НАДЕЖНОСТЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ. ВИДЫ И КРИТИЧНОСТЬ ОШИБОК .

Дроботун Е. Б.

Военная академия воздушно – космической обороны, г.Тверь

201074@nwgsm.ru

В работе рассматриваются качество и надежность программного обеспечения. Приводятся различные показатели надежности программного обеспечения, описываются некоторые подходы к оценке надежности программного обеспечения и анализу видов, последствий и критичности ошибок в программном обеспечении.

Источниками ошибок в программном обеспечении являются специалисты – конкретные люди с их индивидуальными особенностями, квалификацией, талантом и опытом. Вследствие этого плотность потоков ошибок и размеры необходимых корректировок в модулях и компонентах при разработке и сопровождении программного обеспечения могут различаться в десятки раз. Однако в крупных комплексах программ статистика и распределение ошибок и типов выполняемых изменений, необходимых для их исправления, для коллективов разных специалистов нивелируются и проявляются общие закономерности, которые могут использоваться как ориентиры при выявлении ошибок и их систематизации. Этому могут помогать оценки типовых ошибок, модификаций и корректировок путем их накопления и обобщения по опыту создания определенных классов программного обеспечения.

Оценка качества программного обеспечения могут проводиться с двух позиций: с *позиции положительной* эффективности и непосредственной адекватности их характеристик назначению, целям создания и применения, а также с *негативной позиции* возможного при этом ущерба – риска от использования ПС или системы. Показатели качества преимущественно отражают положительный эффект от применения программного обеспечения и основная задача разработчиков проекта состоит в обеспечении высоких значений качества. Риски характеризуют возможные негативные последствия проявившихся в ходе эксплуатации ошибок или ущерб для пользователя при применении и функционировании программного обеспечения.

Согласно [ГОСТ 9126] качество программного обеспечения это весь объем признаков и характеристик программного обеспечения, который относится к ее способности удовлетворять установленным или предполагаемым потребностям.

Качество программного обеспечения оценивается следующими характеристиками:

Функциональные возможности (Functionality). Набор атрибутов, относящихся к сути набора функций и их конкретным свойствам. Функциями являются те, которые реализуют установленные или предполагаемые потребности.

Надежность (Reliability). Набор атрибутов относящихся к способности программного обеспечения сохранять свой уровень качества функционирования при установленных условиях за установленный период времени.

Практичность (Usability). Набор атрибутов, относящихся к объему работ, требуемых для использования и индивидуальной оценки такого использования определенным и предполагаемым кругом пользователей.

Эффективность (Efficiencies). Набор атрибутов, относящихся к соотношению между уровнем качества функционирования программного обеспечения и объемом используемых ресурсов при установленных условиях.

Сопровождаемость (Maintainability). Набор атрибутов, относящихся к объему работ, требуемых для проведения конкретных изменений (модификаций).

Мобильность (Portability). Набор атрибутов, относящихся к способности программного обеспечения быть перенесенным из одного окружения в другое.

В общем случае под ошибкой подразумевается неправильность, погрешность или неумышленное искажение объекта или процесса, что может быть причиной ущерба – риска при функционировании или применении программы [1]. При этом предполагается, что известно правильное, эталонное состояние объекта или процесса по отношению к которому может быть определено наличие отклонения. Исходным эталоном для любого программного обеспечения являются спецификации требований заказчика или потенциального пользователя, предъявляемых к программам и ожидаемый пользователем или заказчиком эффект от использования программного обеспечения. Важной особенностью при этом является отсутствие полностью определенной программы – эталона, которой должны соответствовать текст и результаты функционирования разрабатываемой программы. Поэтому определить качество программного обеспечения и наличие ошибок в нем путем сравнения разрабатываемой программы с эталонной программой невозможно.

Риски проявляются как негативные последствия проявления ошибок в программном обеспечении в ходе его применения и функционирования, которые могут нанести ущерб системе, в которой применяется это программное обеспечение, внешней среде или пользователям этой системы в результате отклонения характеристик программного обеспечения заданных или ожидаемых пользователем или заказчиком.

