Анализ методов оценки надежности оборудования и систем. Практика применения методов

[1]Н.В. Пакулин<npak@ispras.ru>
[2]Е.М. Лаврищева <lavr@ispras.ru>
[3]А.Г. Рыжов <ryzhov@ispras.ru>
[1,2,3]Институт системного программирования РАН,
109004, Россия, г. Москва, ул. А. Солженицына, д. 25.
[2]Московский физико-технический институт,
141700, Россия, Московская обл., г. Долгопрудный, Институтский пер., д. 9.

Аннотация. Проводится анализ моделей и методов оценки надежности технических и программных средств. Определяются основные понятия методов надежности и безопасности таких систем и ситуаций, приводящих к ошибкам, дефектам и отказам. Дано определение надежности и безопасности технических систем и программного обеспечения (ПО) систем. Приведена классификация моделей надежности: прогнозирующего, измерительного и оценочного типов. Описаны оценочные модели, которые применяются на практике. Определен стандарт жизненного цикла ПО (ISO 15288:2002), ориентированный на разработку и контроль компонентов систем на ошибки, начиная с требований к системе. Представлены результаты применения моделей надежности (Мусы, Гоэла-Окомото и др.) к малым, средним и большим проектам и дана сравнительная их оценка. Описан технологический модуль (ТМ) оценки надежности сложных комплексов программ ВПК (1989). Показана модель качества стандарта ISO 9126 (1-4):2002-2004 с показателями функциональность, надежность, эффективность и др., которые используются при определении зрелости и сертификата продукта.

Ключевые слова: надежность, ошибка, дефект, отказ, плотность дефектов, случайный процесс, безопасность, гарантоспособность, восстанавливаемость, отказоустойчивость, завершенность, оценка надежности, сертификат качества.

1. Введение

Рассматривается теория надежности и безопасности технических и программных систем [1-10]. Теория надежности техники возникла в рамках теории массового обслуживания систем и повлияла на становление и развитие теории надежности компьютерных систем и ПО систем.

Теоретики, изучая природу ошибок в системах, разработали более 100 математических моделей надежности, основанных на учете различных ситуаций, возникающих в технических и программных системах на первых и последующих поколениях ЭВМ. Методы надежности обеспечивают повышение надежности систем путем исправления разного рода обнаруженных ошибок в процессе разработки и их эксплуатации.

Надежность систем сформировалась как самостоятельная теоретическая и прикладная наука, способствующая определению качественных показателей систем (функциональность, точность, отказоустойчивость, гарантоспособность, завершенность и др.).

Методы оценки надежности систем позволяют прогнозировать, измерять и оценивать качество продукта с учетом возникающих ошибок, количестве и интенсивности отказов, а также процессов разработки отдельных компонентов систем в жизненном цикле (ЖЦ).

В работе рассматриваются все аспекты обеспечения надежности, безопасности и качества технических и программных систем [1-11].

2. Методы надежности оборудования и систем

Методы оценки надежности технических систем (аппаратуры, устройства, оборудование и др.) были разработаны значительно ранее компьютерных систем и основывались на вероятностных Марковских процессах с множеством числа состояний по теории массового обслуживания [12-16]. Эти методы обеспечивали проверку надежности функционирования техники, приборов и устройств в различных областях (машиностроение, энергетика, космос, медицина и др.). На их работоспособность влияли неисправности и разные недоработки в конструкции, приводящие к разрушительным последствиям и к ущербу системы в целом.

На оценку надежности компьютерных систем и ПО существенным образом влияют следующие особенности:

1. Большое количество кода в ПО, зачастую превышающее емкость физических элементов ЭВМ, и способов взаимодействия отдельных элементов ПО между собой;

2. Нематериальный характер элементов ПО, которые не деградируют, но стареют во времени и в их процессах, программах и конструкциях могут случаться разного рода непредвиденные ситуации;

3. Возникновение ошибочных ситуаций, разных дефектов и отказов, как в задании формальной спецификации отдельных элементов, так и в их выходном коде;

4. Элементы ПО трудно поддаются визуализации, обнаружению и коррекции найденных ошибок, поэтому измерение надежности ПО требует анализа и проверки данных об ошибках, чем для аппаратуры;

5. Системы ПО могут изменяться при функционировании и выходить из рабочего состояния от разных ситуаций внешней среды (вирусов, атак и др.), которые не предусмотрены в соответствующих моделях надёжности и обеспечиваются методами безопасности информации и систем.

Надежность технических систем зависит от двух факторов:

- качества отдельных технических конструктивных элементов системы;

- отсутствия дефектов и неисправностей в конструктивных элементах и их способность работать надежно и качественно.

Надежность программных систем зависит от этих же факторов и от случайных изменений данных и маршрутов исполнения программ, которые могут привести к неверным результатам или отказам, а также нарушить работоспособность, гарантированную разработчиками систем.

Между надежностью аппаратуры и ПО систем имеется сходство, состоящее в возникновении случайных явлений в процессах и системах, которые должны анализироваться методами теории вероятности, надежности и безопасности.

2.1. Определение термина надежности и безопасности систем

Под ***надежностью систем*** понимается способность системы сохранять свои свойства (безотказность, восстанавливаемость, защищенность и др.) на заданном уровне в течение фиксированного промежутка времени при определенных условиях эксплуатации.

Термин ***reliability*** (надежность) обозначает способность системы обладать свойствами, обеспечивающими качественное выполнение функций системы в соответствии с заданными требованиями.

Термин ***dependability*** означает гарантию способности (или гарантоспособность) технических систем [17-20]), состоящую в:

– безотказности (reliability) выполнения;

– готовности (availability) к работе;

– достоверности (high confidence) результатов;

– приспособленности к обслуживанию или ремонту (maintainability);

– информационной безопасности (security);

– конфиденциальности (confidentiality), секретности и целостности информации (integrity);

– безопасности (safety) работы системы без катастрофических последствий;

– эксплуатационной завершенности ПО (maintainability) и способности к восстановлению работоспособности системы.

При этом ***отказоустойчивость*** (fault-tolerance) обозначает способность системы автоматически за ограниченное время прогнозировать, предупреждать и восстанавливать функциональность системы после отказов с помощью механизмов поддержки всех составляющих ***гарантоспособности***.

Системы или процессы, которые обладают таким комплексным свойством, называют гарантоспособными. Им присущи традиционные надежностные свойства (безотказность, готовность, безопасность, целостность, конфиденциальность, восстанавливаемость и др.).

