# 1. Надежность ЭВМ

## 1.1. Основные понятия и определения

В основе всего лежит понятие качества изделия системы.
**Качество** — совокупность свойств продукции, которая позволяет оценить пригодность продукции удовлетворять определенным потребностям.


**Надежность** — особое свойство качества изделия, которое позволяет определить стабильность всех других свойств качества изделия во времени.
**Свойство** — объективная особенность изделия, которое проявляется при его создании, эксплуатации и потреблении.
**Надежность изделия** — свойство изделия выполнять заданные функции, сохраняя свои эксплуатационные показатели в течение требуемого промежутка времени или требуемой наработки.
**Исправное состояние** — состояние, при котором изделие выполняет все заданные функции с параметрами установленными для него в технической документации.
**Работоспособность** — состояние, при котором изделие выполняет все заданные функции, но с некоторыми пониженными показателями качества.
**Отказ** — событие, которое заключается в нарушении работоспособности. Отказы бывают полные, когда отказывает вся система и частичные, в этом случае отказывает часть системы. Так же отказы делятся:

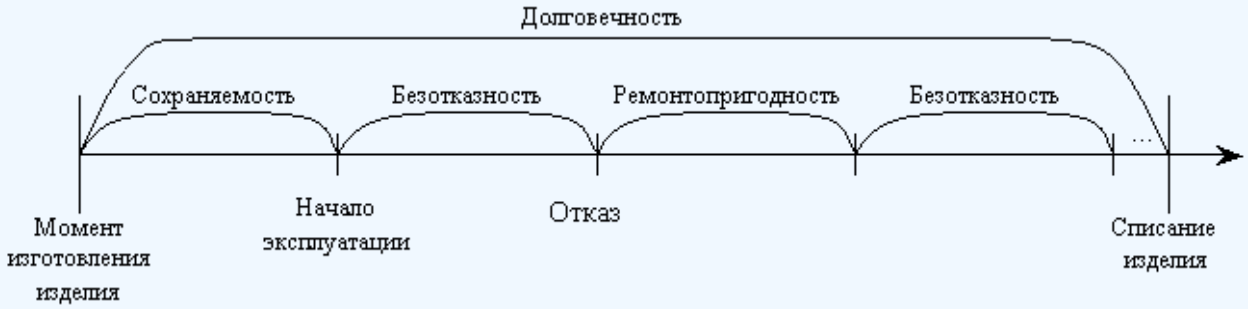
1. По характеру возникновения:
    - Внезапные (катастрофические). Внезапные отказы возникают в результате резкого (скачкообразного) изменения выходных показателей системы.
    - Постепенные (параметрические). Постепенные возникают при постепенном снижении выходных параметров изделия во времени и когда эти параметры пересекают критическое значение, считается, что отказ произошел.
2. По характеру обнаружения (выявления):
    - Явные (очевидные). Явные отказы системы обнаруживаются при внешнем осмотре или включении системы.
    - Скрытые (неочевидные). Скрытые обнаруживаются инструментальными средствами.
3. По влиянию отказа других элементов:
    - Зависимые (вторичные отказы). Возникают под влиянием отказов других элементов.
    - Независимые (первичные отказы).
4. По времени существования:
    - Устойчивые (окончательные).
    - Перемежающие отказы (самопроизвольно возникают и самопроизвольно устраняются).
5. По влиянию на ремонтопригодность:
    - Неисправности (устраняются путем мелкого ремонта).
    - Аварии (большие ремонтные работы).

6. По природе существования:
    - Физические.
    - Функциональные (не выполнение каких-то функций).

Надежность это комплексное свойство изделия, в которое входят:

1. Безотказность.
2. Долговечность.
3. Ремонтопригодность.
4. Сохраняемость.

**Безотказность**— свойство изделия сохранять работоспособность в течение заданной наработки без перерывов.
**Долговечность** — свойство изделия сохранять работоспособность до предельного состояния (до списания) с перерывами на техническое обслуживание и ремонт.
**Ремонтопригодность** — свойство изделия обнаруживать, устранять и предупреждать неисправность и отказы путем проведения технического обслуживания и ремонта.
**Сохраняемость** — свойство изделия сохранять свои эксплуатационные показатели в течение и после срока транспортирования или хранения его на складе.




Все системы подразделяются на восстанавливаемые и невосстанавливаемые. Невосстанавливаемые системы эксплуатируются до первого отказа.
У восстанавливаемых систем может быть поток отказов.
Ремонтируемые и неремонтируемые. Технические термины, говорящие о возможности ремонта системы. Но ремонт может быть дорогой или в условиях эксплуатации не возможен.
Восстанавливаемые и невосстанавливаемые системы для расчетов.

**Задачи обеспечения надежности ЭВМ**

Этапы существования ЭВМ:

1. Проектирование. Схемно-конструктивные методы получения надежности
    - Выбор и обоснование показателя эффективности ЭВМ и определение его взаимосвязи с показателем надежности.
    - Нормирование надежности. Определение оптимального уровня показателя надежности системы, которой она должна обладать во время эксплуатации системы.
    - Расчет показателя надежности всей системы, если известны показатели надежности всех элементов.
    - Решение задачи оптимального резервирования (дублирования отдельных элементов).

2. Изготовление. Производственные методы повышения надежности
    - Степень автоматизации на высшем уровне.
    - Методы для статистического регулирования надежности.
    - Тренировка элементов и систем (испытание сложной системы в течение небольшого промежутка времени с тем, чтобы выявить производственные дефекты).
3. Эксплуатация. Эксплуатационные методы
    - Использование диагностических систем, которые выявляют скрытые дефекты.
    - Прогнозирование отказов системы.
    - Применение гибкой системы технического обслуживания и ремонта (ремонт производится в зависимости от состояния системы).

## 1.2. Показатели надежности

### 1.2.1. Система показателей надежности

Количественные показатели надежности определяются как характеристики случайных величин

**Безотказность** - непрерывное время безотказной работы системы определяется как наработка на отказ T’

**Долговечность** - время от момента изготовления системы до предельного состояния или списания. срок службы T’’

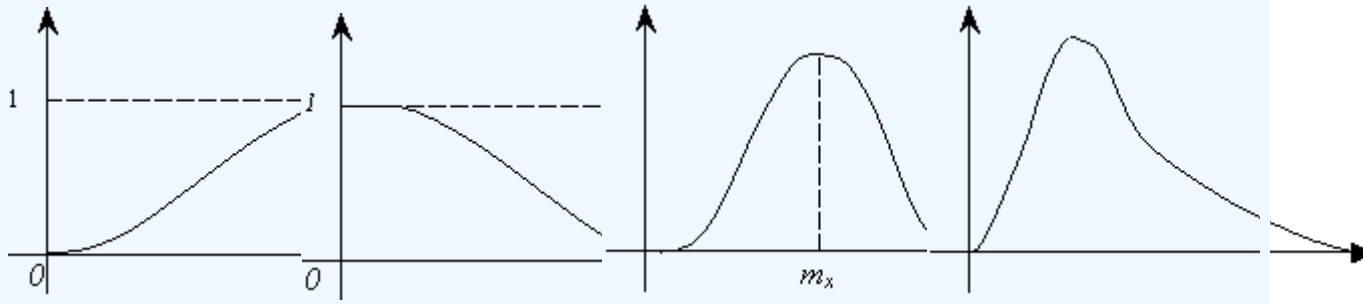
**Ремонтнопригодность** - время восстанавливаемости T’’’

**Сохраняемость** - случайное время сохраняемости T’’’’

**Функция распределения** T = F(t) — вероятность того, что случайная величина примет значение меньше чем *t* заданное:

**P { T < t }**

$F(t) = \int_0^t f(x)\,dx$ F(t) = 1 – F(t) $f(t) = \frac{dF(t)}{dt}$ $\beta(t) = \frac{f(t)}{\overline{F}(t)}$




Найдем зависимость между $\beta(t)$ и $\overline{F}(t)$, если известна $\beta(t)$.

$$\beta(t) = \frac{f(t)}{\overline{F}(t)}$$

$$f(t) = \frac{dF(t)}{dt} = \frac{d[1-\overline{F}(t)]}{dt} = -\frac{d\overline{F}(t)}{dt}$$

подставляем:

$$\beta(t) = \frac{f(t)}{\overline{F}(t)} = \frac{-\frac{d\overline{F}(t)}{dt}}{\overline{F}(t)} = -\frac{d\overline{F}(t)}{\overline{F}(t)\,dt}$$

$$\ln\ \overline{F}(t) = -\int\frac{d\overline{F}(t)}{\overline{F}(t)}$$

$$\int\beta(t)\,dt = -\int\frac{d\overline{F}(t)}{\overline{F}(t)}$$

$$\overline{F}(t) = e^{-\int_0^t \beta(x)\,dx}$$

### 1.2.2. Показатели безотказности невосстанавливаемых систем

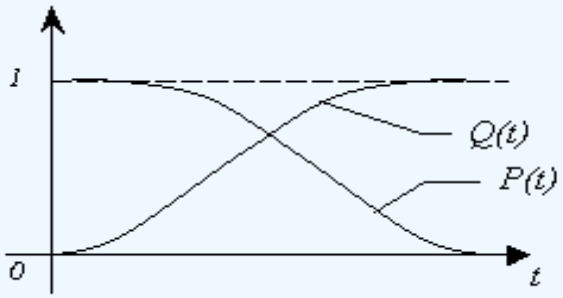
Случайной величиной является наработка на отказ. Закон распределения для безотказной работы $F(t) = Q(t)$ — вероятность отказа.