Исходя из определения ошибки в программном обеспечении, приведенном выше, можно сделать вывод, что ошибки, проявляющиеся в ходе функционирования программного обеспечения, могут влиять на все показатели качества. В работе рассматриваются ошибки, проявления которых влияют на надежность функционирования программного обеспечения.

По определению, установленному в [2], *надежность* – свойство объекта выполнять заданные функции, сохраняя во времени значения установленных эксплуатационных показателей в заданных пределах, соответствующим заданным режимам и условиям использования, технического обслуживания, ремонта, хранения и транспортирования.

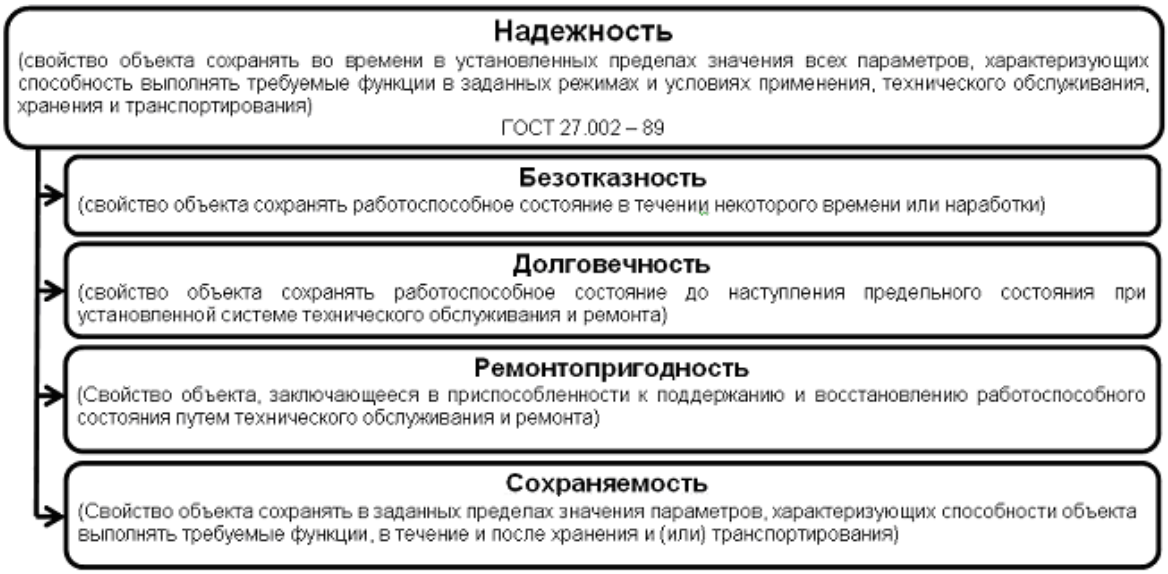

Рис. 1. Надежность по ГОСТ 27.002 – 89

При этом надежность является комплексным свойством, которое в зависимости от назначения объекта и условий его применения может включать безотказность, долговечность, ремонтопригодность и сохраняемость или определенные сочетания этих свойств (рис. 1). Поскольку программное обеспечение в процессе эксплуатации не изнашивается, его поломка и ремонт в общепринятом смысле не производится, то надежность программного обеспечения имеет смысл характеризовать только с точки зрения

безотказности его функционирования и возможности восстановления функционирования после отказов вызванных проявлениями ошибок.

В [3] надежность программного обеспечения предлагается характеризовать с помощью следующих характеристик (рис. 2): стабильность, устойчивость и восстанавливаемость.