Вопросы разработки и использования гарантии качества систем обсуждаются более 25 лет на международных форумах и конференциях (Conference on Dependable Systems and Networks (DSN), European Dependable Computing Conference, (EDCC), International Conference on Computer Safety, Reliability and Security (SAFECOM), Probabilistic Safety Assessment and Management Conference (PSAM), Dependable Systems, Services and Technologies (DESSERT), Conference on Dependability of Complex Systems (DepCoS) и др.).

В 2004 году ассоциация IEEE издает журнал Dependable and Security Computing. В нем обсуждаются бизнес-критические приложения, электронная коммерция, банковские технологии и др. [16, 17].

С точки зрения гарантоспособности надежность является целевой функцией реализации системы. К ней предъявляются высокие требования (недопустимость ошибок, отказов, дефектов и других аварийных ситуаций). Надежность систем зависит от числа оставшихся и не устраненных ошибок в отдельных программах и компонентах системы.

Чем интенсивнее проводится эксплуатация системы, тем интенсивнее выявляются ошибки, быстрее растет надежность системы и соответственно ее качество. Надежность по существу является функция от ошибок, оставшихся в системе после ввода ее в эксплуатацию. Системы без ошибок считаются абсолютно надежными. Для оценки надежности систем используются собранные статистические данные - время безотказной работы, дефекты и частота (интенсивность) отказов.

Исследование надежности систем проводится с помощью методов теории вероятностей, математической статистики, теории массового обслуживания и математических методов надежности и безопасности. Главным источником информации для оценки надежности являются процессы тестирования, эксплуатации и испытания системы и данные, полученные при разработке систем в соответствии со стандартами ЖЦ (ISO/IEC 15846-1998, 15939:2002) системной инженерии [15-23].

2.2. Базовые понятия моделей надежности и безопасности

К базовым понятиям, которые используются в моделях надежности систем, относятся следующие [1-5].

Отказ (failure) – это переход системы из рабочего состояния в нерабочее.

Дефект (fault) – это последствие выполнения элемента программы, приводящее к некоторому непредвиденному событию (неверной интерпретации компьютером); невыявленные дефекты - источник потенциальных ошибок и отказов системы.

Ошибка (error) может быть следствием недостатка в спецификациях любой из программ или при принятии неверных действий в процессе испытания системы.

Интенсивность отказов – это частота появления отказов или дефектов в системе при ее тестировании, эксплуатации и сопровождении системы.

Процесс возникновения ошибок и отказов является, как правило, случайным и зависит от времени их возникновения или частоты их появления. В связи с этим модели надежности основываются на нахождении случайной величины в системе, числом и интенсивностью возникновения отказов в системе.

С точки зрения теории надежности в системе возникают случайные процессы во времени T (на последовательности времен $t_k < t_{k+1}$) и образуют случайную величину ξ в зависимости от значения t в моменты времени $t \in T$.

Если случайная величина дискретна, т.е. принимает конечное число значений $x_1, x_2, \ldots, x_n$, то ее распределение ξ описывается вероятностью $P(\xi = x_i)$ и в общем случае $F(x) = P(\xi < x_i)$ является функцией распределения случайной величины.

Случайный процесс с непрерывным временем, который описывается однородными событиями, называется пуассоновским процессом. Если функция оказывается случайной величиной, то требуется вычислить математическое ожидание или дисперсию, как среднее отклонение от ее реализации.

Поиск случайных величин осуществляется стохастическими методами и процесс соответственно является стохастическим, вероятностным.

Если на множестве времени T определяется случайный процесс, то для всех его точек t вычисляется случайная величина $\xi(t)$, которая и является ее значением.

Если случайные величины получаются между процессами требований и сопровождения распределены по показательному, эрланговскому или гиперэрланговскому законам, то поведение системы описывается Марковским процессом без непрерывных компонентов.

Одним из подходов к исследованию надежности на основе отказов систем является классическая теория вероятностей, согласно которой отказы в системе (в отличие от отказов технических средств) считаются случайными и зависят от дефектов, внесенных при разработке системы. Все модели оценки надежности базируются на статистике отказов и распределении интенсивности выявленных отказов в процессе верификации, тестирования и испытания системы для обеспечения ее работоспособности и гарантоспособности [16, 17].

Тестирование обеспечивает поиск дефектов и отказов, которые могут возникать случайно в системе и определяться с помощью функции $p\ (t, x, s)$,

где $t < s$ – момент времени и $x \in X$ - положение точки в системе координат.

Случайная функция удовлетворяет соотношению:

$P\ (t, x, u) = p\ (s, p\ (t, x, s), u)$,

где t < s < u означает, что в момент времени t в точке x система из состояния p (t, x, u), переходит в состояние $p\ (t, x, s)$.

Марковский процесс с дискретным временем и конечным числом состояний называется марковской цепью.

Функция $p_{ij}\ (t, x, s)$, при которой система в момент t переходит из i в j-состояние момента s, называется функцией Колмогорова [12] и определяется после решения системы уравнений:

$$\frac{d}{ds} p_{ij}(t, s) = \sum_{i=1}^{s} P_{ij}\ (t, s)\ a_{kj}(s)$$

Надежность по существу очень близка задачам безопасности. При разработке систем научными и некоммерческими институтами их трудно оценить на надежность и безопасность из-за того, что они делаются, как правило, не по стандартам. В то время как системы управления авиацией, атомной энергетикой и оборонной промышленностью разрабатываются по стандартам. В них надежность и безопасность определяют работоспособность системы в соответствии с требованиями и с минимум отказов и дефектов.

В характеристиках безопасности учитываются только те отказы, которые могут привести к катастрофическим последствиям и ущербам (например, пожар, взрыв, разрушение здания и др.). Оценка безопасности системы базируется на надежности функционирования ПО и БД. Оценка надежности зависит от метрик стандарта качества (внешние, внутренние, эксплуатационные). Они сравниваются с требованиями заказчика на систему и используются при сертификации продукта. Для оценки надежности и функциональной безопасности используются стандарты ISO/IEC 12207 для ПО и ISO 15288 -2006 систем.

На работоспособность системы влияют дефекты и ошибки проектирования, которые приводят к длительности восстановления и к необходимости преобразовывать программы от ошибочных отказов, сбоев средствами программной и информационной избыточности. Согласно стандарта ISO 9126 (1-4) определяются характеристики надежности с учетом обнаруженных дефектов и ошибок при функционировании гарантоспособного ПО систем.

Степень работоспособности/гарантоспособности зависит от установления соответствия характеристик требований, заданных в проекте и выявленных сбоев, отказов в ПО и возможными неисправностями в конструктивных элементах систем.