$P(t)$ — вероятность безотказной работы

$P(t) = 1 – Q(t)$

$f_H(t)$ — плотность распределения наработки на отказ

$$f_H(t) = \frac{dQ(t)}{dt} = -\frac{dP(t)}{dt}$$



$$Q(t) = \int_0^t f_H(x)\,dx$$

$$P(t) = 1 - \int_0^t f_H(x)\,dx = \int_t^\infty f_H(x)\,dx$$

$\lambda(t)$ — интенсивность отказов

$$\lambda(t) = \frac{f_H(t)}{P(t)}$$

$$P(t) = \exp[-\int_0^t \lambda(x)\,dx]$$

$$f_H(t) = \lambda(t)P(t) = \lambda(t)\exp[-\int_0^t \lambda(x)\,dx]$$

Время наработки на отказ:

$$\tau_0 = \int_0^\infty t \cdot f_H(t)\,dt = -\int_0^\infty t \cdot \frac{dP(t)}{dt}\,dt = -\int_0^\infty t\,dP(t) = -t \cdot P(t)\Big|_0^\infty + \int_0^\infty P(t)\,dt = \int_0^\infty P(t)\,dt$$

Свойства безотказной работы *P(t):*

*1. t=0, P(t)=1*

2. *P(t)* — монотонно убывающая функция во времени

3. t > 0, P(t) > 0

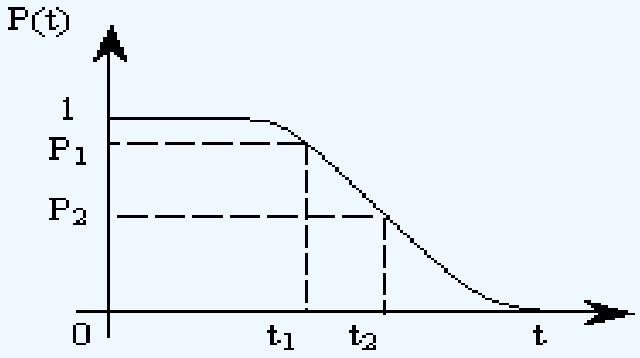
Таблица взаимосвязей показателей безотказности невосстанавливаемых систем

| Определяемый показатель | Заданный показатель | | | |
|---|---|---|---|---|
| | Q(t) | P(t) | $f_n(t)$ | (t) |
| Вероятность отказа Q(t) | — | 1 – P(t) | $\int_0^t f_n(x)\,dx$ | $\int_0^t \lambda_n(x)\,dx$ |
| Безотказная работа P(t) | 1 – Q(t) | — | $1 - \int_0^t f_n(x)\,dx = \int_t^\infty f_n(x)\,dx$ | $e^{-\int_0^t \lambda_n(x)\,dx}$ |

| | | | | |
|---|---|---|---|---|
| Плотность распределения $f_n(t)$ | $\frac{dQ(t)}{dt}$ | $-\frac{dP(t)}{dt}$ | — | $\lambda(t)e^{-\int_0^t \lambda_n(x)\,dx}$ |
| Интенсивность $\lambda$ (t) | $\frac{dQ(t)/dt}{1-Q(t)}$ | $\frac{-dP(t)/dt}{P(t)}$ | $\frac{f_n(t)}{\int_t^\infty f_n(x)\,dx}$ | — |

Определение показателя безотказности на интервале времени $\tau$ при условии, что система безотказно проработала от 0 до $t_1$.

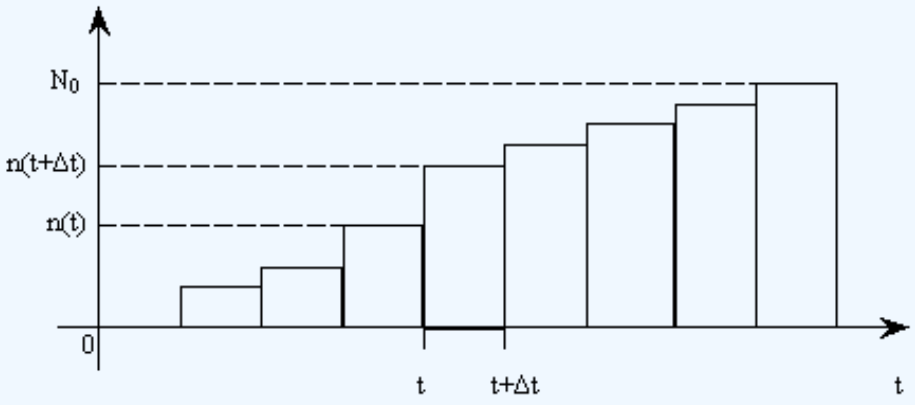
$$P(\tau/t_1) = \frac{P(t_2)}{P(t_1)} = \frac{P(t_1+\tau)}{P(t_1)}$$




$$Q(\tau/t_1) = 1 - P(\tau/t_1) = 1 - \frac{P(t_1+\tau)}{P(t_1)} = \frac{P(t_1) - P(t_1+\tau)}{P(t_1)}$$

$$\lambda(\tau/t_1) = \frac{f_H(\tau/t_1)}{P(\tau/t_1)} = \frac{\frac{d[1-P(\tau/t_1)]}{d\tau}}{\frac{P(t_1+\tau)}{P(t_1)}} = \frac{\frac{d[P(t_1+\tau)]}{P(t_1)d\tau}}{\frac{P(t_1+\tau)}{P(t_1)}} = \frac{d[P(t_1+\tau)]}{P(t_1+\tau)}$$

$$T_0(t_1) = \int_0^\infty P(\tau/t_1)\,d\tau = \int_0^\infty \frac{P(t_1+\tau)}{P(t_1)}\,d\tau = \frac{1}{P(t_1)}\int_{t_1}^\infty P(\tau)\,d\tau$$




$N_o$ — количество изделий.

$n(t)$ — количество отказавших изделий за время $t$.

Вероятность отказа:

$$\hat{Q}(t) = \frac{n(t)}{N_0}$$

Безотказная работа:

$$\hat{P}(t) = 1 - \frac{n(t)}{N_0}$$

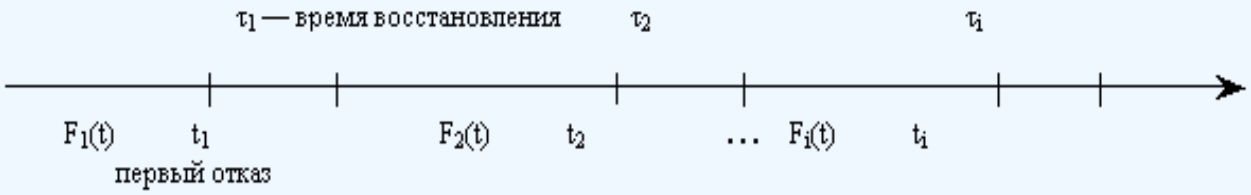
Плотность распределения:

$$\hat{f}_H(t) = \frac{n(t + \Delta t) - n(t)}{N_0 \cdot \Delta t} = \frac{dQ(t)}{dt} = -\frac{dP(t)}{dt}$$

Интенсивность отказов: $\hat{\lambda}_H(t) = \frac{\hat{f}_H(t)}{\hat{P}(t)} = \frac{n(t + \Delta t) - n(t)}{\overline{N_0} \cdot \Delta t}$, где $\overline{N} = N_0 - n(t)$

$\hat{T}_0 = \frac{\sum_{i=1}^{N_0} t_i}{N_0}$, где $t_i$ — наработка на отказ i-ой системы.