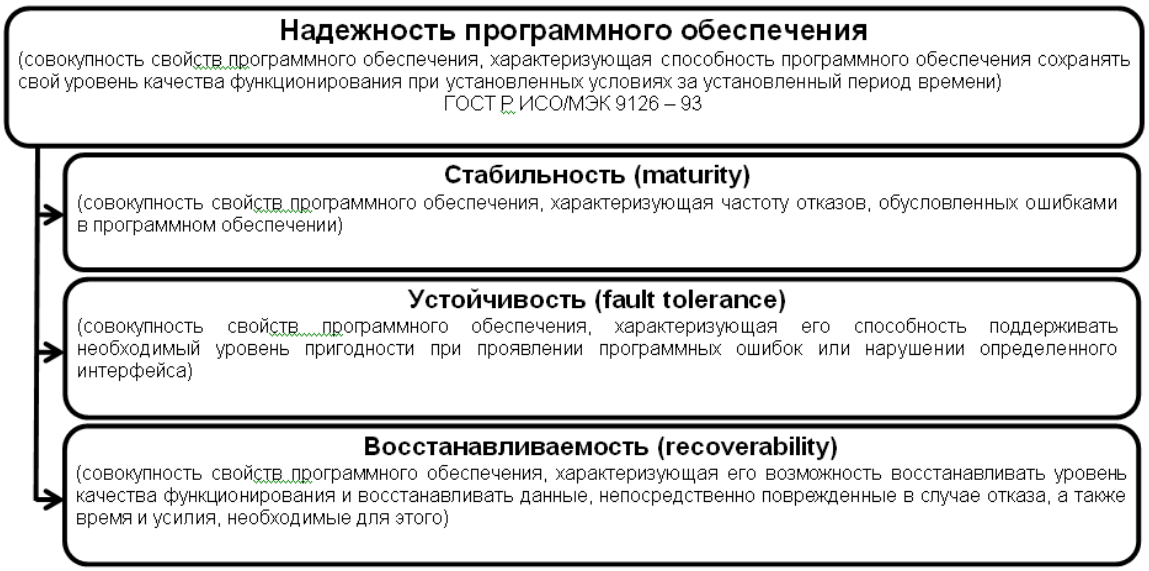

Рис. 2. Надежность программного обеспечения

В этом случае стабильность и устойчивость характеризуют безотказность программного обеспечения, а восстанавливаемость – возможность восстановления функционирования программного обеспечения после его отказа. Для количественной оценки надежности программного обеспечения необходимо определить показатели надежности для каждого свойства и методику их определения (оценки).

Для оценки стабильности программного обеспечения возможно использование показателей характеризующих безотказность технических устройств [2] (рис. 3).

