НАДЕЖНОСТЬ ФУНКЦИОНИРОВАНИЯ ПРОГРАММНЫХ СРЕДСТВ

Основные понятия и показатели надежности ПС

Надежность ПС - свойство ПС выполнять заданные функции, сохраняя во времени значения установленных эксплуатационных показателей в заданных пределах

Надежность технических систем определяется в основном двумя фактора-ми: ***надежностью компонент*** *и* ***дефектами в конструкции***

Надежность сложных программных средств определяется этими же факторами, однако доминирующими являются **дефекты** и **ошибки проектирования**

Источниками ненадежности являются непроверенные сочетания исходных данных, при которых функционирующее ПС дает неверные результаты или **отказы** (случайное изменение исходных данных при обработке информации, множество условных переходов создают огромное число маршрутов исполнения каждого сложного ПС).

Особенности надежности ПС 2

Предметом изучения **теории надежности комплексов программ** (Software Reliability) является работоспособность сложных программ обработки информации **в реальном времени при взаимодействии с внешней средой**.

Традиционные **методы испытаний надежности** путем физического воздействия на их компоненты не применимы для ПС и их следует заменять на методы **форсированного воздействия информационных потоков внешней среды.**

Задачи **теории и анализа надежности** сложных ПС:

- исследование дефектов и ошибок, динамики их изменения при отладке и сопровождении,
- исследование методов и средств контроля и защиты от искажений программ, вычислительного процесса и данных путем использования различных видов избыточности и помехозащиты,

Диагноз состояния ПС делится на **тестовый** (*используются специально подготовленные исходные данные и эталонные результаты*)и **функциональный** (*на базе реальных исходных данных*).

Надежные программы д.б. устойчивы к различным негативным возмущениям (естественного и искусственного происхождения) и способны сохранять требуемое качество результатов в реальных условиях функционирования