3. Классификация моделей надежности

Большинство моделей надежности исходят из предположения, что найденные ошибки и дефекты устраняются немедленно или определяются временем их устранения и новые дефекты не вносятся. В результате количество дефектов в системе уменьшается, а надежность возрастает, такие модели получили

название моделей роста надежности. Shick G. [6, 16, 17] предложил следующую классификацию моделей надежности.

Прогнозирующие модели надежности основаны на измерении технических характеристик создаваемой программы: длина, сложность, число циклов и степень их вложенности, количество ошибок на страницу операторов программы и др.

Модель Мотли–Брукса основывается на длине и сложности структуры программы (количество ветвей и циклов, вложенность циклов), количестве и типах переменных, а также интерфейсов.

Модель Холстеда дает прогнозирование количества ошибок в программе в зависимости от ее объема и таких данных, как число операций (n_1) и операндов (n_2), а также их общее число (N_1, N_2).

Измерительные модели предназначены для измерения надежности ПО, работающего с заданной внешней средой и следующими ограничениями:

– ПО не модифицируется во время периода измерений свойств надежности;

– обнаруженные ошибки не исправляются;

– измерение надежности проводится для зафиксированной конфигурации ПО.

Примером таких моделей является модель Нельсона, Рамамурти–Бастани и др.[3].

Модель Нельсона основывается на выполнении k-прогонов программы при тестировании и позволяет определить надежность по формуле:

$$R(k) = \exp[-\sum \nabla t_j \lambda(t)],$$

где t_j – время выполнения j-прогона, $\lambda(t) = -[\ln(1-q_i)\nabla j]$ и при $q_i \le 1$ интерпретируется как функция интенсивности отказов.

Оценочные модели основываются на серии тестовых прогонов и проводятся на этапах тестирования системы. В тестовой среде определяется вероятность отказа программы при ее тестировании или выполнении. Эти типы моделей могут применяться на этапах ЖЦ и могут быть следующих видов [7-11].

Модели без подсчета ошибок основаны на измерении интервала времени между отказами и позволяют спрогнозировать количество ошибок, оставшихся в программе. К этим моделям относятся модели Джелински и Моранды, Шика Вулвертона, и Литвуда–Верала.

Модели с подсчетом отказов базируются на количестве ошибок, обнаруженных на заданных интервалах времени. К этому классу моделей относятся модели Шика–Вулвертона, Шумана, Пуассоновская модель и др.

Модели с подсевом ошибок основаны на количестве устраненных ошибок и подсеве, внесенном в программу искусственных ошибок, тип и количество которых заранее известны. При внесении изменений в программу проводится повторное тестирование и оценка надежности. Этот подход базируется на тестировании и редко используется из-за дополнительного объема работ для покрытия тестами компоненты системы.

Модели с выбором области входных значений основываются на генерации множества тестовых выборок из входного распределения. К этому типу моделей относится модель Нельсона и др. На процессах выявления отказов, их интенсивности используются еще такие группы:

1) модели, рассматривающие интенсивность отказов, как Марковский и пуассоновский процесс;

2) модели роста надежности.

Четкой границы между этими моделями провести нельзя, однако по фактору распределения интенсивности отказов и их поведения можно разделить на экспоненциальные, логарифмические, геометрические, байесовские и др.

Для практической оценки надежности более всего представляет интерес оценочная модель Мусы, Мусы-Окомото и др. Рассмотрим их.

1. Оценочная модель Мусы [8] основана на следующих положениях:

- тексты адекватно представляют среду функционирования;
- происходящие отказы учитываются (оценивается их количество);
- интервалы между отказами независимы;
- время между отказами распределено по экспоненциальному закону;
- интенсивность отказов пропорциональна числу ошибок;
- скорость исправления ошибок (относительно времени функционирования) пропорциональна интенсивности их появления.

Эта модель учитывает интервалы между отказами, которые распределяется по экспоненциальному закону, а интенсивность отказов пропорциональна числу обнаруженных ошибок.

Исходя из этой модели, можно установить зависимость:

1) среднего числа отказов от времени функционирования τ (рис.1), которое задается в виде:

$$m=M_0 [1 - exp (\frac{c\tau}{M_0 T_0})],$$

где M_0 – общее число ошибок; T_0 – начальная наработка на отказ; c – коэффициент времени испытаний; τ – время функционирования.

2) средней наработки на отказ T от времени функционирования τ (рис.2):

$T = T_0 \exp (\frac{c\tau}{M_0 T_0})$, где M_0, T_0, c – зависят от наработки на отказ.

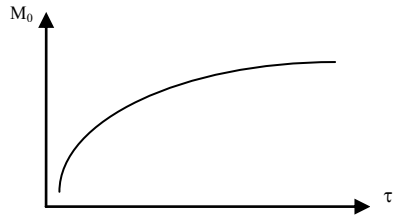


Рис. 1.

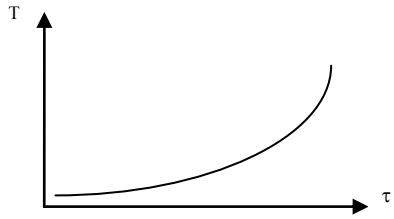


Рис 2.

График этой зависимости представлен областью 1 (рис. 3), для которой M_1 = 1, 2, ... – номера наблюдений, а $\tau_1, \tau_2 \ldots \tau_{M1}$ – время между отказами. Область 2 (рис.3) соответствует достижению средней наработки T_p на отказ за время $\Delta\tau$.

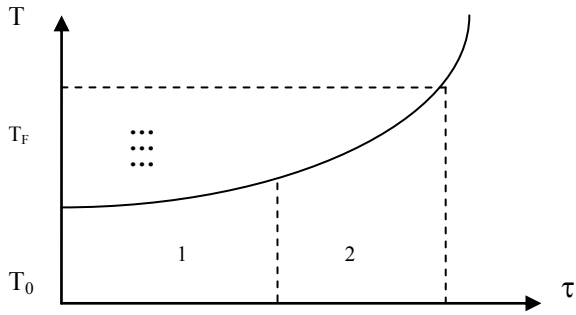


Рис. 3.

По собранным данным об ошибках оцениваются параметры T_0 и M_0, с помощью которых определяются дополнительное число ошибок по формуле:

$$\Delta m = M_0 T_0 \left[\frac{1}{T} - \frac{1}{T_0}\right].$$

2. Модель Муса–Окумото (логарифмическая) допускает, что некоторые дефекты имеют большую вероятность проявления в виде отказов, снижают интенсивность отказов с каждым устраненным дефектом и дают экспоненциальное распределение. Функция $m(t)$ зависит от времени проявления отказов и имеет вид:

$m(t) = ln(loqt + 1)$,

где q – задает экспоненциальный спад интенсивности отказов с каждым устраненным дефектом, а функция интенсивности отказов $\lambda(t)$ имеет вид:

$\lambda(t) = \lambda_0 / \lambda_0 \theta t + 1$.