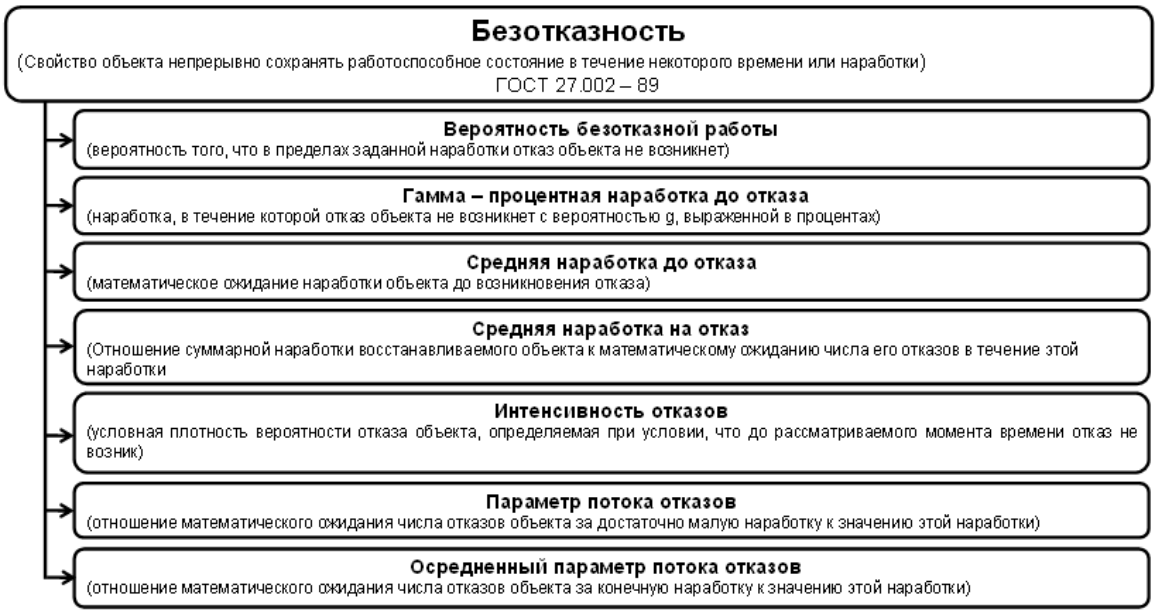

Рис. 3. Показатели безотказности

В большинстве случаев поток программных ошибок может быть описан негомогенным процессом Пуассона [4]. Это означает, что программные ошибки происходят в статистически независимые моменты времени, наработки подчиняются экспоненциальному распределению, а интенсивность проявления ошибок изменяется во времени. Обычно используют убывающую интенсивность проявления ошибок. Это означает, что ошибки, как только они выявлены, эффективно устраняются без введения новых ошибок. Главная цель анализа надежности программного обеспечения заключается в том, чтобы определить форму функции интенсивности проявления ошибок и оценить ее параметры по наблюдаемым данным. Как только функция интенсивности проявления ошибок определена, могут быть найдены такие показатели надежности как:

- общее количество ошибок;
- количество остающихся ошибок;
- время до проявления следующей ошибки;
- вероятность безошибочной работы;
- интенсивность проявления ошибок;
- остаточное время испытаний (до принятия решения);
- максимальное количество ошибок (относительно срока службы).

При этом следует различать понятия *ошибка* и *отказ.* Применительно к надежности программного обеспечения ошибка это погрешность или искажение кода программы, неумышленно внесенные в нее в процессе разработки, которые в ходе функционирования этой программы могут вызвать отказ или снижение эффективности функционирования. Под отказом в общем случае понимают событие, заключающееся в нарушении работоспособности объекта [2]. Состояние объекта, при котором значения всех параметров характеризующих способность выполнять заданные функции, соответствуют требованиям нормативно – технической и (или) конструкторской (проектной) документации – называется работоспособным. При этом критерии отказов, как признаки или совокупность признаков нарушения работоспособного состояния программного обеспечения, должны определяться исходя из его предназначения в нормативно – технической и (или) конструкторской (проектной) документации.

В общем случае отказ программного обеспечения можно определить как:

прекращение функционирования программы (искажения нормального хода ее выполнения, зацикливание) на время превышающее заданный порог;

прекращение функционирования программы (искажения нормального хода ее выполнения, зацикливание) на время не превышающее заданный порог, но с потерей всех или части обрабатываемых данных;

прекращение функционирования программы (искажения нормального хода ее выполнения, зацикливание) потребовавшее перезагрузки ЭВМ, на которой функционирует программное обеспечение.

При этом исходя из [2], все отказы в программном обеспечении следует трактовать как сбои (самоустраняющиеся отказы или однократные отказы, устраняемые незначительным вмешательством оператора), поскольку восстановление работоспособного состояния программного обеспечения может произойти без вмешательства оператора (перезагрузка ЭВМ не требуется), либо при участии оператора или эксплуатирующего персонала (перезагрузка ЭВМ необходима).

Приведенные выше критерии отказов приводят к необходимости анализа временных характеристик функционирования программы и динамических характеристик потребителей данных, полученных в ходе функционирования программного обеспечения. Временная зона перерыва нормальной выдачи информации и потери работоспособности, которую следует рассматривать как зону сбоя (отказа), тем шире, чем более инертный объект находится под воздействием данных, полученным в ходе работы программы. Пороговое время восстановления работоспособного состояния системы, при превышении которого следует

фиксировать отказ, близко к периоду решения задач для подготовки информации (данных) соответствующему потребителю (абоненту).

Для любого потребителя данных существует допустимое время отсутствия данных от программы, при котором его характеристики находятся в допустимых пределах. Исходя из этого времени, можно установить границы временной зоны, которая разделяет работоспособное и неработоспособное состояние программного обеспечения и позволяет использовать данные критерии отказов.