http://www.nspru.ru/downloads/iso9126.pdf

ISO 9126: Информационная технология. Оценка ПО. Характеристики

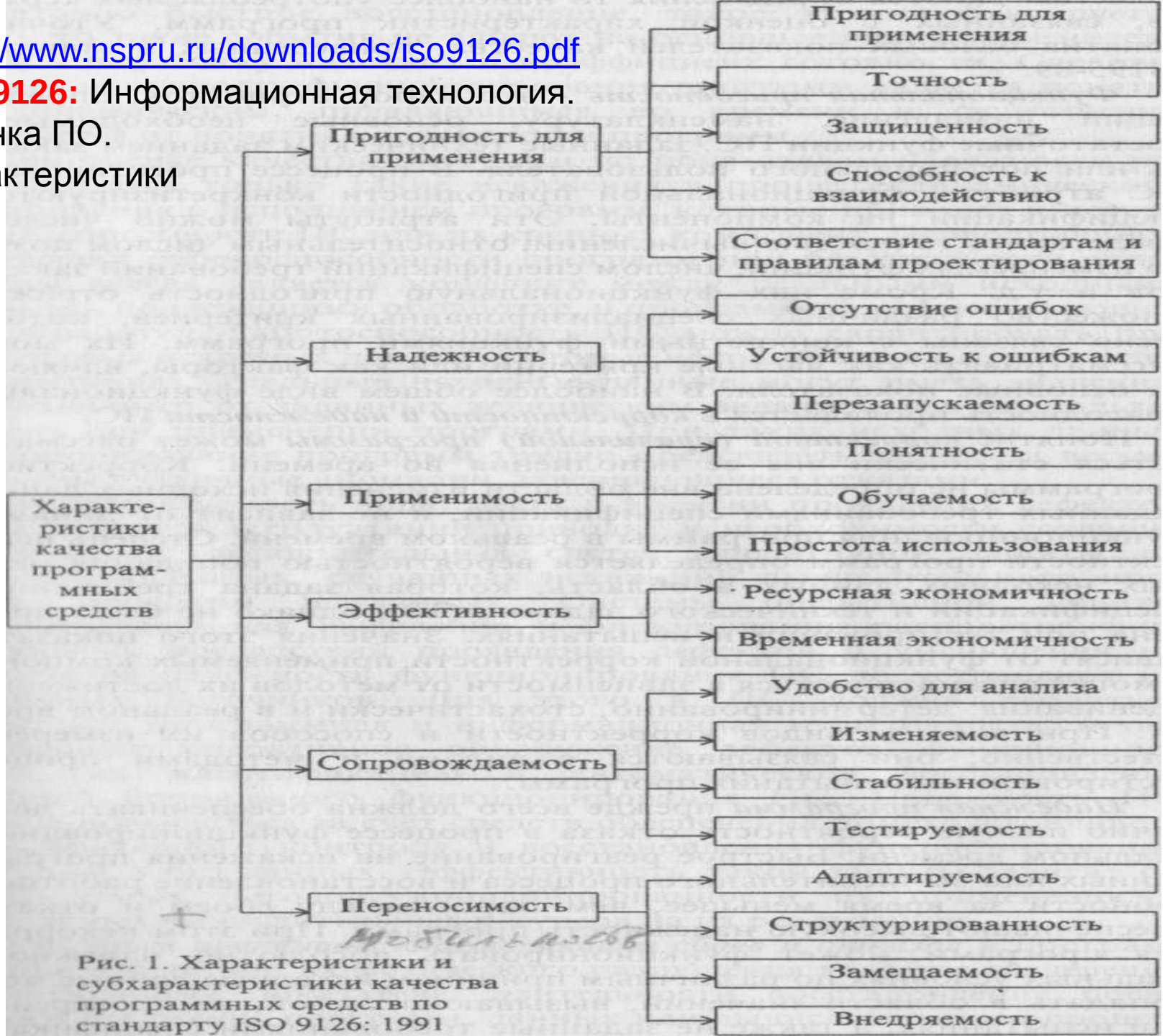


Рис. 1. Характеристики и субхарактеристики качества программных средств по стандарту ISO 9126: 1991

- Характеристики и субхарактеристики в стандарте определены очень кратко, без комментариев и рекомендаций по их применению к конкретным системам и проектам.
- Близким к описанному стандарту по идеологии, структуре и содержанию является стандарт **ГОСТ 28195-89**.
- В стандарте **ГОСТ 28806-90** формализуются общие понятия программы, программного средства, программного продукта
- ***В программах и данных всегда остаются дефекты и ошибки***, часть которых выявляется в процессе эксплуатации ПС в реальной среде.
- Для ***удостоверения качества, надежности и безопасности применения*** ПС следует подвергать ***обязательной сертификации*** аттестованными, **проблемно-ориентированными испытательными лабораториями**
- Если все испытания проходят успешно, то на версию ПС оформляется специальный документ - ***сертификат соответствия.***

Цели и виды сертификационных испытаний программ 6

Цели:

- защита интересов пользователей, государственных и ведомственных интересов на основе контроля качества ПО,
- подготовка и принятие решения о целесообразности выдачи серти-фиката соответствия

Виды:

- ***Обязательная сертификация*** - для ПС, выполняющих особо ответственные функции, в которых ошибки или отказы могут нанести большой ущерб или опасны для жизни и здоровья людей,
- ***Добровольная сертификация*** - для удостоверения качества ПС с целью повышения их конкурентоспособности, расширения сферы использования и получения дополнительных экономических преимуществ

Решение о выдаче сертификата на ПС основывается на оценке степени его соответствия специально разработанным ***документам:***

- международным и национальным стандартам на тестирование, испытания, аттестацию программ и баз данных,
- международным и государственным стандартам на технологию создания компонент ПС, языки программирования, их синтаксическим, семантическим и лексическим требованиям;
- стандартам на сопровождающую ПС документацию;
- нормативным документам - техническим условиям, техническим описаниям, эксплуатационным документам на ПС по выбору заказчика, разработчика и испытателя

- Заявитель для получения сертификата соответствия направляет в орган по сертификации заявку на проведение испытаний с указанием схемы проведения сертификации