3. Модель Гоело–Окумото (экспоненциального роста) обеспечивает процесс обнаружения ошибок с помощью неоднородного пуассоновского процесса. В ней интенсивность отказов зависит от времени, а количество выявленных ошибок при тестировании трактуется как случайная величина.

Исходные данные m, X_i и T аналогичны данным предыдущих моделей. Функция среднего числа отказов, обнаруженных к моменту t, имеет вид:

$m(t) = N(1 - e^{-bt})$,

где b – интенсивность обнаружения отказов; $q(t)=b$ - показатель роста надежности.

Функция интенсивности $\lambda(t)$ зависит от времени работы системы до отказа:

$\lambda(t) = Nbe^{-bt}$, $t \geq 0$,

где N и b решаются с помощью уравнения:

$-m/N - 1 + \exp(-bT) = 0$

Процесс оценки надежности включает:

– протоколирование отказов в ходе функционирования системы, измерение надежности по отказам и использование результатов измерений для определения потерь надежности в период времени эксплуатации;

– анализ частоты и серьезности отказов при определении порядка устранения соответствующих ошибок;

– оценка влияния времени функционирования системы на надежность с помощью инструментов разработки системы для измерения надежности.

3.1. Оценка надежности систем реального времени

Некоторые типы систем реального времени с обеспечением безопасности требуют высокой надежности (недопустимость ошибок, точность, достоверность и др.), которая в значительной степени зависит от количества оставшихся и не устраненных ошибок в процессе ее разработки на этапах ЖЦ.

В ходе эксплуатации ошибки также могут обнаруживаться и устраняться. Если при их исправлении не вносятся новые ошибки или их меньше, чем устраняется, то в ходе эксплуатации надежность системы непрерывно растет. Чем интенсивнее проводится эксплуатация, тем интенсивнее выявляются ошибки и быстрее растет надежность.

На надежность ПО влияют, с одной стороны, угрозы, приводящие к неблагоприятным последствиям, риск нарушения безопасности системы, с другой стороны, способность совокупности компонентов системы сохранять устойчивость в процессе ее эксплуатации. Риск уменьшает свойства надежности, особенно если обнаруженные ошибки могут быть результатом проявления угрозы извне [16].

Методы и модели надежности постоянно развиваются и уточняются, поскольку надежность является одной из ключевых проблем измерения качества современных распределенных по Интернет систем.

Появилось новое направление – *инженерия надежности ПО* (Software reliability engineering – SRE), которое ориентировано на количественное изучение операционного поведения компонентов системы по отношению к пользователю, ожидающему получить надежную работу системы [16-20] путем:

1) измерения надежности, т.е. проведения количественной ее оценки методами предсказаний, собранными данными о поведении системы в процессе тестирования и эксплуатации системы;

2) оценки стратегии и применения метрик для готовых компонентов, созданных в процессе разработки компонентов системы в заданной среде и стандартов на измерение надежности системы;

3) современных методов инспектирования, верификации, валидации и тестирования в процессе разработки отдельных компонентов и системы в целом.

3.2. Обеспечение надежности на этапах ЖЦ

Для получения высокой надежности системы требуется наблюдать за достижением показателей надежности и качества на всех этапах ЖЦ согласно рекомендациям стандарта ISO/IEC 12207: ЖЦ [16]. К основным процессам стандарта ЖЦ относятся:

– спецификация требований,

– проектирование,

– реализация,

– тестирование,

– испытание,

– сопровождение.

На этапе спецификации требований определяются задачи и внешние спецификации основных (целевых) требований к системе с заданием метрик для оценки надежности, в терминах интенсивности отказов или вероятности безотказного его функционирования. Разработчики системы формируют:

– приоритеты функций системы по критерию важности их реализации;

– параметры среды и интенсивности использования функций и их отказов;

– входные и выходные данные для каждой функции модели;

– категорий отказов и их интенсивности при выполнении функций в единицу календарного времени.

На этапе проектирования определяются:

– размеры информационной и алгоритмической сложности всех типов проектируемых компонентов;

– категории дефектов, свойственные всем типам компонентов системы;

– стратегии функционального тестирования компонентов по принципу «черного ящика» с помощью тестов для выявления дефектов в классе категорий данных.

Для достижения надежного продукта проводится анализ:

– вариантов архитектуры системы на соответствие требованиям к надежности;

– анализ рисков, режимов отказов, деревьев ошибок для критических компонентов с целью обеспечения отказоустойчивости и восстанавливаемости системы;

– прогнозирование показателей размера системы, чувствительности к ошибкам, степени тестируемости, оценки риска и сложности системы.

На этапе реализации и тестирования системы проектные спецификации переводятся в коды и подготавливаются наборы тестов для автономного и комплексного их тестирования. При проведении автономного тестирования обеспечение надежности состоит в предупреждении появления дефектов в компонентах и создание эффективных методов защиты от них. Все последующие этапы разработки не могут обеспечить надежность систем, а лишь способствуют повышению уровня надежности за счет обнаружения ошибок с помощью тестов различных категорий.

На этапе испытаний проводится системное тестирование для соответствия внешних спецификаций функций целям проекта. Испытание проводится в реальной среде функционирования или на испытательном стенде для имитации функций компонентов. При подготовке к испытаниям изучается "история" тестирования на процесса ЖЦ в целях использования ранее разработанных тестов, а также составления специальных тестов испытаний. На этом этапе осуществляется:

– управление ростом надежности путем неоднократного исправления и регрессионного тестирования ПС;

– принятие решения о степени готовности ПС и возможности его передачи в эксплуатацию;

– оценка надежности по результатам системного тестирования и испытаний по соответствующим *моделям надежности*, подходящих для заданных целей.

На этапе сопровождения оценка надежности ПС проводится:

– протоколирование отказов в ходе работы системы, измерения надежности функционирования и использования результатов измерений для определения потерь надежности в период времени эксплуатации;

– анализ частоты и серьезности отказов для определения порядка их устранения;

– оценка влияния функционирования системы на надежность в условиях усовершенствования технологии и применения новых инструментов разработки.

