ОСОБЕННОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ КАК ОБЪЕКТА НАДЕЖНОСТИ

- Программа представляет собой закодированную в виде инструкций на некотором языке программирования информацию о способе преобразования входных данных в выходные. Ее отказ есть событие, состоящее в прекращении выполнения требуемых функций с учетом заданных ограничений.

Основной источник ненадежности программы – ошибки, сделанные разработчиками программ, на разных стадиях проектирования. Истинные причины возникновения ошибок в программных системах состоят в том, что:

- сложность ПО как по объему элементов, так и по структуре и взаимной связности как правило существенно выше, в то время как аппаратная система проектируется из сравнительно небольшого числа типовых элементов;
- используются более значительные наборы входных данных;
- имеется большее число внутренних состояний;
- длиннее зависимости между состояниями во времени;

- аппаратные решения меньше подвержены изменениям;
- для технических систем характерно меньшее взаимное влияние элементов друг на друга;
- программные ошибки в большей степени определяются «человеческим фактором», а аппаратные внешними условиями.

Следует также отметить, что:

- отказы не имеют случайной природы, а возникают тогда, когда повторяются вызвавшие их условия;
- отказы не зависят от времени, ПО, которое не эксплуатируется, не может отказать;
- при оценке надежности ПО не существует понятия «выборка», ибо каждый экземпляр программного модуля или компонента является клоном, точно копирующим дефекты всех других экземпляров.

- При оценке и анализе надежности технической системы модели надежности ее элементов, как правило, известны, так как они определяются на этапе их разработки и изготовления. При этом определение надежности элементов производится путем проверки достаточно большого числа этих элементов, с фиксацией сбоев и отказов. Такие модели надежности элементов позволяют получить оценку надежности всего устройства.

При оценке надежности программ:

- может быть использовано большое количество разных моделей надежности, описывающих один и тот же процесс;
- отсутствует универсальная модели надежности, которую можно было бы применить для описания любых программных систем с удовлетворительной точностью;
- в существующих моделях надежности отсутствует возможность учета «вторичных дефектов»

Процедура оценки надежности программной системы усложняется целым набором факторов, к числу которых можно отнести:

- трудность проведения декомпозиции (разделения на части), что приводит к трудностям при измерении надежности каждой части;
- параллельное тестирование двух экземпляров ПО дает почти полностью зависимые результаты;
- параллельное тестирование нескольких экземпляров имеет смысл только при различном сочетании входных данных и управляющих воздействий, однако при этом сложнее оценивать корреляцию проводимых тестов и обрабатывать полученные результаты;
- количество всех возможных состояний для любой нетривиальной программы насколько велико что не позволяет осуществить полный объем тестовых испытаний.

Исходя из приведенных выше аргументов можно заключить, что:

- отказ ПО не может рассматриваться с точки зрения общепринятой теории надежности технических систем, это самостоятельное понятие, применимое к информационным продуктам и связанное с ошибками программирования;
- использование понятия интенсивности отказов также не имеет смысла;
- надежность программы следует рассматривать в контексте информационной достоверности ее работы;
- теория вероятностей не представляет собой метод, который может эффективно использоваться для исследования процессов, происходящие в ПО.

ОШИБКИ ПО

Дефект или ошибка в программе представляет собой неправильность, погрешность или неумышленное искажение объекта или процесса, что может быть причиной ущерба – неверного результата вычисления, невозможность выполнения требуемой функции, отказ в дальнейшем применении программного продукта и другие последствия.

Для того, чтобы обнаружить факт возникновения ошибки, должно быть известно или задано правильное, эталонное состояние информационного объекта или процесса, относительно которого определяется отклонение. В качестве такого эталона выступает обычно техническое задание со спецификацией требований заказчика. В этом документе должны быть определены состав, содержание и значения результатов использования программы, которые при определенных условиях и исходных данных должен получать пользователь.

Следует отметить, что для программного кода характерной особенностью является отсутствие полностью определенной программы-эталона, которой должны соответствовать результаты работы проектируемой программы. то является существенной особенностью процесса выявления ошибок в ПО. По этой причине установить наличие и локализовать ошибку в программе путем ее сравнения непосредственно с эталонной невозможно. Кроме того, при отладке и тестировании могут сначала обнаруживаются вторичные ошибки, которые являются последствиями и результаты проявления других внутренних дефектов или недостаточно корректного определения требований и порядка функционирования программы.

Значительное число ошибок вносится на стадии кодирования – эти ошибки находятся в исходном коде программы. Будем рассматривать в основном такие ошибки.

Функциональные ошибки – нарушения программной спецификации (несоответствие функциональным или нефункциональным требованиям). Приводят к ухудшению функциональности ПО (пригодность, точность).

Нефункциональные ошибки – нарушения правил языка программирования, неправильное использование библиотечных функций и т.п. Приводят к снижению надежности (зрелости) и ухудшению функциональности (защищенности):

- ошибки в последовательных программах;
- ошибки синхронизации.

Основные виды нефункциональных ошибок в последовательных программах на языке C:

- ошибки использования неинициализированного, освобожденного указателя или указателя на NULL;
- утечки ресурсов, в том числе динамической памяти;
- ошибочно игнорируемые участки кода;
- ошибки отсутствия инициализации интервальных переменных;
- ошибки выхода за границы статических и динамических объектов;
- отсутствие проверки возвращаемого значения функций.

Сама по себе ошибки в программе является ненаблюдаемой, увидеть и оценить можно не саму ошибку, а результат ее проявления, который по аналогии с техническими системами принято называть отказом. Надежность связана с частотой проявления ошибок, но не с их количеством – разные ошибки имеют разную частоту проявления. При этом, отказ может быть следствием не одной, а сразу нескольких ошибок, причем ошибки могут компенсировать друг друга, при этом может возникнуть ситуация, когда после исправления ошибки интенсивность отказов возрастает. Исправление ошибки, так же как и любое другое изменение ПО приводит к новой программе, характеризующейся уже другими показателями надежности.

Для ПО систем управления объектов, потеря работоспособности которых может повлечь за собой катастрофические последствия, могут использоваться категории тяжести ошибок приведены в таблице ниже.

КАТЕГОРИИ ТЯЖЕСТИ ОШИБОК ПО

Номер категории ошибки	Наименование категории ошибки	Описание последствий проявления ошибки
IV	Катастрофическая	Проявление ошибки с высокой вероятностью влечет за собой прекращение функционирования ПО (отказ) и может вызвать повреждение системы автоматизации, объекта управления и окружающей среды, а также гибель и травмы людей
III	Критическая	Проявление ошибки с высокой вероятностью влечет за собой прекращение функционирования ПО (отказ) и может вызвать повреждение системы автоматизации, но не угрожает объекту управления, окружающей среде, жизни и здоровью человека
II	Существенная	Проявление ошибки влечет за собой снижение эффективности функционирования ПО и может вызвать прекращение его работы (отказ) без заметного повреждения системы, объекта автоматизации, окружающей среды, жизни и здоровью людей
I	Несущественная	Проявление ошибки может повлечь за собой снижение эффективности ПО, но практически не приводит к возникновению отказа в нем.