### 1.2.3. Показатели безотказности восстанавливаемых систем




$t_i$ — i-ая наработка на отказ.

$\omega(t) = \frac{d\Omega(t)}{dt}$ $\tau_i$ — i-ое восстановление после i-ого отказа

$F_1(t)=Q_1(t)$ — вероятность первого отказа.

Практическое применение нашли показатели первого отказа.

$\Omega(t)=M[N(t)]$ — ведущая функция потока отказов, математическое ожидание случайного числа отказов от 0 до $t$.

**Параметр потока отказов** — среднее значение количества отказов в единицу времени за

рассматриваемый интервал времени.

Среднее значение отказов за рассматриваемую наработку $T_P$:

$$\omega_{T_p} = \frac{d\Omega(T_p)}{T_p}, \quad \Omega(t) = \int_0^t \omega(x)\,dx \quad \Rightarrow \quad \omega_{T_p} = \frac{1}{T_p}\int_0^{T_p} \omega(x)\,dx$$

Средняя наработка между отказами восстанавливаемого изделия:

$$T = \frac{1}{\omega_{CP}} = \frac{T_p}{\Omega(T_p)} = \frac{T_p}{\int_0^{T_p} \omega(x)\,dx}$$

$$\lim_{t\to\infty} \omega(t) = \omega_{cp} = \frac{1}{T}$$

$$f(t) \le \omega(t) \le \lambda(t)$$

$$F_1(t) = F_2(t) = \ldots = F_i(t)$$

$$\omega(t) = f(t) + \int \omega(t-\tau) f(\tau) d\tau$$

$$\omega^*(S) = \frac{f^*(S)}{1 - f^*(S)}$$

Упрощаем:

$$f^*(S) = \int_t^{\infty} e^{-St} f(t) dt$$

Если функция распределения наработки между отказами подчиняется экспоненциальному закону распределения, то расчет ещё более упрощается:

$$\omega = \omega_{cp} = \frac{1}{T} = \lambda$$

### 1.2.4. Показатели сохраняемости

Свойства сохраняемости описывают надежность систем, которые хранятся на складе или транспортируются.

**Средний срок сохраняемости** — математическое ожидание случайной величины хранения до отказа.

$$T_{хр} = \int_0^{\infty} t f_{хр}(t) dt$$

**Процентный срок сохраняемости** — срок сохраняемости, который достигается объектом с вероятностью $\gamma$ выраженной в %.

$$T_{хр}\,\gamma > \int_{T_{хр}}^{\infty} f_{хр}(t) dt = \frac{\gamma}{100}$$

для вероятности в 50 %

$$\int_{T_{хр}}^{\infty} f_{хр}(t) dt = 0{,}5$$

Функция распределения случайного времени восстановления системы.

$$\mu(t) = \frac{f_в(t)}{1 - F_B(t)} = \frac{\frac{dF_B(t)}{dt}}{1 - F_B(t)} = \frac{f_B(t)}{1 - \int_0^{\infty} f_B(x) dx}$$

Среднее время восстановления:

$$T_B = \int_0^{\infty} t f_B(t) dt$$

Вероятность восстановления:

$$P_B(t) = F_B(t) = \int_0^{\infty} f_B(t) dt = \int_0^{\infty} [1 - P(t)] dt$$

### 1.2.5. Показатели ремонтопригодности

**Ресурс** — наработка системы до списания выраженная во временных единицах измерения без учета простоев.

Средний ресурс:

$$T_P = \int_0^{\infty} t f_p(t) dt$$

**Срок службы** — календарная продолжительность службы объекта до его списания.

Средний срок службы:

$$T_{СЛ} = \int_0^\infty t f_{сл}(t)dt$$

Средний ресурс с вероятностью $\gamma$ выраженной в %

$$T_р\ \gamma > \int_{T_{р\gamma}}^\infty f_р(t)dt = \frac{\gamma}{100}$$

Средний срок службы с вероятностью $\gamma$ выраженной в %

$$T_{сл}\ \gamma > \int_{T_{СЛ\gamma}}^\infty f_{сл}(t)dt = \frac{\gamma}{100}$$

Медиана ресурса и срока службы

$$\int_{T_{0\gamma}}^\infty f_р(t)dt = 0{,}5 \qquad \int_{T_{СЛ\gamma}}^\infty f_{сл}(t)dt = 0{,}5$$

### 1.2.6. Комплексные показатели надежности

Комплексные показатели характеризуют сразу несколько свойств надежности.

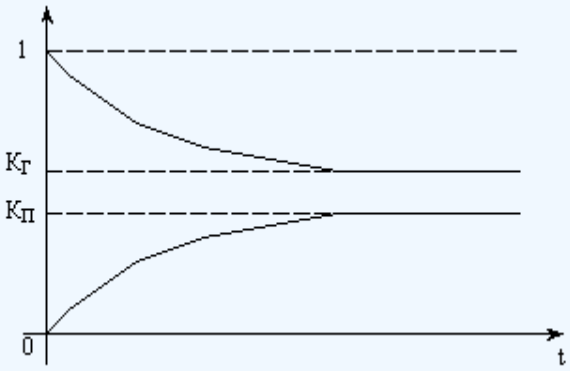
Безотказность и ремонтопригодность определяются:

- функции готовности и простоя;
- коэффициенты готовности и простоя;
- коэффициент технического использования;
- коэффициент оперативной готовности.

**Функция готовности** — вероятность того, что в любой произвольный момент времени система будет находиться в работоспособном состоянии.

**Функция простоя** — вероятность того, что в произвольный момент времени система будет находиться в неработоспособном состоянии.

$$\lim_{t\to\infty} K_Г(t) = K_Г$$




$$\lim_{t\to\infty} K_П(t) = K_П$$

Коэффициент готовности можно узнать из статистических данных:

$t_i$— i-ая наработка на отказ.

$t_i$— время восстановления после i-го отказа.

$$K_Г = \frac{\sum_{i=1}^{n} t_i}{\sum_{i=1}^{n} t_i - \sum_{i=1}^{n} \tau_i}$$

$$K_Г = \frac{\frac{\sum_{i=1}^{n} t_i}{n}}{\frac{\sum_{i=1}^{n} t_i}{n} - \frac{\sum_{i=1}^{n} \tau_i}{n}} = \frac{T_O}{T_O + T_B} \quad , \quad n \to \infty \quad K_Г = \frac{T_O}{T_O + T_B}$$

$$\boldsymbol{K_П = 1 - K_Г}$$

**Коэффициент готовности** — вероятность в произвольной момент времени застать систему в работоспособном состоянии, кроме периодов, предусмотренных на плановое техническое обслуживание и ремонт.

**Коэффициент технического использования** — отклонение математического ожидания нахождения системы в работоспособном состоянии к сумме математических ожиданий нахождения системы в работоспособном состоянии с учетом плановых и неплановых перерывов.

$$K_{ТП} = \frac{\sum_{i=1}^{n} t_i}{\sum_{i=1}^{n} t_i + \sum_{j=1}^{m} \tau_{П\,j} + \sum_{l=1}^{k} \tau_{Н\,l}}$$

$t_{п\,i}$ — время планового простоя

$t_{н\,i}$ — время непланового простоя

Коэффициент оперативной готовности

$$\boldsymbol{K_{ОГ} = K_Г * P(t_{ОГ})}$$

$K_{ОГ}$ — вероятность выполнения ожидаемой задачи.

### 1.3. Показатели безопасности

#### 1.3.1. Понятия и определения

Одним из важнейших свойств качества оборудования является безопасность его работы. Поэтому для систем управления необходимы расчеты на безопасность.

> **Безопасность —** особое свойство, которое характеризует степень безаварийности и безопасности функционирования системы. Учитываются свойства безопасности, когда имеются потенциально опасные объекты. В практике часто путают определения надежности и безопасности. Рассмотрим подход к определению безопасности.

Потенциально-опасные объекты могут находиться в одном из восьми состояний в отношении безопасности. Все состояния подразделяются на три группы:

1. Состояние, которое не может привести к несчастному случаю.

2. Состояния, которые могут привести к несчастному случаю.

3. Состояния, соответствующие несчастным случаям.

Номера состояний обозначают следующие ситуации:

0 – Безошибочное функционирование.

1 – Возникновение нарушений или отказов отдельных элементов.

2 – Нарушение функций в результате наступления события 1.

3 – Недопустимое нарушение функций.

