

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. Это 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.