3.3. Применение моделей для оценки надежности ПО

Практика применения моделей показывает, что среди названных моделей наиболее перспективными являются модели оценочного типа, которые базируются на пуассоновских процессах (модели Мусы, Гоэла–Окомото, S-образные и др.). По этим моделям надежность стремиться к 1. Одним из недостатков является форма кривой интенсивности выявленных отказов или неисправностей (экспоненциальная) и строго спускается при t>0. Это говорит о том, что при тестировании проведено недостаточно экспериментов или мало найдено ошибок, когда интенсивность отказов была близка 0. В системе остаются ошибки и их поиск требует больше времени.

Что касается S-образной модели, функция интенсивности $\lambda(t)$ выявления ошибок в зависимости от времени работы имеет вид:

$\lambda(t) = a\beta^2 t \exp(-\beta t)$,

где a – общее количество дефектов, обнаруженных от начала и до конца тестирования;

β – скорость изменения функции интенсивности выявления отказов.

Введение в формулу параметра в степени 1 модели Мусы и Гоэла–Окомото дает изменение формы кривой так, что она сначала растет, а потом спадает. Практика применения этих моделей в автоматизированных системах привела к уточнению функции интенсивности при введении дополнительного параметра n:

$\lambda(t) = a\beta^{n+1} t^n \exp(-\beta t)$,

где n отражает сложность и размер проекта некоторой системы. Это позволяет более точно определить форму кривой интенсивности с учетом получаемых практических результатов.

В таблице 1 представлены практические значения функций интенсивности отказов $\lambda(t)$ и количество отказов $\mu(t)$ для базовых и общих моделей. В них значения a и β находятся в следующих соотношениях:

$N = a,\ \beta = a,\ b = \beta,\ \beta^1 = \beta,\ a_0 = a\beta$.

Таблица 1. Характеристика моделей надежности Пуассоновского типа

Название модели	Функции интенсивности отказов $\lambda(t)$	Функции кумулятивного количества отказов $\mu(t)$
Модель Гоэла-Окумото	$\lambda(t) = Nb \exp(-bt)$	$\mu(t) = N(1 - \exp(-bt))$
Модель Мусы	$\lambda(t) = \beta_0 \beta_1 \exp(-\beta_1 t)$	$\mu(t) = \beta_0 (1 - \exp(-\beta_1 t))$
S –подобная модель	$\lambda(t) = a\beta^2 t \exp(-\beta t)$	$\lambda(t) = a\{1-(1+\beta t) \exp(-\beta t)$
Модель Шнайдевинда	$\lambda(t) = a_0 \exp(-\beta t)$	$\mu(t) = a_0/\beta (1 - \exp(-\beta t))$

Название модели	Функции интенсивности отказов $\lambda(t)$	Функции кумулятивного количества отказов $\mu(t)$
Общая модель пуассоновского процесса	$\lambda(t) = a\beta^{n+1} t^n \exp(-\beta t)$	$\mu(t) = a\,(n^1 - \sum n\beta^{n-1}/(n-1)^1 t^n \exp(-\beta t)))$

Для метода максимального правдоподобия задаются данные *a, β, n,* решим систему уравнений:

$$\begin{cases} \alpha = \dfrac{m}{1 - \sum\limits_{i=0}^{n} \dfrac{n!\beta^{n-i} t_m^{n-i}}{(n-i)!} \exp(-\beta t_m)} \\ \dfrac{n+1}{\beta} m = \sum\limits_{k=1}^{m} t_k + \dfrac{m\beta^n t_m^{n+1} \exp(-\beta t_m)}{1 - \sum\limits_{i=0}^{n} \dfrac{n!\beta^{n-i} t_m^{n-i}}{(n-i)!} \exp(-\beta t_m)} \end{cases}$$

где параметр n зависит от процесса тестирования и его рекомендуемых значений:

n=0 – для небольшого проекта, в котором разработчик является также тестером (модель Мусы, Гоэло-Окомото и др.);

n=1 – для среднего проекта, в котором тестирование и проектирование ПО исполняются несколькими разработчиками из одной рабочей группы (S-образная модель);

n=2 – для большого проекта, в котором группы тестирования и проектирования работают параллельно;

n=3 – для очень большого проекта, в котором группы тестирования и разработки работают независимо друг от друга.

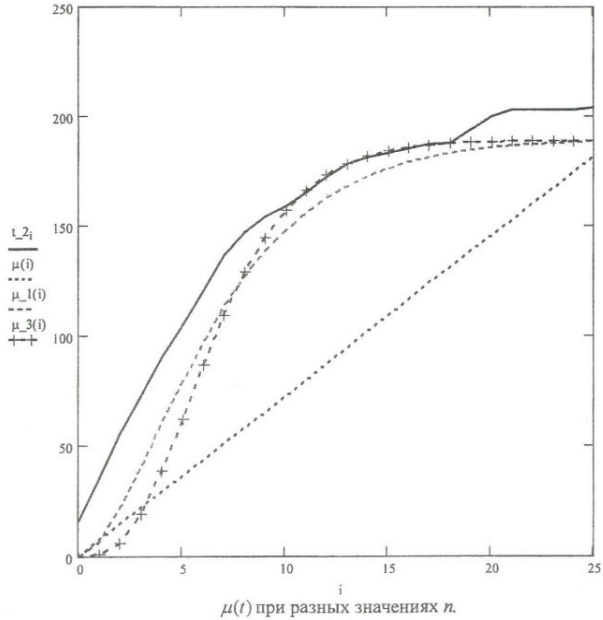
На основе экспериментальных данных получены функции о количестве отказов $\mu(t)$ и интенсивности отказов $\lambda(t)$ на выходных данных и значениях параметра n (рис. 4). Этот рисунок показывает вид функций $\mu(t)$ при разных значений n=0,1, 2, 3.

Наибольшее приближение достигается при n=3, а наименьшее при n=0 (модель Мусы, Гоэло-Окомото и др.). Это подтверждается соответствующими статистическими данными (табл.2), которые задают разницу между выходными данными (t_2) и соответствующими значениями функции $\mu(t)$ при значениях n = 0, 1, 2, 3.

На основе экспериментальных данных *a, β, n,* (табл. 2) приведены значения функций $\mu(t)$ и $\lambda(t)$ при n = 3, 2, 1, полученные при использовании методов оценки надежности Мусы, Мусы-Окомото и Шнайдевинда. Функции $\mu(t)$ для этих методов приведены на графике (рис. 4). Им соответствуют кривые экспоненциального типа. Графики этих функций близки друг другу из-за близких значений, полученных по заданным моделям.