4 – Выдача недопустимой величины управляющего сигнала.

5 – Возникновение недопустимых значений параметров в процессе управления.

6 – Возникновение предаварийного состояния, появление возможности несчастного случая.

7 – Аварийное состояние, несчастный случай.

Как видно, в число состояний наряду с нарушениями и отказами технических средств входит и нарушение функций систем управления (состояние 3), т.е. функциональный отказ. Дадим определение безопасности. ***Безопасность*** есть свойство системы выполнять требуемые функции: при этом исключается возможность для заданных условий и в заданном интервале времени недопустимых состояний, приводящих к авариям и несчастным случаям.

Безопасность подразделяется на пять видов согласно пяти недопустимым состояниям:

1) недопустимость нарушения функций (состояние 3);

2) недопустимость выдачи управляющего сигнала, превышающего предельные уровни (состояние 4);

3) недопустимость возникновения величин параметров системы, превышающих предельные уровни (состояние 5);

4) недопустимость возникновения предаварийных состояний и возможностей несчастного случая (состояние 6);

5) недопустимость аварийных состояний и несчастных случаев (состояние 7).

Состояния под номерами 1 и 2 можно количественно описать соответствующими показателями надежности. Остальные состояния (с 3 по 7) количественно определяются функциями риска.

Функция риска есть вероятность того, что случайное время $T_i$ работы системы от начала включения в работу до первого возникновения одного из недопустимых состояний $i$ $(i = \overline{3,7})$ меньше заданного времени $t$:

$$R_i(t) = P(T_i < t), \quad i = \overline{3,7}$$

Отсюда получаем функцию безопасности:

$$S_i(t) = 1 - R_i(t) = P(T_i \geq t), \; i = \overline{3,7} \quad (1.1)$$

Обозначим $M_i$ как математическое ожидание случайной величины времени от начала включения элемента в работу до первого возникновения одного из недопустимых состояний $i$ $(i = \overline{3,7})$, т.е.

$$M_i = \int_0^\infty [1 - R_i(t)]dt = \int_0^\infty S_i(t)dt \quad (1.2)$$

Таким образом получаем ряд показателей $M_3$, $M_4$, $M_5$, $M_6$, $M_7$ наглядно определяющих уровень безопасности для каждого из недопустимых состояний. Выделим первые два из них $M_3$ и $M_4$, так как они характеризуют появление недопустимых отклонений в функционировании системы. Эти отклонения прямо не ведут к авариям и несчастным случаям, они свидетельствуют об опасности их возникновения через вторичные недопустимые отклонения и через отказы системы управления.

***Рекомендуется применять следующие показатели:***

1. Средняя наработка до опасного нарушения функций:

$$MF = M_3$$

2. Средняя наработка до выдачи опасной величины управляющего сигнала:

$$MS = M_4$$

3. Фактор безопасности $K_S$ есть отношение средней наработки до выдачи опасной величины управляющего сигнала $MS$ к средней наработке на отказ системы управления $T_0$:

$$K_S = MS / T_0$$

4. Выигрыш безопасности $G_S$ есть отношение средней наработки до выдачи опасной величины управляющего сигнала в условиях применения определенного мероприятия безопасности $MS$ и в случае неприменения этого мероприятия безопасности $MS_0$:

$$G_S = MS / MS_0$$

Общая схема повышения безопасности объектов содержит два этапа:

- анализ риска;

- управление риском.

Первый этап состоит в определении характеристики рисков, определении их приоритетов, получении обобщенной оценки риска, формировании мероприятий по снижению степени риска.

На основе полученной оценки степени риска будет выбираться соответствующая стратегия управления риском, в результате которой производится оптимальный синтез системы управления конкретным объектом или группой объектов с минимизацией целевой функции или критериев эффективности при заданных ограничениях на отведенные ресурсы и время.

***Стратегия управления риском предусматривает:***

- определение уровня приемлемого риска;

- разработку системы организационных мер и средств оперативно-технического управления, обеспечивающих минимизацию или полное исключение риска;

- разработку системы контроля для оценки негативных последствий воздействия производственных процессов на экосистему, биосферу и человека с учетом факторов времени;

- разработку системы поддержки принятия решений для выбора наиболее рационального способа снижения риска до приемлемого уровня в кратчайшее время.

### 1.4. Методы расчета показателей надежности сложных систем

Методы расчета надежности сложных систем применяются на первом этапе при проектировании. Они относятся к схемно-конструкторским методам. Суть этих методов сводится к тому, что при известных показателях надежности отдельных элементов необходимо рассчитать надежность системы. Показатели надежности сложных систем можно количественно оценивать, используя информацию о надежности отдельных элементов сложных систем. Для этого необходимо знать показатели элементов систем управления и математическую модель соединения этих элементов в систему. При этом не следует смешивать понятие “соединение элементов” сложных систем на технологических, принципиальных и других схемах с понятием “соединение элементов” в виде математических моделей для решения задач надежности. В общем случае эти схемы соединений не совпадают. К наиболее перспективным методам расчета надежности систем управления можно отнести следующие:

- классические;
- топологический;
- логико-вероятностный;
- структурный.

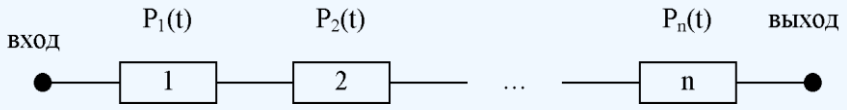
### 1.4.1. Классический метод расчета надежности

К классическим методам относятся модели надежности с последовательным, параллельным, параллельно-последовательным соединением элементов, их различные модификации.

1**. Модель с последовательным соединением элементов**

При расчетах надежности последовательным называется такое соединение элементов, при котором отказ хотя бы одного из них приводит к отказу всего соединения в целом. Последовательное соединение в указанном выше смысле не всегда совпадает с физическим последовательным соединением элементов. Отказы элементов предполагаются независимыми, то есть отказ любой группы элементов никак не повлияет на вероятностные характеристики остальных элементов. Элемент понимается в широком смысле слова - это один из самостоятельных участков последовательного соединения.

1.




**В данном случае вероятность безотказной работы системы можно рассчитать по формуле:**

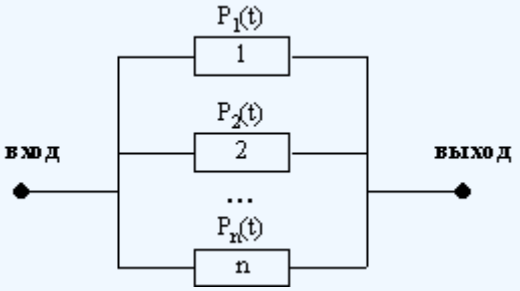
$$P_C = \prod_{i=1}^{n} P_i(t)$$

где $P_c$ - вероятность безотказной работы системы,

$P_i(t)$ - вероятность безотказной работы i-го элемента системы

**2. Модель с параллельным соединением элементов**

При расчетах надежности параллельным называется такое соединение элементов, при котором отказ всего соединения в целом происходит при отказе всех элементов системы (элементы дублируют друг друга).

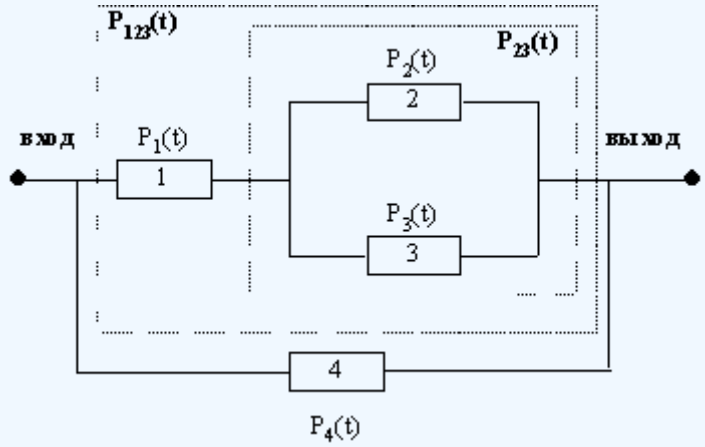



$$\left.\begin{array}{l} q_1(t) = 1 - P_1(t) \\ q_n(t) = 1 - P_n(t) \end{array}\right\} \Rightarrow Qc(t) = \prod_{i=1}^{n} q_i(t)$$

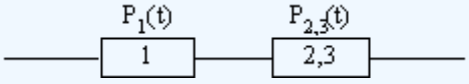
$$P_c(t) = 1 - Q_c(t) = 1 - \prod_{i=1}^{n} q_i(t) = 1 - \prod_{i=1}^{n} (1 - p_i(t))$$

**3. Модель с параллельно-последовательным соединением элементов**

1.