Из приведенного выше определения программной ошибки с точки зрения надежности, можно сделать вывод о том, что ошибки, при их проявлении, не всегда вызывают отказ программного обеспечения и каждую ошибку можно характеризовать условной вероятностью возникновения отказа при проявлении этой ошибки. Следует также отметить, что само по себе наличие ошибки в исходном коде не определяет надежность программы до тех пор, пока не произойдет проявления этой ошибки, поэтому пользоваться для оценки надежности программного обеспечения только показателями характеризующие общее количество ошибок в программе, количество оставшихся ошибок и максимального количества ошибок нельзя.

В [5] стабильность предлагается оценивать вероятностью безотказной работы, которая оценивается исходя из модели относительной частоты, при этом применение ее ограничено периодом эксплуатации программного обеспечения, что не всегда приемлемо, поскольку надежность объекта, как правило, необходимо оценивать не только в процессе его эксплуатации, но и до начала эксплуатации этого объекта. Ограничение модели относительной частоты вызвано тем, что в этой модели не учитываются процессы тестирования и отладки, а конкретно то, что при возникновении отказа программного обеспечения, ошибка, вызвавшая этот отказ, исправляется.

Наиболее приемлемыми показателями характеризующими стабильность (безотказность) программного обеспечения представляются показатели сходные с показателями безотказности технических систем: вероятность безотказной работы, интенсивность отказов, и среднее время наработки на отказ. Эти показатели взаимосвязаны и, зная один из них, можно определить другие [2]. При определении этих показателей в большинстве случаев можно исходить из модели надежности, предполагающей, что интенсивность проявления ошибок убывает по мере исправления этих ошибок, время между проявлениями ошибок распределено экспоненциально, а интенсивность проявления ошибок постоянна между двумя соседними проявлениями ошибок. Применение такой модели надежности программного обеспечения позволит оценить надежность программного обеспечения во время тестирования и отладки.

Устойчивость, как свойство или совокупность свойств программного обеспечения, характеризующие его возможность поддерживать приемлемый уровень функционирования при проявлениях ошибок в нем, можно оценивать условной вероятностью безотказной работы при проявлении ошибки. Согласно [5] устойчивость оценивается с помощью трех метрик, включающих двадцать оценочных элементов (рис. 4). Результаты оценки каждой метрики определяются результатами оценки определяющих ее оценочных элементов, а результат оценки устойчивости определяются результатами соответствующих ему метрик. Программное обеспечение по каждому из оценочных элементов оценивается группой экспертов – специалистов, компетентных в решении данной задачи, на базе их опыта и интуиции. Для оценочных элементов принимается единая шкала оценки от 0 до 1.

Недостатком такого подхода является одинаковая оценка устойчивости для всех возможных ошибок. Поскольку вероятность возникновения отказа при проявлении разных ошибок может быть разной, возникает необходимость разделения ошибок на несколько категорий. Признаком, по которому в этом случае можно относить ошибки к той или иной категории, можно считать тяжесть ошибки. Под тяжестью ошибки в этом случае следует понимать количественную или качественную оценку вероятного ущерба при проявлении

этой ошибки [6], а если говорить о надежности, то оценку вероятности возникновения отказа при проявлении ошибки. При этом категорией тяжести последствий ошибки будет являться классификационная группа ошибок по тяжести их последствий, характеризуемая определенным сочетанием качественных и/или количественных учитываемых составляющих ожидаемого (вероятного) отказа или нанесенного отказом ущерба.