Основным выходным документом является **Протокол испытаний**

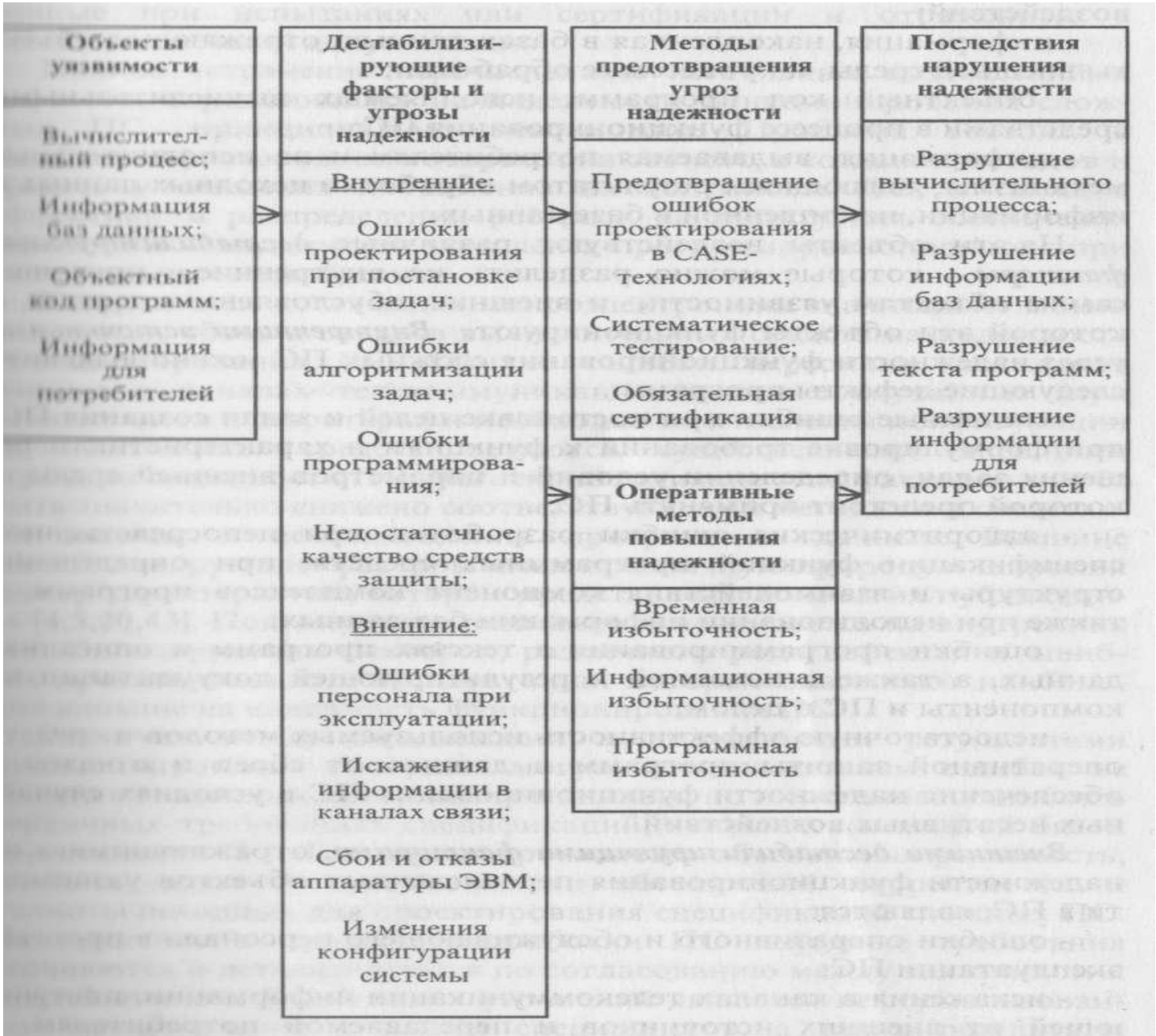
Протокол испытаний должен содержать информацию:

- служебную - регистрационный номер, дату, реквизиты утверждения, сведения о ПС;
- исходную - краткие сведения об испытанном объекте и заявителе, основание для проведения испытаний;
- выходную - краткие сведения о проведенных испытаниях, (в т.ч – рез-ты испытаний структурных компонент ПС с указанием реквизитов протоколов испытаний компонент;
- сведения о нормативных документах, на соответствие которым проверялось ПС;
- итоговую - краткие сведения о результатах испытаний, выводы и предложения.

Систематическое тестирование импортных ПС

- ***Для обеспечения надежности функционирования зарубежных ПС следует полностью отказаться от применения нелегальных импортных программ и баз данных.***
- В импортных программах кроме случайных ошибок возможны ***преднамеренные фрагменты - " закладные элементы" и вирусы***, с целью реализации вредных для эксплуатации функций, которые не описаны в документации.

Модель факторов, определяющих надежность ПС



Пример ИС на основе ПО повышенной надежности

1.Средства, использующие *временную избыточность*:

- авторизация доступа пользователей к системе;
- анализ доступных пользователю ресурсов; выделение ресурсов согласно ролям и уровням подготовки пользователей;
- разграничение прав доступа пользователей к отдельным задачам, функциям управления, записям и полям БД.

2. **Средства, использующие *информационную избыточность*:**

- открытая система кодирования, позволяющая пользователю в любой момент изменять коды любых объектов, обеспечивает стыковку системы классификации ИС с ПО других разработчиков;
- средства автоматического резервного копирования и восстановления данных;
- механизмы проверки значений контрольных сумм записей системы

3. Средства, использующие *программную избыточность*:

- распределение реализации одноименных функций по разным модулям ИС с использованием разных алгоритмов и системы накладываемых ограничений с возможностью сравнения полученных результатов;
- специальные алгоритмы пересчетов обеспечивают в ручном и автоматическом режимах переформирование групп документов, цепочек порождаемых документов;
- средства обнаружения и регистрации ошибок в сетевом и локальных протоколах;
- в программные модули системы встроены средства протоколирования процессов сложных расчетов с выдачей подробной диагностики ошибок;

Типы ошибок в ПО

Ошибки, скрытые в программе. При разработке сложного ПО возможно возникновение ошибок, которые не всегда удается обнаружить и ликвидировать в процессе отладки. В силу этого в программах остается некоторое количество скрытых ошибок, которые классифицируются как

1. Ошибки вычислений связаны с некорректной записью или программированием математических выражений, а также неверное преобразование типов переменных.