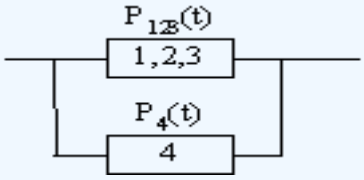



$$P_{23}(t) = 1 - (1 - P_2(t))(1 - P_3(t))$$

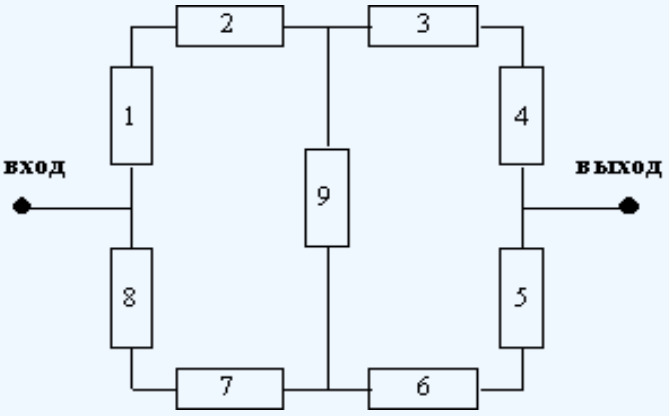



$$P_{123}(t) = P_1 * \overbrace{(1 - (1 - P_2(t))(1 - P_3(t)))}^{P_{23}}$$

$$P_c(t) = 1 - (1 - P_4(t)) * (1 - P_1(t) * (1 - (1 - P_2(t)) * (1 - P_3(t))))$$




**4. Модели несводимые к параллельно - последовательным**




Система является работоспособной, если работоспособны:

- **1,2,3,4,5,6,7,8,9;**
- **1,2,9,3,4;**
- **…….**
- **8,7,9,6,5;**
- **8,7,9,3,5;**

Работоспособность i-го элемента - $\mathbf{X_i}$ представляет собой функцию:

$$X_i = \begin{cases} 1, \text{если элемент работоспособен} \\ 0, \text{если элемент не работоспособен} \end{cases}$$

Таким образом, **функция работоспособности** системы представляет собой:

$$X_1 - X_2 - X_3 - X_4 - X_5 - X_6 - X_7 - X_8 - X_9 \vee X_1 - X_2 - X_9 - X_3 - X_4 \vee$$
$$\vee \ldots \vee X_8 - X_7 - X_9 - X_3 - X_5 \vee X_8 - X_7 - X_9 - X_3 - X_5$$

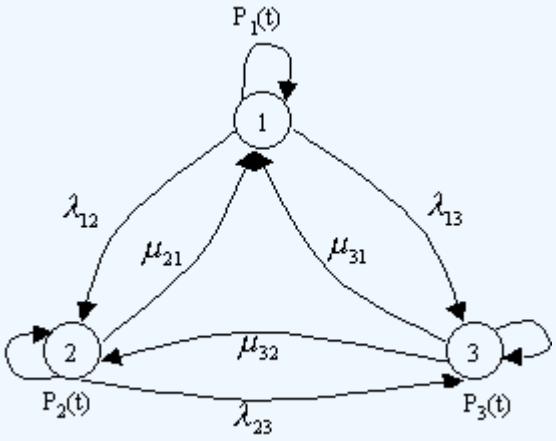
**Функция работоспособности системы** приводится к ортогональной форме. Затем производится замена на математические операции.

**Модель с использованием марковских процессов**

Модель задается в виде состояний, в которых система может находиться, и возможных переходов из одного состояния в другое.

$$P_c(t) = P_1(t)$$

**Допущение: нахождение состояния системы в данный момент времени не зависит от предыдущего состояния**




где: $\lambda$ - показатель интенсивности отказа,

$\mu$ - показатель ремонтопригодности

На графике изображены следующие состояния:

1. Работают оба элемента системы
2. Отказ одного из элементов - ИС работоспособна
3. Отказ второго элемента - ИС неработоспособна

При представлении ВС с помощью данной модели используется теория марковских процессов, в том случае, если нахождение системы не зависит от того, в каком состоянии находилась ВС в прошлом.

Составляется уравнение Колмогорова - Смирнова:

Производная от вероятности нахождения системы в i - том состоянии равна алгебраической сумме произведений интенсивности перехода на вероятности соответствующих состояний. Тем произведениям, которым соответствуют уходящие из данного состояния стрелки, приписывают знак "-", а входящим - "+".

Таким образом, для системы, изображенной на рисунке, имеем:

$$\begin{cases} \frac{dP_1(t)}{dt} = -(\lambda_{12} + \lambda_{13})P_1(t) + \mu_{21}P_2(t) + \mu_{31}P_3(t), \\ \frac{dP_2(t)}{dt} = \lambda_{12}P_1(t) - (\mu_{21} + \lambda_{23})P_2(t) + \mu_{32}P_3(t), \\ \frac{dP_3(t)}{dt} = \lambda_{13}P_1(t) + \lambda_{23}P_2(t) - (\mu_{31} + \mu_{32})P_3(t), \\ P_1(t) + P_2(t) + P_3(t) = 1. \end{cases}$$

### 1.4.3. Логико-вероятностный метод расчета надежности

Метод основан на математическом аппарате алгебры логики. Расчет надежности системы управления предполагает определение связи между сложным событием (отказ системы) и событиями, от которых оно зависит (отказы элементов системы). Следовательно, расчеты на надежность основаны на проведении операций с событиями и высказываниями. В качестве событий и высказываний принимаются утверждения о работоспособности или отказе элемента (системы). Каждый элемент системы представляется логической переменной, которая может принимать значение 1 или 0.

События и высказывания при помощи операций дизъюнкции, конъюнкции и отрицания объединяются в логические уравнения, соответствующие условию работоспособности системы. Составляется логическая функция работоспособности. Расчет, основанный на непосредственном использовании логических уравнений, называется логико-вероятностным.

**Расчет логико-вероятностным методом выполняется в семь этапов:**

1) словесная формулировка условий работоспособности объекта. Описывается зависимость работоспособности информационной системы от состояния ее отдельных элементов;

2) составление логической функции работоспособности, которая представляет собой логическое уравнение, соответствующее условию работоспособности системы управления;

$$F_{Л} = f(x_1, x_2, \ldots, x_n) = \bigvee_{i=1}^{s} D_i$$

$$F_1 = D_1 \vee D_2 = \underbrace{x_1 \cdot x_2 \cdot x_3}_{D_1} \vee \underbrace{x_1 \cdot x_2 \cdot \bar{x}_3}_{D_2}$$

$x_i$ – условие работоспособности i-го элемента $F_л$ представляется в дизъюнктивной форме.

3) приведение к логической функции работоспособности к ортогональной бесповторной форме. Сложную логическую функцию работоспособности необходимо привести к ортогональной бесповторной форме.

$$F_{Л} = f(x_1, x_2, \ldots x_n) = \bigvee_{i=1}^{s} D_i$$

$$F_{ЛО} = \bigvee_{i=1}^{s} D_i$$

Функция называется ортогональной, если все ее члены $D_i$ попарно ортогональны (т.е., их произведение равно нулю), и бесповторной, если каждый ее член $D_i$ состоит из букв $x$, с разными номерами (т. е. отсутствуют повторяющиеся аргументы).

$$F_{ЛО} = \underbrace{x_1 \cdot x_2 \cdot x_4}_{D_1} \vee \underbrace{x_3 \cdot \bar{x}_2}_{D_2}$$

$(x_2 \cdot \bar{x}_2 = 0) \Rightarrow$ ортогональная функция

$F_{ЛО} = x_1 \cdot x_2 \cdot x_4 \vee x_3 \cdot x_2 \cdot \bar{x}_2 \Rightarrow$ ортогональная бесповторная форма