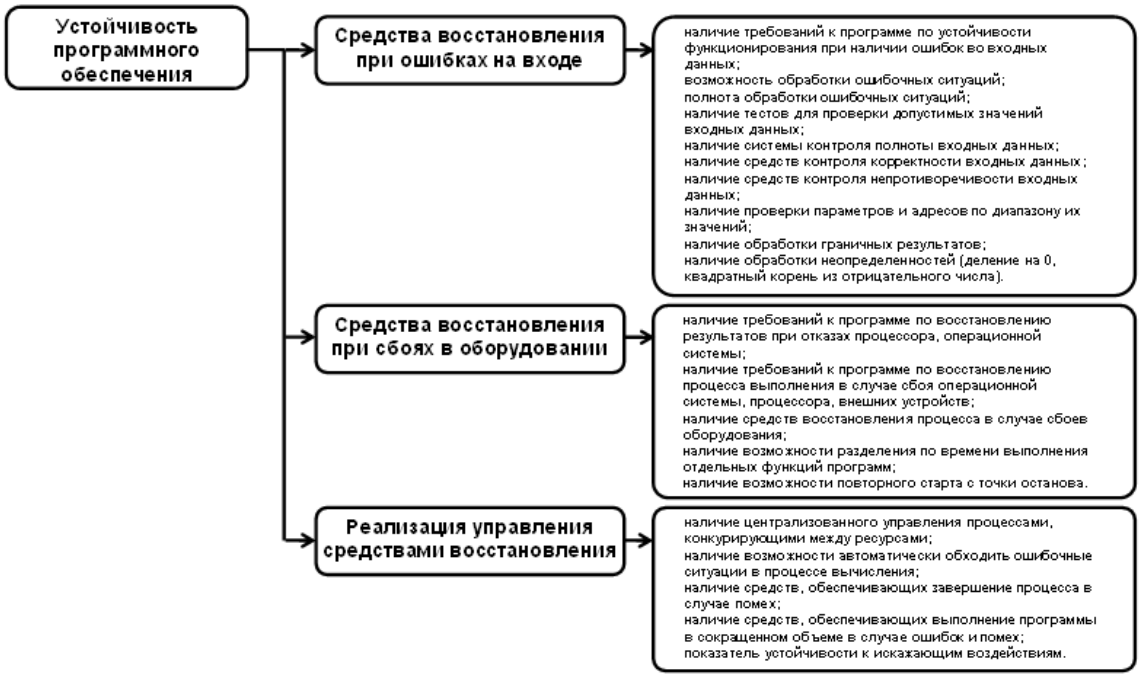

Рис. 4. Метрики и оценочные элементы устойчивости программного обеспечения по ГОСТ 28195 – 89

В качестве показателя степени тяжести ошибки, позволяющего дать количественную оценку тяжести проявления последствий ошибки целесообразно использовать условную вероятность отказа и его возможных последствий при проявлении ошибок разных категорий. Для программного обеспечения, создаваемого для систем управления, потеря работоспособности которых может повлечь за собой катастрофические последствия, возможные категории тяжести ошибок приведены в таблице 1.

Таблица 1. Категории тяжести ошибки в программном обеспечении, нарушение работоспособности которого могут привести к катастрофическим последствиям

Номер категории ошибки	Наименование категории тяжести ошибки	Описание последствий проявления ошибки
IV	Катастрофическая	проявление ошибки с высокой вероятностью влечет за собой прекращение функционирования программного обеспечения (его отказ) и может вызвать повреждение системы и окружающей среды и/или гибель и тяжелые травмы людей
III	Критическая	проявление ошибки с высокой вероятностью влечет за собой прекращение функционирования программного обеспечения (его отказ), может вызвать повреждение системы и окружающей среды, но не угрожает жизни и здоровью людей
II	Существенная	проявление ошибки влечет за собой снижение эффективности функционирования программного обеспечения и может вызвать прекращение функционирования программного обеспечения (его отказ) без заметного повреждения системы и угрозы жизни и здоровью людей
I	Несущественная	проявление ошибки может повлечь за собой снижение эффективности функционирования программного обеспечения и практически не приводит к возникновению отказа в нем (вероятность возникновения отказа очень низкая)

Для программного обеспечения общего применения или программного обеспечения систем, нарушение работоспособности которых не представляет угрозы жизни людей и не приводит к разрушению самой системы, возможные категории тяжести приведены в таблице 2.