2. *Логические ошибки* являются причиной искажения алгоритма решения задачи. К ошибкам подобного рода можно отнести неверную передачу управления, неверное задание диапазона изменения параметра цикла, неверное условие и другие ошибки.

3.*Ошибки ввода-вывода* связаны с неправильным управлением ввода-вывода, формированием выходных записей, определением размера записей и другими неправильно свершенными действиями.

4. *Ошибки манипулирования данными.* К числу таких ошибок относятся: неверное определение числа элементов данных; неверные начальные значения, присвоенные данным; неверное указание длины операнда или имени переменной и другие ошибки.
5. *Ошибки совместимости* связаны с отсутствием совместимости разрабатываемого или применяемого ПО с ОС или другими прикладными программами.

- **Программисты-профессионалы тратят больше времени на изучение существующих программ, чем на создание новых**
- **Характеристики языка программирования существенно вляют на ошибки в ПО**

Пример. В ПО, управляющим первым полетом амер. на ВЕНЕРУ программист написал код (оператор DO)на Фортране:

DO 3 I=1.3

Точка вместо запятой.

В Фортране пробелы игнорируются, а переменные не обязательно объявляются явно.

Компилятор прочитал: новой переменной ***DO3I*** присвоить значение ***1,3***

Основные показатели надежности

1. **Вероятность безотказной работы, $P(T)$ – вероятность того, что время работы ПС (устройства) до отказа окажется больше заданного времени наработки t**

2. **Средняя наработка до отказа, T_0 – математическое ожидание наработки ПС (устройства) до первого отказа :**

$$T_0 = M = \int_0^\infty P(T)\, dT \qquad (1)$$

3. **Интенсивность отказов, $\lambda(T)$ – условная плотность вероятности возникновения отказа для рассматриваемого момента времени при условии, что до этого отказ не возникает :**

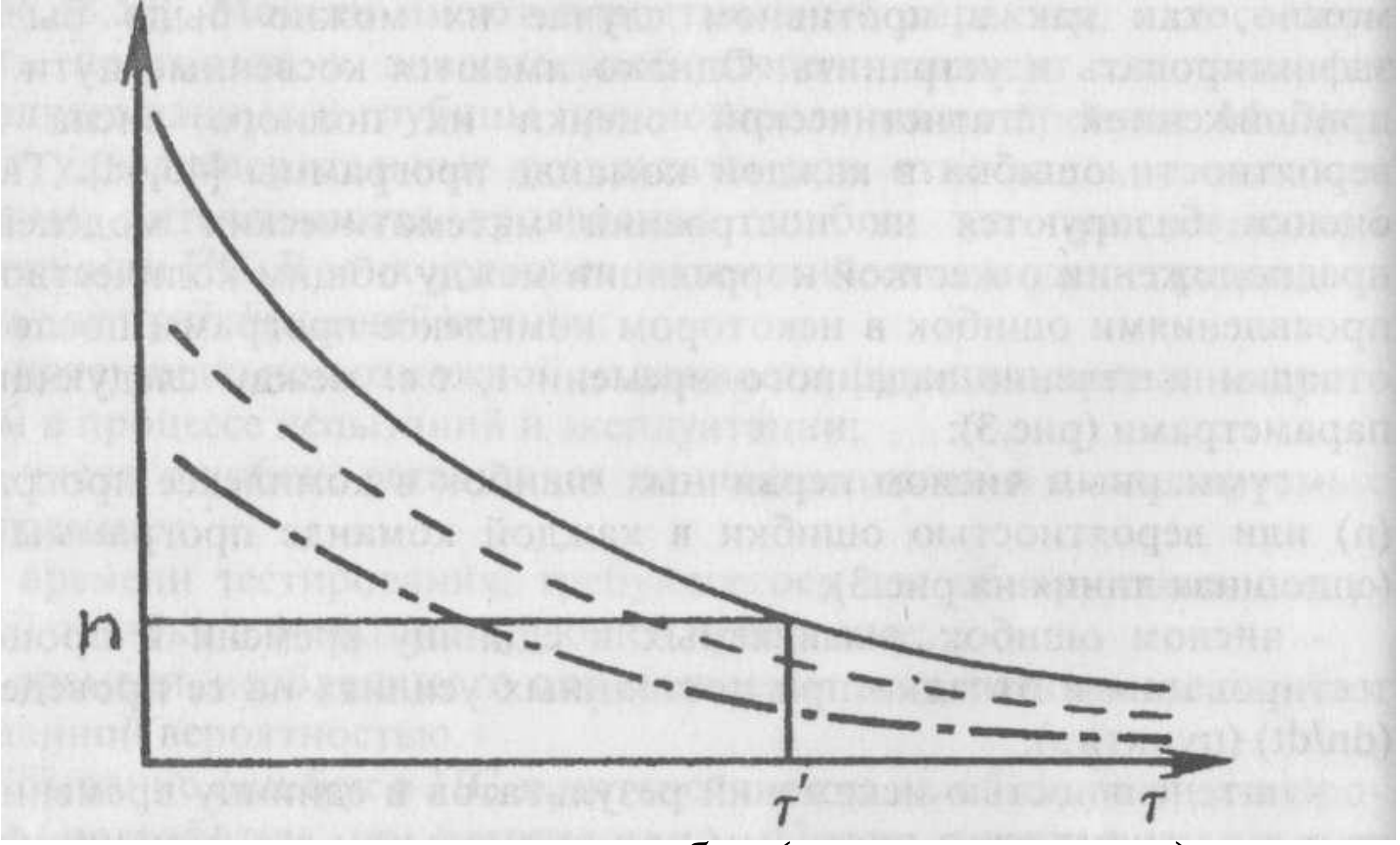
$$\lambda(T) = - [\, d\, P(T)/dT \,] / P(T) \qquad (2)$$