Функцию $F_л$ можно преобразовать к ортогональной бесповторной форме $F_{ло}$, используя законы и правила преобразования сложных высказываний. При расчетах наиболее употребительны следующие правила:

$$x_1 \cdot x_2 = x_2 \cdot x_1$$
$$x_1 \vee x_2 = x_2 \vee x_1$$
$$x_1 \cdot (x_2 \vee x_3) = x_1 \cdot x_2 \vee x_1 \cdot x_3$$
$$x_1 \cdot \overline{x}_1 = 0$$
$$x_1 \vee 1 = 1$$
$$x_1 \cdot 1 = 1$$
$$x_1 \cdot (x_1 \vee x_2 \vee \ldots \vee x_n) = x_1$$
$$x_1 \vee (x_1 \cdot x_2 \cdot \ldots \cdot x_n) = x_1$$
$$f(x_1, x_2, \ldots, x_n) = x_1 \cdot f(x_1, x_2, \ldots, x_n) \vee$$
$$\vee \overline{x}_1 \cdot f(x_1, x_2, \ldots, x_n)$$

4) арифметизация $F_{ло}$. По найденной ортогональной бесповторной логической функции работоспособности определяется арифметическая функция $F_a$ согласно формуле

$$F_{ЛО} = f(x_1, x_2, \ldots, x_n) = \bigvee_{i=1}^{s} D_i$$
$$F_a = \sum_{i=1}^{s} A_i$$

где $A_i$ - арифметическая форма членов $D_i$ функции $F_{ло}$.
Арифметизация членов $D_i$, в общем виде содержащих операции дизъюнкции, конъюнкции и отрицания, осуществляется заменой логических операций арифметическими по следующим правилам:

$$x_1 \wedge x_2 = x_1 * x_2$$
$$x_1 \vee x_2 = x_1 + x_2 - x_1 * x_2$$
$$\overline{x} = 1 - x$$

5) Определение вероятности безотказной работы системы.
Вероятность безотказной работы системы определяется как вероятность истинности логической функции работоспособности, представленной в ортогональной бесповторной форме, и вычисляется как сумма вероятностей истинности всех ортогональных членов этой функции алгебры логики. Все события (высказывания) заменяются их вероятностями (вероятностями безотказной работы соответствующих элементов).

$$P\{f(x_1, x_2, \ldots, x_n) = \bigvee_{i=1}^{s} D_i = 1\} = \sum_{i=1}^{s} P\{D_i = 1\}$$

6) вычисление требуемых показателей надежности системы управления по найденному показателю $P_c(t)$:

- $P_c(t)$ вероятность отказа;
- $Q_c(t) = 1 - P_c(t)$ плотность распределения наработки до отказа;
- $\lambda_c(t) = -\dfrac{\dfrac{dP_c(t)}{dt}}{P_c(t)}$ интенсивность отказов;

- $T_o = \int_{i=0}^{\infty} P_c(t)dt$ средняя наработка до отказа;

7) анализ соответствия полученных показателей надежности заданным техническим требованиям системы.

Допущения, принимаемые при логико-вероятностном методе: для элементов системы возможны только два состояния; метод применим для невосстанавливаемых систем; отказы элементов системы должны быть независимы.

**Пример решения задачи логико-вероятностным методом.**

Проиллюстрируем логико-вероятностный метод расчета надежности конкретным примером.

Пусть имеется 2 автоматизированных рабочих места, между которыми есть 5 каналов связи, которые соединены в виде мостиковой схемы. Определить вероятность безотказной работы каналов связи. Найти вероятность безотказной работы системы, имеющей мостиковую структуру, если известны вероятности безотказной работы каждой элемента $P_1 = P_2 = ... = P_5 = 0{,}9$.

Согласно выделенным этапам:

I) ИС будет работоспособна, если в работоспособном состоянии находятся следующие каналы:

- 1,2,3,4,5;
- 1,2,3,4;
- 1,5,4;
- 2,5,3;
- 1,3;
- 2,4.

2,3) Составление логической функция работоспособности приведение $F_л$ к ортогональной бесповторной форме $F_{ло}$. Применяя **формулу** вышеприведенных правил (см. 3 этап методики расчета), получаем:

$$F_Л = x_1 \cdot x_2 \cdot x_3 \cdot x_4 \cdot x_5 \vee x_1 \cdot x_2 \cdot x_3 \cdot x_4 \vee$$
$$\vee x_1 \cdot x_5 \cdot x_4 \vee x_2 \cdot x_5 \cdot x_3 \vee x_1 \cdot x_3 \vee x_2 \cdot x_4 =$$
$$= x_5 \cdot [(x_3 \vee x_4) \cdot (x_1 \vee x_2)] \vee \overline{x}_5 \cdot (x_1 \cdot x_3 \vee x_2 \cdot x_4)$$

4) арифметизация $F_{ло}$ с целью нахождения $F_a$:

$$F_a = \sum_{i=1}^{s} A_i = x_5[(x_3 + x_4 - x_3 \cdot x_4) \cdot (x_1 + x_2 - x_1 \cdot x_2)] +$$
$$+ (1 - x_5) \cdot (x_1 \cdot x_3 + x_2 \cdot x_4 - x_1 \cdot x_3 \cdot x_2 \cdot x_4)$$

5) вычисление вероятности безотказной работы. Поскольку вероятности безотказной работы каналов не зависят от времени и равны по величине 0,9, вероятность безотказной работы системы $P_c(t)$ определится следующим образом:

$$P_c(t) = P_5[(P_3 + P_4 - P_3 \cdot P_4) \cdot (P_1 + P_2 - P_1 \cdot P_2)] +$$
$$+ (1 - P_5) \cdot (P_1 \cdot P_3 + P_2 \cdot P_4 - P_1 \cdot P_3 \cdot P_2 \cdot P_4) =$$
$$|P_i = 0{,}9| = 0{,}978$$

## 1.5. Надежность программного обеспечения

### 1.5.1. Введение в надежность программного обеспечения

Развитие информационно-вычислительных систем (ИВС), как элемента управления, тесно связано с изменениями, происходящими в областях их применения. Под *ИВС* понимается система, предназначенная для хранения, поиска и выдачи информации по запросам пользователей.

Поскольку в новых условиях повышается значение результатов функционирования ИВС, это усиливает интерес пользователей и разработчиков к анализу качества создаваемых и эксплуатируемых программ, то есть программного обеспечения (ПО) ИВС.

Предпринят ряд попыток дать количественную оценку понятию качества ПО путем разработки ряда показателей, численно характеризующих многочисленные свойства, лежащие в основе этого понятия. Так, очевидно, что в современных условиях проектирование ПО следует осуществлять научными и промышленными методами, с обязательным применением различных способов измерения характеристик программ.

Для этого необходимым условием является измерение и прогнозирование характеристик качества программ, а также изучение зависимости этих характеристик от различных параметров. ПО характеризуется прежде всего конкретными функциональными показателями качества или показателями назначения, номенклатура и значения которых определяется целями и областью применения программ. Каждый комплекс программ, кроме того, характеризуется конструктивными показателями качества, номенклатура которых почти не зависит от назначения и области использования программ. Набор этих показателей весьма близок к общим характеристикам обычных сложных промышленных изделий и включает сложность, трудоемкость, надежность и т.д.

Надежность – один из важнейших факторов, определяющих общую производительность и эффективность систем. Теория надежности как наука первоначально получила развитие применительно к сложным техническим системам, что в дальнейшем позволило адаптировать и развить эту теорию в направлении анализа надежности ПО.

Для оценки надежности применяется ряд показателей, выбор и приоритет которых зависят от конкретного типа и области применения объекта или системы

Существует несколько методов достижения высокого качества ПО: безошибочное проектирование («пассивные» методы) и выявление и устранение ошибок («активные» методы). Методы безошибочного проектирования (предотвращения ошибок) основываются на применении организационных и методологических правил проектирования ПО. Активные методы поиска и устранения ошибок дополняют пассивные в процессе достижения заданного качества ПО и позволяют оценивать ряд показателей качества. Основным активным методом является тестирование, которое состоит в проверке программ на соответствие заданным правилам построения и конкретным результатам их исполнения. Для эффективного тестирования программ необходима методологическая и инструментальная (средства автоматизации) база.

Оценка показателей надежности ПО и их прогнозирование производятся на базе математических моделей надежности программ, которые могут быть классифицированы следующим образом:

1) *модели, связанные с теорией надежности аппаратуры* и содержащие предположения о вероятностном распределении ошибок в ПО;

2) *модели, не базирующиеся на теории надежности аппаратуры*, но позволяющие получить приемлемые результаты оценки;

3) *модели сложности*, позволяющие оценить продукт проектирования и разработку с учетом сложности.

Таким образом, предпринята попытка обобщить и систематизировать работы в области надежности ПО с целью сформулировать основные концепции теории и практики

определения надежности. Проведенные обобщения позволят специалистам более целеустремленно и критически подходить к новым публикуемым методам повышения надежности программ, реклама которых не всегда соответствует реальной практической ценности.