Таблица 2. Категории тяжести ошибки в программном обеспечении, нарушение работоспособности которого не приводят к катастрофическим последствиям

Номер категории ошибки	Наименование категории тяжести ошибки	Описание последствий проявления ошибки
III	Критическая	проявление ошибки с высокой вероятностью влечет за собой прекращение функционирования программного обеспечения (его отказ)
II	Существенная	проявление ошибки влечет за собой снижение эффективности функционирования программного обеспечения и может вызвать прекращение функционирования программного обеспечения (его отказ)
I	Несущественная	проявление ошибки может повлечь за собой снижение эффективности функционирования программного обеспечения и практически не приводит к возникновению отказа в нем (вероятность возникновения отказа очень низкая)

Приведенные категории тяжести ошибки определены с помощью методов изложенных в [6; 7]. В этих документах установлены методы анализа видов и последствий отказов, видов, последствий и критичности отказов и даны рекомендации по их применению.

Оценку степени тяжести ошибки как условной вероятности возникновения отказа (последствий этого отказа), можно производить согласно [5], используя метрики и оценочные элементы, характеризующие устойчивость программного обеспечения. При этом оценка производится для каждой ошибки в отдельности, а не для всего программного обеспечения. Далее исходя из проведенных оценок возможно определение устойчивости программного обеспечения к проявлениям ошибок каждой из категорий.

Восстанавливаемость программного обеспечения, как свойство или совокупность свойств характеризующих способность программного обеспечения восстановления своего уровня пригодности и восстановления данных, непосредственно поврежденных вследствии проявлении ошибки (отказа), характеризуется полнотой и длительностью восстановления функционирования программ в процессе перезапуска или перезагрузки ЭВМ. В [5] восстанавливаемость предлагается оценивать по среднему времени восстановления. При этом следует учитывать, что время восстановления функционирования программного обеспечения складывается не только из времени потребного для перезагрузки ЭВМ и загрузки самого программного обеспечения, но и из времени необходимого для восстановления данных и это время в ряде случаев может значительно превышать время перезагрузки.

Показатели надежности программного обеспечения в значительной степени адекватны аналогичным характеристикам, принятых для других технических систем. Наиболее широко используется показатель наработки на отказ. Наработка на отказ – это отношение суммарной наработки объекта к математическому ожиданию числа его отказов в течении этой наработки. Для программного обеспечения использование данного показателя затруднено, в силу особенностей тестирования и отладки программного обеспечения (ошибка вызвавшая отказ, как правило, исправляется и больше не повторяется). Поэтому целесообразно использовать показатель средней наработки до отказа – математического ожидания времени функционирования программного обеспечения до отказа. При использовании модели надежности программного обеспечения предполагающей экспоненциальное распределение времени между отказами, среднее время наработки до отказа равно величине обратной интенсивности отказов. Интенсивность отказов можно оценить исходя из оценок стабильности и устойчивости программного обеспечения. Обобщение характеристик отказов

и восстановлений производится в показателе коэффициент готовности [2]. Коэффициент готовности программного обеспечения это вероятность того, что программное обеспечение окажется в работоспособном состоянии в произвольный момент времени. Значение коэффициента готовности соответствует доле времени полезной работы программного обеспечения на достаточно большом интервале времени, содержащем отказы и восстановления.

Литература

1. *Липаев В. В. /* Программная инженерия. Методологические основы. // М.: ТЕИС, 2006.
2. ГОСТ 27.002 – 89. Надежность в технике. Основные понятия. Термины и определения. // М.: Издательство стандартов, 1990.
3. ГОСТ Р ИСО/МЭК 9126 – 93. Информационная технология. Оценка программной продукции. Характеристики качества и руководства по их применению. // М.: Издательство стандартов, 1994.
4. ГОСТ 51901.5 – 2005. Менеджмент риска. Руководство по применению методов анализа надежности. // М.: Издательство стандартов, 2007.
5. ГОСТ 28195 – 89. Оценка качества программных средств. Общие положения. // М.: Издательство стандартов, 1989.
6. ГОСТ 27.310 – 95. Надежность в технике. Анализ видов, последствий и критичности отказов. // М.: Издательство стандартов, 1995.
7. ГОСТ 51901.12 – 2007. Менеджмент риска. Метод анализа видов и последствий отказов. // М.: Издательство стандартов, 2007.