$$P(T) = \exp [\, - \int_0^t \lambda(T)\, d\, T \qquad (3)$$

Математические модели описания статистических характеристик ошибок в программах

Математические модели предназначены для приближенной оценки:

- потенциально возможной надежности функционирования ПС в процессе испытаний и эксплуатации;
- числа ошибок, оставшихся не выявленными в анализируемых программах;
- времени тестирования, требующегося для обнаружения ошибки в функционирующей программе;
- времени, необходимого для выявления всех имеющихся ошибок с заданной вероятностью.



Суммарное число первичных ошибок (сплошная кривая), число ошибок, выявляемых в единицу времени (пунктир) и интенсивность искажения результатов в единицу времени (интенсивность ошибок) (штрих- пунктир) в зависимости от времени отладки ПС

Интенсивность обнаружения ошибок снижается настолько, что разработчик ПС попадает в ***зону нечувствительности к ошибкам и отказам*** .

Экспоненциальная математическую модель распределения ошибок в программах

Используемые параметры:

- число первичных ошибок, **n,** (*вторичные* ош. – рез-т испр. первич)
- интенсивность обнаружения ошибок при отладке, **dn/dT**

Предположения:

- в начале отладки комплекса программ при **T=0** в нем содержалось **N** первичных ошибок,
- после отладки в течение **T** осталось n_0 первичных ошибок и устранено **n** ошибок ($n_0+n = N$),
- между значениями n_0 и **dn/dT** существует достаточно сильная корреляция (подтверждено экспериментально).

Модель дает удовлетворительные результаты при относительно высоких уровнях интенсивности проявления ошибок, т.е. при невысокой надежности ПС.

- Значение к-та **К** - изменение скорости проявления искажений при переходе от функционирования программ на специальных тестах к функционированию на типовых исходных данных

Интенсивность обнаружения ошибок (**dn/dT**)в программе и **абсолютное число устраненных первичных ошибок** (**n**) связываются уравнением:

$$dn/dT + K\,n = K\,N \qquad (5)$$

При **t = 0** отсутствуют обнаруженные ошибки, и решение уравнения (5) имеет вид (при экспоненциальном характере зависимсти выявленных ошибок от времени - рис. на сл.18):

$$n = N\,[\,1 - \exp(-K\,T)] \qquad (6)$$

Число оставшихся первичных ошибок в ПО:

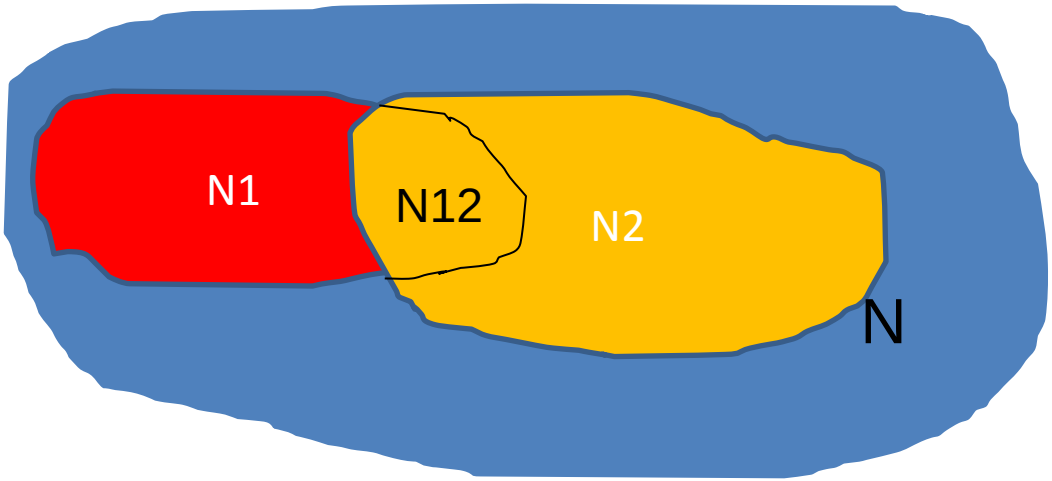
$$n_0 = N\ \exp(-K\,T)] \qquad (7)$$

Простая интуитивная модель

ПО тестируется 2 группами

Независимые наборы тестов

В теч некот врем группы работ параллельно, затем рез-ты сравнив



Эффективность обнаруж ошибок каждой из групп: **E1=N1/N, E2=N2/N**

Предположение: **E1=N1/N = N12/N2 = N12/ (N*E2)**

или **N= N12/ (E1*E2)**

Пример. N1=20, N2=30, N12=8

Имеем: E1=0.27, E2=0.4 в предположении, что **E1=N12/N2, E2=N12/N1**

Тогда N= N12/ (E1*E2)= 8/(0.27*0.4)=74, те. Не обнар ? ошибок

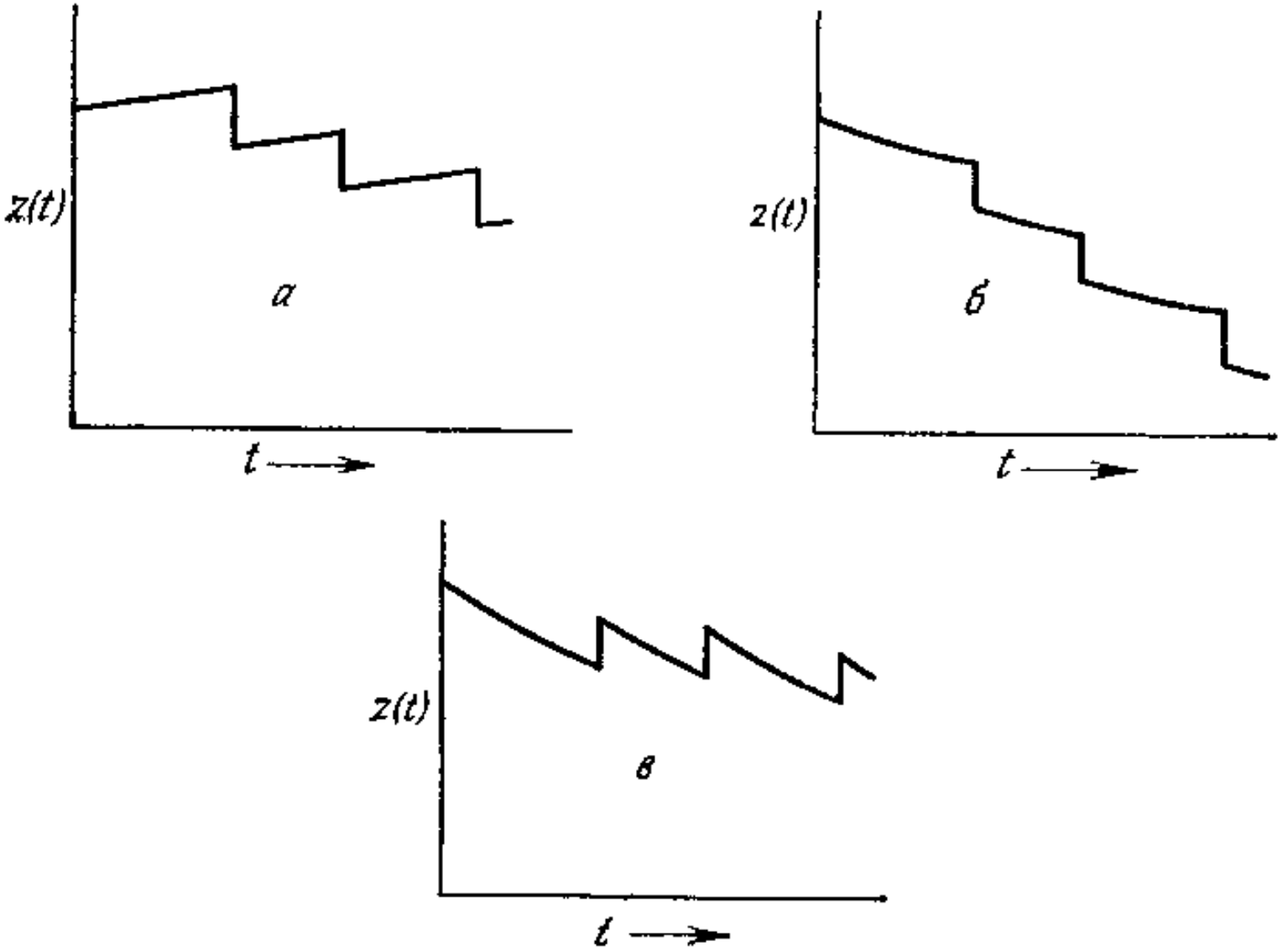
- **Из всех неизвестных параметров надежности программного обеспечения, самым важным является число ошибок, оставшихся в программе (n_0).**
- **Если знать n_0, то можно оценить стоимость работ по сопровождению ПО и уровень доверия к нему**
- **Все ошибки одинаково серьезны** (отказ системы и орфографическая ошибка в сообщении одинаково важны).
- Предполагают, что в ПО имеется **1ош/100 операт**