Таблица 2. Статистические данные к разнице μ(t) при n=3, 2, 1 и данных t_2

Статистические показатели	**Разница функций *t_2 - μ_3***	**Разница функций *t_2 - μ_2***	**Разница функций *t_2 - μ_1***	**Разница функций *t_2 - μ***
Среднее отклонение	16.13522	16.22889	19.88387	58.93807
Медианное отклонение	15.27700	14.11600	16.0000	60.89700
Максимум отклонение	33.58100	54.23600	49.10800	88.80200
Минимум отклонение	4. 848000	-1.280000	4.175000	15.96200
Среднеквадрати-ческое отклонение	8.374089	17.37143	14.07056	23.63765



μ(t) при разных значениях n.

Рис. 4.

Для более эффективного применения приведеннях моделей надежности требуется значительное количество статистических данных о количестве и распределении отказов. А это требует увеличения большего количества экспериментов на процесах тестирования, системного тестирования для покрытия тестами всех компонентов и маршрутов их прохождения.

3.4. Технологический модуль (ТМ) оценки надежности систем

ТМ разработан в рамках работ по проекту ПРОТВА ВПК (1986-1989). В состав ТМ надежности входит четыре программных модуля (ПТМ) [15, 16, 22 с.283-296]: распределение надежности, прогнозирование плотности дефектов, прогнозирование надежности и оценки надежности.

1. ПТМ «Распределения надежности» реализует метод распределения надежности по компонентам системы путем парного их сравнения и построения квадратной матрицы *A* размером $n \times n$ из элементов вида:

$$a_{11} = a_{22} = ... = a_{nn} = 1,\ a_{ij} = \frac{1}{a_{ji}}\ ,\ u, j = 1,...,n;\ i \neq j;\ n=k,\ l,\ m,$$

где n – количество сравниваемых компонентов, *k, l, m* – количество функций и модулей соответственно. Матрица включает относительный вес i-го компонента и вычисляется по формулам:

$$w_i = \frac{\sum_{j=1}^{n} a_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}},\ \sum_{i=1}^{n} w_i = 1$$

В случае больших размеров матрицы в целях получения более точных относительных оценок компонентов иерархии, вычисляются, так называемые, собственный вектор и собственные значения матрицы согласно известным уравнениям [24]. В них используются следующие данные: λ_{max} – максимальное собственное значение матрицы *A* *n*–порядка, w_i – коэффициент относительного веса элементов матрицы *A*, W = (w_1, w_2, ... w_n) – собственный вектор, которому соответствует λ_{max}. Общность решения задачи сравнения устанавливается соотношением $\alpha = \sum_{i=1}^{n} w_i$ и значением $\sum_{i=1}^{n} w_i = 1$. Если матрица *A* имеет *n-1* собственных значений λ, равных нулю и $\lambda_{max} = n$, то она является согласованной.

Определение индекса согласованности *CI* и коэффициента согласованности *CR* проводится по формулам:

$$CI = \frac{\lambda_{\max} - n}{n-1},\ CR = \frac{CI}{E(CI)},\ \text{где}$$

E(CI) – математическое ожидание для матрицы парных сравнений *A* ($n \times n$).

Критерий приемлемости парного сравнения элементов в матрицах размером $n \geq 3$ получен такой: $CR \leq 0.05$ и *CR < 0.1* для *n>5*. По результатам сравнения формируется квадратная матрица *F(k × k)*.

Аналогично проводится сравнение приложений ПС. В результате сравнения получают k матриц. Возможный порядок каждой матрицы – l, а максимальный порядок каждой из них – m.

Инструмент для поддержки метода сравнения – ExpertChoice для входной матрицы A автоматически получает собственный вектор W, собственное значение λ_{max} и коэффициент согласованности CR. Для вычисления λ_{max} и W используются соответствующие функции пакета Matlab [15].

Результаты сравнений заносятся в форму, содержащую перечень весовых коэффициентов программ, критерии, индексы и коэффициенты согласованности. Они предоставляются в виде готовых результатов обработки матриц. Полученные весовые коэффициенты синтезируются с помощью пакета MATLAB 6.5. Результаты отображаются в виде отчета о распределении надежности по объектам системы.

2. **ПТМ «Прогнозирование плотности дефектов»** реализует набор моделей надежности для заданного класса программ системы обработки данных.

Прогнозирование плотности дефектов проводится по модели RLM (Rome Laboratory Model) и состоит в оценке влияния на плотность дефектов согласно следующих действий:

1. Анализ значений параметров модели прогнозирования надежности, включая остаток дефектов от предыдущего этапа работ с ПО, используется для целевого распределенного значения надежности ПО.

2. Сравнение прогнозируемого значения надежности с целевым распределенным значением.

3. Корректировки переменных параметров для учета текущего состояния проекта ПО.

4. Оценка параметров модели прогнозирования надежности.

5. Прогнозирование плотности дефектов.

6. Определение пороговых значений (допусков) для оценок результатов прогнозирования и анализа альтернатив.

7. Расчет прогнозного значения надежности для ПО.

Полученная оценка является модификатором базового значения плотности дефектов для определенного класса ПО.

Расчет плотности дефектов делается по модели RLM (Rome Laboratory Model). Сначала выполняется однократное прогнозирование плотности дефектов по формуле:

$$D_0 = \prod_{i=1}^{9} K_i ,$$

где K_i – модификаторы плотности дефектов D_0, с учетом пороговых значений данных о плотности дефектов.

Затем для каждого ПО результаты сравниваются с полученными по модели RLM. Проверка показала, что для ПО объемом 10 - 25 KSLOC погрешность прогнозировании плотности дефектов – примерно составляет 30-35%. Это

объясняется некоторыми ограничениями системы Hugin Lite 6.5. Полученные результаты по определению плотности дефектов используются при прогнозировании надежности ПО.

3. ПТМ «Прогнозирование надежности» реализует метод прогнозирования значения надежности по каждому модулю системы по следующей модели надежности:

$$R_i = \exp[-D_i I_i \cdot (1 - \exp(\frac{\rho_i \cdot K}{I_i \cdot \varphi_i} \cdot t))],$$

где ρ_i – параметр среды эксплуатации i-го модуля, φ_i – характеристика среды ее разработки, I_i – оцененный размер начального кода, а D_i – прогнозируемая плотность дефектов в системе. Коэффициент дефектов K – константа, предвиденная для всех объектов ПС, а значения ρ_i и φ_i – известны на момент первоначального прогнозирования надежности, они не изменяются во время разработки компонентов системы.

