Под ред. А.Б. Челюсткина. Изд. 2-е. М.: КомКнига, 2006. 280 с.
5. Блауберг И.В., Садовский В.Н., Юдин Э.Г. Системный подход в современной науке // Проблемы методологии системных исследований. М.: Мысль, 1970. С. 7-48.
6. Василенко Н.В., Макаров В.А. Модели оценки надежности программного обеспечения // Вестник Новгородского государственного университета. 2004. №28. С. 126-132.
7. Воскобойников А.Э. Системные исследования: базовые понятия, принципы и методология // Знание. Понимание. Умение. 2013. №6. URL: [zpu-journal.ru/e-zpu/2013/6/Voskoboinikov\\_Systems-Research/](http://zpu-journal.ru/e-zpu/2013/6/Voskoboinikov_Systems-Research/).
8. Ивутин А.Н., Суслин А.А. О некоторых применениях статистических распределений в оценке надежности программного обеспечения // Известия ТулГУ. Технические науки. 2011. №2. С. 568-575.
9. Карпов В.С., Ивутин А.Н., Суслин А.А. Подход к реализации методики оценки надежности ПО на основе комплексных метрик // Известия ТулГУ. Технические науки. 2009. №4. Ч.1. С. 116-125.
10. Липаев В.В. Качество программных средств. М.: Янус-К, 2002. 400 с.
11. Липаев В.В. Надёжность программных средств. М.: СИНТЕГ, 1998. 232 с.
12. Майерс Г. Надёжность программного обеспечения. М.: Мир, 1980. 360 с.
13. Павловская О.О. Статические методы оценки надежности программного обеспечения // Вестник Южно-Уральского государственного университета. Серия: Компьютерные технологии, управление, радиоэлектроника. 2009. №26 (159). С. 35-37.
14. Степович-Цветкова Г.С. Функциональный подход к надежности компьютерных программ // Перспективы развития науки и образования. – М.: «АР-Консалт», 2013. – С. 154-155.
15. Фатрелл Р.Т., Шафер Д.Ф., Шафер Л.И. Управление программными проектами. Достижение оптимального качества при минимуме затрат. М.: Вильямс, 2003. 1136 с.
16. Чернов А.В, Парашенко И.Г. Классификация моделей надежности программного обеспечения // Инженерный вестник Дона. 2012. №4. Ч.2. URL: [ivdon.ru/ru/magazine/archive/n4p2y2012/1319](http://ivdon.ru/ru/magazine/archive/n4p2y2012/1319).

**Рецензент:** Шварев Евгений Анатольевич, старший преподаватель кафедры «Государственного надзора и экспертизы пожаров» (в составе УНК «Государственный надзор»), капитан внутренней службы, кандидат технических наук, ФГБОУ ВО «Ивановская пожарно-спасательная академия ГПС МЧС России».

**Stepovitch-Tsvetkova Galina Sergeevna**

Ivanovo State University  
Russia, Ivanovo  
E-mail: [sgs-2608@yandex.ru](mailto:sgs-2608@yandex.ru)

## **Improving of the software reliability: the system-functional approach**

**Abstract.** In the paper it is discussed the gradual application of the functional and system approach to identify the necessary measures to improve the software reliability. As a result of the functional approach application software reliability consists of the aggregate achievements of the reliability of each functional unit, which has its own special set of possible faults and failures. functional units - structured sets of logically related commands to ensure the implementation of a well-defined range of the one character's tasks. And functional unit is a structured set of logically related commands to ensure the implementation of a well-defined range of the one character's tasks. In addition, the technique of evaluating the software reliability, based on the allocation of functional units. Unlike most of the existing models for assessment the software reliability assessment in this method does not occur in the testing phase of software product but the final result evaluated. Subsequent application of the system approach reveals the peculiarities arising from the relationship between the elements of the program and determine new ways to manage the software reliability. Adaptation of the basic system principles to improving the software reliability is offered, the author proves useful effect from their application. The combined application of the functional and system approach to identifying ways to improve the reliability allows to consider both internal features of the structural elements of the program, and the relationship between them.