### 1.5.2. Основные понятия надежности программного обеспечения

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ [ЭКИН]
*Программное обеспечение*- это совокупность программных средств для создания и эксплуатации систем средствами вычислительной техники. В состав программного обеспечения входят базовые (общесистемные) и прикладные (специальные) программные продукты.
Базовые программные средства служат для автоматизации взаимодействия человека и компьютера, организации типовых процедур обработки данных, контроля и диагностики функционирования технических средств вычислительных систем (ВС).
Прикладное программное обеспечение представляет собой совокупность программных продуктов, предназначенных для автоматизации решения функциональных задач систем. Они могут быть разработаны как универсальные средства (текстовые редакторы, электронные таблицы, СУБД), и как специализированные - реализующие функциональные подсистемы (бизнес-процессы) объектов различной природы (экономические, инженерные, технические и т.п.).

КАЧЕСТВО ПО [Липаев_1]
*Качество* рассматривается в традиционной трактовке как совокупность свойств системы, которые обуславливают пригодность применения этой системы по назначению. Поскольку программное обеспечение в общем случае также можно рассматривать как систему, то есть как совокупность подсистем или элементов, функционально объединенных в соответствии с некоторым алгоритмом взаимодействия при выполнении задачи, то общепринятое определение качества применимо и к понятию ПО.

НАДЕЖНОСТЬ [Шураков]
Фундаментальным является понятие *надежности* как свойства объекта выполнять заданные функции, сохраняя во времени значения установленных эксплуатационных показателей в заданных пределах, соответствующих заданным режимам и условиям использования, технического обслуживания, ремонтов, хранения и транспортирования. Свойство надежности проявляется в том, что система выполняет поставленные перед ней задачи. Потеря надежности системой связывается с появлением отказов в работе.

НАДЕЖНОСТЬ ПО [Липаев_1]
Общее определение надежности применимо и для ПО с учетом различия между типами возможных ошибок, степени влияния ошибок на функционирование системы и частоты их проявления.
Таким образом, *надежность ПО* – это вероятность того, что программа некоторый период времени будет работать без сбоев с учетом степени их влияния на выходные результаты. Следовательно, надежность ПО есть функция от ущерба, наносимого ошибкой пользователю. Надежность ПО в значительной степени отличается от надежности оборудования, поскольку ПО не подвержено износу, практически отсутствуют ошибки производства, так как встречаются они редко и легко могут быть исправлены (например, ошибки перезаписи при копировании).

ПОКАЗАТЕЛИ НАДЕЖНОСТИ [Шураков]
*Показатель надежности*- это совокупность величин, характеризующих качественно или

количественно степень приспособленности систем к выполнению поставленной задачи при применении по назначению. В практике проведения исследований вводятся в рассмотрение три вида показателей надежности сложных систем: качественные, порядковые и количественные

*Качественные показатели надежности* [Шураков] - это показатели, которые не могут быть выражены в виде числа и не содержат информации, позволяющей обосновать предпочтение одного из нескольких конкурирующих вариантов системы при их сравнении. Качественные показатели надежности указывают на то, что рассматриваемая система обладает каким-либо свойством для выполнения поставленной задачи. Качественные показатели дают возможность отличать системы друг от друга, но не позволяют сравнивать их по степени выполнения поставленной задачи, т.е. по надежности. Часто при проведении исследований надежности анализ ограничивается исключительно качественными показателями. Такой подход сужает сферу применения методов исследован надежности систем и ограничивает значимость полученных результатов с помощью этой теории.

*Порядковые показатели надежности* [Шураков] - это показатели, которые содержат информацию, позволяющую обосновывать предпочтение одного из вариантов при их сравнении без количественной оценки степени предпочтения. Порядковые показатели дают возможность расположить в ряд по степени возрастания надежности исследуемые варианты системы, но не позволяют оценить, на какую величину отличается достигнутый уровень надежности рассматриваемых вариантов. В результате исследования можно построить доминирующую последовательность вариантов системы путем их ранжирования, т.е. установления порядкового номера или применительно к каждому варианту системы, указывающего его место среди других вариантов.

*Количественные показатели надежности* [Шураков] - это показатели, которые содержат информацию, обеспечивающую оценку степени предпочтения одного варианта системы по отношению к другому при применении по назначению. Количественные показатели выражают надежность в виде числа: при помощи количественных показателей надежность измеряется или оценивается в принятой шкале оценок в абсолютных или относительных единицах. Количественные показатели определяются путем непосредственных статистических наблюдений на основе обработки результатов применения или испытания систем, а также путем аналитических расчетов или моделирования процесса функционирования систем. Они являются основными показателями надежности, обобщающими наиболее ценную информацию о степени приспособленности систем к применению по назначению. В ряде исследований вопроса надежности ПО показатель надежности рассматривался как функция времени наработки. Однако характерно, что ПО не изменяется существенно по мере наработки, а изменяется в результате устранения ошибок. Кроме того, вероятность отказа в работе не имеет прямого отношения к времени наработки. Зависимости для оценки надежности ПО могут быть непосредственно получены на основе свойств программы и элементарных понятий статистики.

### 1.5.3. Оценка надежности программного обеспечения

В основе показателей надежности лежат понятия о возможных состояниях ПО как системы [Липаев_1]: работоспособном и неработоспособном. *Работоспособным* называется такое состояние объекта, при котором он способен выполнять заданные функции с параметрами, установленными требованиями технической документации. В процессе функционирования возможен переход из работоспособного состояния в

*неработоспособное* и обратно. Первопричиной нарушения работоспособности программ при безотказности аппаратуры всегда является конфликт между реальными исходными данными, подлежащими обработке, и программой осуществляющей эту обработку. Реальные исходные данные могут иметь значения, отличающиеся от заданных техническим заданием и от проверенных при тестировании программ. При таких исходных данных функционирование программ трудно предсказать заранее и весьма вероятны различные аномалии, завершающиеся отказами.

Наиболее широко используется критерий *длительности наработки на отказ*. Для определения этой величины измеряется время работоспособного состояния системы между последовательными отказами от начала функционирования системы. Вероятностные характеристики этой величины в нескольких формах используются как разновидности критериев надежности. Критерии надежности восстанавливаемых систем учитывают возможность многократных отказов и восстановлений. Для оценки надежности таких систем, которыми чаще всего являются сложные комплексы программ, кроме вероятностных характеристик наработки на отказ, важную роль играют характеристики функционирования после отказа в процессе восстановления. Основным показателем процесса восстановления является *длительность восстановления* и ее вероятностные характеристики. Этот критерий учитывает возможность многократных отказов и восстановлений.

Обобщение характеристик отказов и восстановлений производится в критерии *коэффициент готовности*. Этот показатель отражает вероятность иметь восстанавливаемую систему в работоспособном состоянии в произвольный момент времени. Значение коэффициента готовности соответствует доле времени полезной работы системы на достаточно большом интервале, содержащем отказы и восстановления. Применение основных понятий теории надежности для оценки качества сложных комплексов программ позволяет получить ряд четких хорошо измеряемых интегральных показателей качества программ.

В качестве *исходных предпосылок* при построении зависимостей для оценки надежности используются следующие утверждения:

1. в результате выполнения программы для каждого множества $Ni$ входных данных получается однозначный выходной результат;
2. множество $N_i$ входных данных определяет все вычисления, выполняемые программой;
3. каждая ошибка в программе вызывает сбои для некоторой части входных данных;
4. пропуск программы с некоторым подмножеством входных данных представляет собой единичное наблюдение ее действия.

Тогда вероятность $\boldsymbol{P}$ появления сбоя есть вероятность того, что множество входных данных $N_i$, выработанное для пропуска программы, таково, что порождает сбои. $\boldsymbol{P}$ может быть выражено через вероятность $P_i$ выбора для работы множества $\boldsymbol{M}$ и переменную $\boldsymbol{y_i}$:

$\boldsymbol{y_i} = 0$, если выходной результат верен для $N_i$,

$\boldsymbol{y_i}$ =1, в противном случае.

Вероятность безотказной работы ПО может быть представлена следующим образом

$$P = 1 - \sum_{i=1}^{n} P_i y_i \ ,$$

где $n$ – число всех возможных входов ПО.