4. ПТМ «Оценка надежности системы» согласно классификации дефектов (Orthogonal Defect Classification), в соответствии с которой для каждого выявленного дефекта определяются параметры: тип дефекта, триггер дефекта, влияние дефекта. Эти параметры используется одной или двумя подходящими моделями надежности из выше приведенного в целях проведения оценки прогнозного значения надежности отдельных модулей и системы в целом. Результаты оценки сравниваются, и выбирается из них наиболее правдоподобная модель.

По стандарту ISO/IEC 9126 (1-4) определяются показатели качества (табл. 3).

Таблица 3. Характеристики качества в стандарте ISO/IEC 9126

№	Наименование характеристики	Определение характеризуемых свойств ПС
q1	Функциональность (functionality)	Свойства ПП, обусловливающие способность выполнения функций в соответствии требованиям в процессе тестирования и испытания системы в заданной среде
q2	Надежность (reliability)	Свойства ПП, обусловливающие ее способность сохранять уровень функционирования и низкую вероятность отказов в процессе выполнения
q3	Применимость (usability)	Свойства ПП, обусловливающие ее способность быть понимаемой и удобной для использования в указанных условиях
q4	Эффективность (efficiency)	Свойства ПП для рационального использования выделенных ресурсов при работе системы в установленных условиях

q5	Сопровождаемость (maintainability)	Свойства ПП, которые обеспечивают модификацию, усовершенствование или адаптацию системы к изменениям среды, требований и функциональности.
q6	Переносимость (portability)	Свойства ПП, обусловливающие ее способность быть перенесенным из одной среды в другую.

На основе полученных данных о надежности и других показателях качества (функциональность, эффективность и др.) рассчитывается целевое значение завершенности и полезности системы (ПС), адекватных потребностям заказчика. При этом мера эксплуатационного качества системы определяется функцией полезности вида: $Q_{nc} = \sum_{i=1}^{k} a_i \cdot R_i$

где a_i – мера важности i-й функции системы для процесса, R_i – надежность выполнения функций в заданном периоде t эксплуатации системы.

Данные по всем показателям качества (q-quality) q_1-q_6 в табл. 3 оцениваются по формуле:

$$q_1 = \sum_{j=1}^{6} a_{1j} m_{1j} w_{1j}$$

где a_i - атрибуты каждого показателя качества (i=1-6); m_{ij} – метрики q_i показателя с j-атрибутами качества; w_{ij} - вес i -показателя качества системы с j-атрибутами. Полученные данные по показателям (характеристикам) модели качества и R_i – надежность выполнения функций входят в сертификат качества [15].

4. Заключение

В работе рассмотрены подходы к оценке надежности технических и программных систем с применением моделей надежности из множества существующих моделей разных видов и типов. Определены основные базовые понятия надежности, обеспечивающие оценку надежности по соответствующим моделям надежности ПС, основанным на времени функционирования и/или количестве отказов (ошибок), получаемых в компонентах в процессах ЖЦ тестирования, системного тестирования и эксплуатации системы. Согласно приведенной классификация моделей надежности процессы обнаружения ошибок в программах носят случайный Марковский и пуассоновский характер и обеспечивают поиск ошибок, дефектов и отказов. Некоторые модели надежности позволяют прогнозировать число ошибок в процессе тестирования, другие оценивать надежность с помощью функций надежности по данным, собранным на этапах ЖЦ разработки системы и испытания. Для примера приведены экспериментальные данные для оценки интенсивности отказов $\lambda\ (t)$ и количества отказов $\mu(t)$ с помощью базовых (Мусы, Гоэла-Окомото и др.) и

общей модели надежности, собранных данных на этапах ЖЦ и приведены сравнительные оценки результатов оценки. Дано описание инструментального комплекса модулей ПТМ, обеспечивающих распределение надежности, прогнозирование плотности дефектов и оценки надежности. Приведены показатели качества в стандартной модели ISO 9126 (1-4) и оценки качества, включая измерение показателя надежности, а также других показателей качества, которые входят в сертификат готового продукта.

Список литературы

[1]. Липаев В.В. Надежность программного обеспечения. – М.: СИНТЕГ, 1998.–231с.
[2]. Липаев В.В. Методы обеспечения качества крупномасштабных программных систем. – М.: СИНТЕГ, 2003.–510 с.
[3]. Майерс Г. Надежность программного обеспечения,– М.: Мир, 1980.–360с.
[4]. Мороз Г.Б., Лаврищева Е.М. Модели роста надежности программного обеспечения.– Киев: Препринт 92–38, 1992.– 23с.
[5]. Липаев В.В.. Надежность и функциональная безопасность комплексов программ реального времени.- Москва, ЗАО «Светлица», 2013.-193 с.
[6]. Shick G.J., Wolverton R.W. An analysis of computing software reliability models /IEEE Tras. Software Eng. – V. SE–4. – № 2. – 1978. – P. 104–120.
[7]. Shanthikumar J.G. Software reliability models: A Review // Microelectron. Reliab. – 1983. –V. 23. –№ 5 – P. 903–943.
[8]. Goel Amrit L., “Software reliability models: Assumptions, limitations, and applicability. //IEEE Transactions on Software Engineering, Vol. SE–11, № 12. – 1985. –P. 1411–1423.
[9]. Musa J.D. Okumoto K. A. Logarithmic Poisson Time Model for Software Reliability Measurement //Proc. 7- International Conference on Software Engineering. – Orlando, Florida. – 1984. – P. 230–238.
[10]. Yamada S., Ohba M., Osaki S. S–shaped software reliability grows modeling for software error detection // IEEE Trans. Reliability. – 1983. – R–32. – № 5. – P. 475–478.
[11]. Chulani S. Constructive quality modeling for defect density prediction: COQUALMO // International Symposium on Software Reliability Engineering (ISSRE'99), Boca Raton, N. 1–4, 1999.
[12]. Гнеденко Б.В., Коваленко И.Н. Введение в теорию массового обслуживания.- Наука, М.: 1966.
[13]. Гнеденко Б.В., Шпак В.Д. Вероятностные характеристик сложных систем. - Известия АН СССР, М.: 1972.
[14]. Duval P., Matyas R., Grover A. Continuous integration improving Software quality and reducing risk.-Addison Wesley, 2009.-691 p.
[15]. Коваль Г.И. Модели и методы инженерии качества систем на этапах ЖЦ.- Реф. дис. ИК НАНУ, 2005.-20 с.
[16]. Андон Ф.И., Коваль Г.И. и др. Основы инженерии качества программных систем.- К.: Наук. думка, 2007.- 670 с.
[17]. Безопасность ракетно-космической техники и надежность компьютерных систем / А.В. Горбенко, С.А. Засуха, В.И. Рубан и др.] // Авиационно-космическая техника и технология. – 2011. – №1(78). – С. 9–20.