**Keywords:** software reliability; reliability assessment; ensuring the reliability; improving of the software reliability; functional approach; functional unit; system approach; system principles; systemic-functional approach.

## REFERENCES

1. O'Connor J., McDermott I. The Art of Systems Thinking: Essential Skills for Creativity and Problem Solving. Thorsons, 1997. 288 p.
2. Pan J. Software Reliability // Carnegie Mellon University. Dependable Embedded Systems. 1999. URL: [users.ece.cmu.edu/~koopman/des\\_s99/sw\\_reliability/](http://users.ece.cmu.edu/~koopman/des_s99/sw_reliability/).
3. Trachtenberg M. Discovering How to Ensure Software Reliability // RCA Engineer. Jan/Feb. 1982. Pp. 53-57.
4. Bir S.T. Kibernetika i menedzhment. Perevod s angl. V.Ya. Altaeva / Pod red. A.B. Chelyustkina. Izd. 2-e. M.: KomKniga, 2006. 280 s.
5. Blauberg I.V., Sadovskiy V.N., Yudin E.G. Sistemnyy podkhod v sovremennoy nauke // Problemy metodologii sistemnykh issledovaniy. M.: Mysl', 1970. S. 7-48.
6. Vasilenko N.V., Makarov V.A. Modeli otsenki nadezhnosti programmnoy obespecheniya // Vestnik Novgorodskogo gosudarstvennogo universiteta. 2004. №28. S. 126-132.
7. Voskoboinikov A.E. Sistemnye issledovaniya: bazovye ponyatiya, printsipy i metodologiya // Znanie. Ponimanie. Umenie. 2013. №6. URL: [zpu-journal.ru/e-zpu/2013/6/Voskoboinikov\\_Systems-Research/](http://zpu-journal.ru/e-zpu/2013/6/Voskoboinikov_Systems-Research/).
8. Ivutin A.N., Suslin A.A. O nekotorykh primeneniyyakh statisticheskikh raspredeleniy v otsenke nadezhnosti programmnoy obespecheniya // Izvestiya TulGU. Tekhnicheskie nauki. 2011. №2. S. 568-575.
9. Karpov V.S., Ivutin A.N., Suslin A.A. Podkhod k realizatsii metodiki otsenki nadezhnosti PO na osnove kompleksnykh metrik // Izvestiya TulGU. Tekhnicheskie nauki. 2009. №4. Ch.1. S. 116-125.
10. Lipaev V.V. Kachestvo programmnykh sredstv. M.: Yanus-K, 2002. 400 s.
11. Lipaev V.V. Nadezhnost' programmnykh sredstv. M.: SINTEG, 1998. 232 s.
12. Mayers G. Nadezhnost' programmnoy obespecheniya. M.: Mir, 1980. 360 s.
13. Pavlovskaya O.O. Statische metody otsenki nadezhnosti programmnoy obespecheniya // Vestnik Yuzhno-Ural'skogo gosudarstvennogo universiteta. Seriya: Komp'yuternye tekhnologii, upravlenie, radioelektronika. 2009. №26 (159). S. 35-37.
14. Stepovich-Tsvetkova G.S. Funktsional'nyy podkhod k nadezhnosti komp'yuternykh programm // Perspektivy razvitiya nauki i obrazovaniya. – M.: «AR-Konsalt», 2013. – S. 154-155.
15. Fatrell R.T., Shafer D.F., Shafer L.I. Upravlenie programmnyimi proektami. Dostizhenie optimal'nogo kachestva pri minimume zatrat. M.: Vil'yams, 2003. 1136 s.
16. Chernov A.V., Parashchenko I.G. Klassifikatsiya modeley nadezhnosti programmnoy obespecheniya // Inzhenernyy vestnik Dona. 2012. №4. Ch.2. URL: [ivdon.ru/ru/magazine/archive/n4p2y2012/1319](http://ivdon.ru/ru/magazine/archive/n4p2y2012/1319).