- При моделировании иногда вводят **функцию риска** ***z(T)*** - условную вероятность того, что ошибка проявится на интервале от *T* до T+ ΔT, при условии, что до момента *T* ошибок не было.

Если *T* — время появления ошибки, то

$$z(t)\Delta t = P\{t < T < t + \Delta t \mid T > t\},$$

Некоторые функции риска



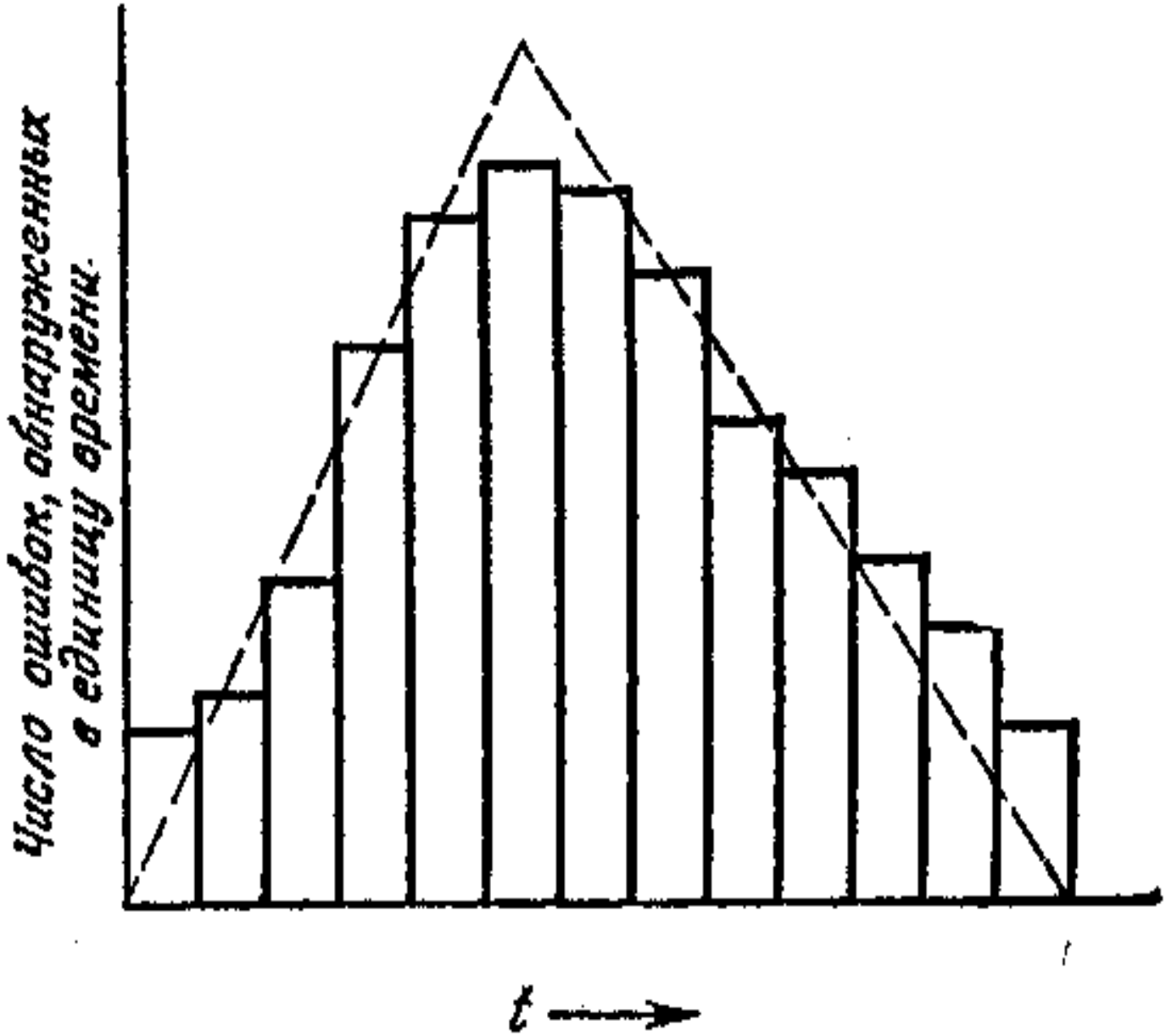
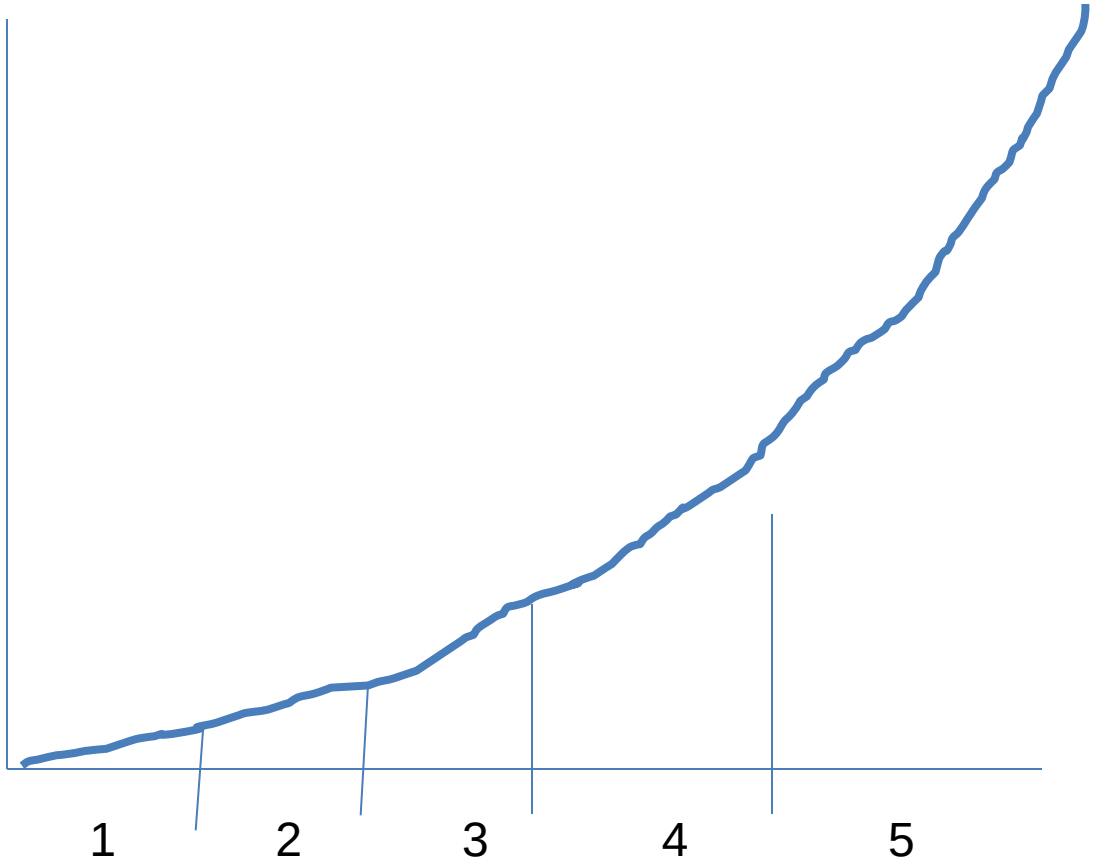


Рис. 18.3. Приближение треугольником.



Y – вер-ть новой ошибки при исправлении, стоимость исправл-я ош

X : 1 – **контроль проекта**, 2 – **автономное тестирование**, 3 – **тестирование функций** – поиск расхождений между ПО и внешними спецификациями ; функциональные требования определяют, что именно делает ПО, какие задачи оно решает , 4 – **комплексное тестир-е** (комплекс разл тестов) – поиск несоответствия системы исходным целям, 5 – **тестирование приемлемости** и **использование**

Альфа-тестирование — имитация реальной работы с системой штатными разработчиками, либо реальная работа с системой потенциальных пользователей.

Проводится на ранней стадии разработки продукта. Обнаруженные ошибки могут быть переданы тестировщикам для дополнительного исследования в окружении, подобном тому, в котором будет использоваться ПО.

Бета-тестирование —выполняется распространение версии с ограничениями (по функциональности или времени работы) для некоторой группы лиц, с тем чтобы убедиться, что продукт содержит достаточно мало ошибок. Иногда бета-тестирование выполняется для того, чтобы получить обратную связь о продукте от его будущих пользователей.