[18]. Basic Concepts and Taxonomy of Dependable and Secure Computing / A. Avizienis, J.-C. Laprie, B. Randell, C. Landwehr // IEEE Transactions on Dependable and Secure Computing. – 2004. – Vol. 1, No. 1. – C. 11-33.
[19]. IEC 62628. Guidance on software aspects of dependability.– Geneva: IEC, 2011.– 63 p.
[20]. ISO 15288:2002. Systems Engineering. Cycle Life Processes of Systems.
[21]. Лаврищева Е.М. Методы программирования. Теория, инженерия, практика. -К.: 2006.- Наук.Думка.- 371 с.
[22]. Лаврищева Е.М., Грищенко В.Н. Сборочное программирование. Основы индустрии программных продуктов. - К.: Наук. думка, 2009.-371с.
[23]. Лаврищева Е.М. Software Engineering компьютерных систем. Парадигмы, технологии, CASE-средства. – К.: Наук.думка, 2014. - 284 с.
[24]. Саати Т. Принятие решений. Метод анализа иерархий. – М.: Радио и связь, 1993. - 315 с.

Analysis of methods for assessing the reliability of equipment and systems. Practice of methods

[1]N.V. Pakulin<npak@ispras.ru>
[2]E.M. Lavrischeva <lavr@ispras.ru>
[3]A.G. Ryzhov <ryzhov@ispras.ru>
[1,2,3] Institute for System Programming of the Russian Academy of Sciences,
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia,
[2]Moscow Institute of physics and technology (MIPT)
141700, Russia, Moscow region, Dolgoprudny, Campus per., 9.

Abstract. The analysis of models and methods of reliability evaluation of hardware and software is carried out. The basic concepts of reliability and safety methods of such systems and situations leading to errors, defects and failures are defined. The definition of reliability and safety of technical systems and software systems is given. The classification of reliability models: predictive, measuring and evaluation types. Evaluation models that are used more in practice are described. The standard of Software life cycle (ISO 15288:2002) is defined, focused on the development and control of system components for errors, starting with the system requirements. The results of application of reliability models (Moussa, Goel-Okomoto, etc.) to small, medium and large projects are presented and their comparative assessment is given. The technological module (TM) of reliability evaluation of complex software systems VPK (1989) is described. The quality model of the standard ISO 9126 (1-4): 2002-2004 with indicators of functionality, reliability, efficiency, etc., which are used in determining the maturity and certificate of the product is shown.

Keywords: reliability, model, method, error, defect, failure, random process, safety, dependability, recoverability, fault tolerance, completeness, reliability assessment, quality certificate.

REFERENCE

[1]. Lipaev V. V. Software Reliability. – M.: SINTEG, 1998.- 231p.
[2]. Lipaev V. V. Methods of quality assurance of large-scale software systems. –

M.: SINTEG, 2003.-510 p.
[3]. Myers G. Software Reliability, - M.: Mir, 1980.- 360p.
[4]. Moroz G. B., lavrisheva E. M. Models of software reliability growth.- K.: Preprint 92-38, 1992.- 23p.
[5]. Lipaev V. V. Reliability and functional safety of software systems real time.- Moscow, JSC "Svetlitsa", 2013.- 193 p.
[6]. Shick G.J., Wolverton R.W. An analysis of computing software reliability models /IEEE Tras. Software Eng. – V. SE–4. – № 2. – 1978. – P. 104–120.
[7]. Shanthikumar J.G. Software reliability models: A Review // Microelectron. Reliab. – 1983. –V. 23. –№ 5 – P. 903–943.
[8]. Goel Amrit L., "Software reliability models: Assumptions, limitations, and applicability. //IEEE Transactions on Software Engineering, Vol. SE–11, № 12. – 1985. –P. 1411–1423.
[9]. Musa J.D. Okumoto K. A. Logarithmic Poisson Time Model for Software Reliability Measurement //Proc. 7- International Conference on Software Engineering. – Orlando, Florida. – 1984. – P. 230–238.
[10]. Yamada S., Ohba M., Osaki S. S–shaped software reliability grows modeling for software error detection // IEEE Trans. Reliability. – 1983. – R–32. – № 5. – P. 475–478.
[11]. Chulani S. Constructive quality modeling for defect density prediction: COQUALMO // International Symposium on Software Reliability Engineering (ISSRE'99), Boca Raton, N. 1–4, 1999.
[12]. Gnedenko B. V., Kovalenko I. N. Introduction to the theory of mass service.- Science, M.: 1966.
[13]. Gnedenko B. V., Shpak V. D. Probabilistic characteristics of complex systems. - News of the USSR Academy of Sciences, M.: 1972.
[14]. Duval P., Matyas R., Grover A. Continuous integration improving Software quality and reducing risk.-Addison Wesley, 2009.-691 p.
[15]. Koval G.I., Models and methods for engineering quality systems at the stages CL.- Ref. dis. IK NANU, 2005.- 20 p.
[16]. Andon F. I., Koval G. I. and others. Bases of quality engineering software system.- K.: Of Sciences. Dumka, 2007.- 670 p.
[17]. Gorbenko A.V., Drought S. A., Ruban V. I., etc. the Safety of rocket-space engineering and reliability of computer systems // Aerospace technics and technology. - 2011. - №1 (78). - P. 9-20.
[18]. Avizienis A., Laprie J.-C., Randell B., C. Landwehr C. Basic Concepts and Taxonomy of Dependable and Secure Computing / // IEEE Transactions on Dependable and Secure Computing. – 2004. – Vol. 1, No. 1. – C. 11-33.
[19]. IEC 62628. Guidance on software aspects of dependability. – Geneva: IEC, 2011. – 63 p.
[20]. ISO 15288:2002. Systems Engineering. Cycle Life Processes of Systems.
[21]. Lavrischeva E. M. Programming Methods. Theory, engineering, practice. – K.: 2006.- Sciences'.Dumka.- 371 p.
[22]. Lavrisheva E. M., Grishchenko V. N. Assembly programming. Basics software industries. - K.: Sciences dumka, 2009 - 371p.
[23]. Lavrisheva E. M. Software Engineering of computer systems. Paradigms, technologies, CASE-means. – K.: Sciences dumka, 2014. - 284 p.
[24]. Saati T. Decision-Making. Method of hierarchy analysis. – M.: Radio and communications, 1993. - 315p. (in Russian).