Надежность $\boldsymbol{R}$ для некоторого рабочего участка $P_i$ может быть определена:

$$R = 1 - P = \sum_{i=1}^{n} P_i(1 - y_i)$$

Для получения вероятности отказа ПО экспериментальным путем можно применить следующий подход. Если испытывается определенное ПО относительно $n$ различных входов и относительно $l$ из них имеют место отказы, то вероятность отказа ПО оценивается как

$$P_i = \frac{l}{n}$$

При равномерном распределении испытуемых входов во входном множестве и при достаточно большом $n$ $P \gg P_i$

В обычном ПО входы во входном множестве редко распределяются однородно, можно говорить лишь о некотором используемом множестве входов. В связи с этим устранение ошибок будет производиться именно в используемом множестве и таким образом вероятность отказа в этом множестве ниже, чем в полном множестве входов. Необходимо рассмотреть входное множество $S_в$ и используемое множество $S_и$. Вероятность отказа в $S_и$ определяется во время программирования, а вероятность отказа в $S_и$ определяется в соответствии с устранением ошибок. Предположив, что $P_в$ и $P_и$ как вероятности отказа в $S_в$ и $S_и$ соответственно постоянны, можно записать на основе эксперимента:

$$P_и \approx \frac{1}{n}.$$

В предположении, что $m$ причин отказа устранено, можно записать значение вероятности отказа после устранения ошибок для некоторой промежуточной точки:

$$P_и^{j+m} = P_и^j - \frac{m}{n_и}.$$

Здесь $P_и^j$ может быть получено на основании предыдущей зависимости. При полном устранении всех ошибок ПО становит надежным, т. е.

$$P_и^{j+m} = P_и^j - \frac{m}{n_и} = 0 \text{ и } m = P_и^j \times n_и.$$

Однако в процессе устранения ошибок могут быть внесены новые ошибки. Допускается, что если новые ошибки могут возникать с одинаковой вероятностью во входном множестве, то вероятность ошибок, возникающих в используемом множестве, может быть представлена в виде отношения $n_и:n_в$.

Если предположить, что устранение одной ошибки вызывает $K$ новых ошибок, то вероятность отказа после устранения $m$ ошибок может быть представлена следующим образом:

$$P_и^{j+m} = P_и^j - \frac{m}{n_и} \times (1 - \frac{n_и}{n_в} \times K).$$

Программное обеспечение становится совершенным тогда, когда

$$P_u^j - \frac{m}{n_u} \times (1 - \frac{n_u}{n_e} \times K) = 0 .$$

Наличие подобного механизма обеспечивает возможность предсказания развития процесса устранения ошибок. С точки зрения надежности ПО, чем меньше значение, тем быстрее можно провести устранение ошибок и тем выше будет надежность. Таким образом, надежность ПО представляется через определение вероятности отказа в отношении входного множества данных и через наблюдение интенсивности изменения вероятности отказа по мере устранения ошибок. Приведенные показатели в силу их количественной интерпретации могут быть использованы в качестве стандартных оценок ПО.

### 1.5.4. Отказы программного обеспечения

Основным принципом классификации *сбоев* и *отказов* в программах является разделение по временному показателю длительности восстановления после любого искажения программ, данных или вычислительного процесса, регистрируемого как нарушение работоспособности [Липаев_3]. При длительности восстановления меньше заданного порога $t_д$, аномалии при функционировании программ следует относить к *сбоям*, а при восстановлении, превышающем по длительности пороговое значение $t_д$, происходящие искажения соответствуют *отказу*.

Отказы по степени их влияния на функционирование комплекса программ делятся на следующие группы:

искажения вычислительного процесса и данных, вызывающие полное прекращение выполнения функций системой управления на длительное или неопределенное время, — полный отказ, в значительной степени обесценивающий результаты предыдущего функционирования;

искажения, кратковременно прерывающие функционирование системы и мало искажающие накопленные данные и выдаваемые результаты, — частичный отказ или длительный сбой, в некоторой степени обесценивающий предыдущие результаты;

искажения, кратковременно и мало отражающиеся на вычислительном процессе и обрабатываемых данных, — сбои, практически не обесценивающие результаты функционирования комплекса программ.

В зависимости от *глубины контроля и длительности запаздывания в обнаружении отказа*, а также в зависимости от *качества средств, осуществляющих восстановление*, одни и те же ситуации искажений вычислительного процесса или данных могут быть отнесены к разным типам отказов или сбоев.

Наиболее типичными полными отказами [Липаев_2] при функционировании сложных комплексов программ являются:

*зацикливание*, т. е. последовательное повторяющееся исполнение группы команд, которое не прекращается без внешнего вмешательства, блокируя функционирование всех остальных программ данного комплекса;

*останов* технических средств и полное прекращение решения функциональных задач, при этом может сохраниться возможность приема и выдачи информации и выполнения некоторых функций, стимулируемых прерываниями;

*значительное искажение или полная потеря данных* о состоянии внешних абонентов и процессе их функционирования;

*прекращение или недопустимое снижение темпа решения некоторых задач*, пропуск или потеря необработанных групп сообщений вследствие перегрузки технических средств или

вычислительной системы по пропускной способности.
Эти отказы существенно влияют на выполнение основных функций ПО.
В несколько меньшей степени на вычислительный процесс и обрабатываемые данные влияют искажения, приводящие к следующим типам частичных отказов или длительных сбоев [Липаев_2]:
*искажение заданной последовательности вызова подпрограмм*, приводящее к пропуску исполнения отдельных подпрограмм или их частей, что в свою очередь, может привести к неправильному или неполному решению некоторых задач и к искажению выходных результатов;
*использование и обработка искаженных исходных данных*, отражающиеся на логике решения задач и приводящие к искажению отдельных накопленных и выдаваемых данных.
Эти типы искажений заключаются в кратковременном нарушении нормального вычислительного процесса либо в искажении исходных, промежуточных или результирующих данных. В зависимости от повторяемости и глубины распространения искажения квалифицируются как частичные отказы либо как сбои с длительными последствиями.

В наименьшей степени на надежность функционирования влияет третий тип сбоев [Липаев_2], которые незначительно искажают общие результаты. Не нарушая практически логики функционирования комплекса программ, такие сбои искажают преимущественно *отдельные обрабатываемые и результирующие данные*. При поступлении аналогичных данных в последующие моменты времени можно ликвидировать последствия сбоев, не проводя специальных восстановительных работ.

### 1.5.5. Основные факторы, влияющие на надежность программного обеспечения

Одни и те же типы сбоев и отказов при исполнении задач ПО могут быть вызваны различными факторами [Липаев_2].
Факторы, влияющие на надежность функционирования ПО, можно разделить на три группы. В первую группу входят факторы, непосредственно вызывающие сбой или отказ при исполнении программы.
Ко второй группе факторов относятся архитектура ПО и структурное построение его компонент. Структура программ определяет возможность расширения последствий искажения информации или вычислительного процесса, влияет на вероятность превращения искажения в отказ и на время восстановления после отказа.
Третья группа факторов влияет на длительность восстановления и глубину последствий от возникающих отказов. В эту группу входят факторы, определяющие качество контроля вычислительного процесса и обрабатываемых данных, запаздывание в обнаружении искажений, качество классификации искажений и длительность проявления их последствий. Они определяют прежде всего длительность восстановления, время наработки на отказ и способствуют быстрой локализации искажений.
Причинами сбоя или отказа при выполнении программы могут быть:
1. *Искажения исходной информации*, поступающей от внешних абонентов [Липаев_2]. Искажения исходной информации в большинстве случаев непосредственно не влияют на надежность исполнения программ. Искаженные данные могут обрабатываться и являться причиной ошибок в результатах, выдаваемых внешним пользователям или накапливаемых в самой ИС. Однако некоторые искажения выходят за область допустимых значений переменных. При этом возрастает вероятность того, что искаженная величина будет обрабатываться некоторым сочетанием команд, приводящим к отказу либо к сбою функционирования.
*Причинами* искажений данных, поступающих от внешних абонентов, могут быть:

1) искажения данных на первичных носителях информации;
2) сбои и частичные отказы в аппаратуре ввода данных первичных носителей информации;
3) шумы и сбои в каналах связи при передаче сообщений по телекоммуникациям;
4) сбои и частичные отказы в аппаратуре передачи или приема телекодовой информации;
5) ошибки в документах, используемых для подготовки данных, вводимых в ЭВМ.

2. *Самоустраняющиеся отказы или сбои*. Они в технических средствах вычислительных систем являются фактором, существенно влияющим на надежность функционирования ПО [Липаев_2]. Наиболее часто происходят *сбои* или трудно обнаруживаемые *кратковременные отказы*. Большинство из них выявляется и устраняется средствами аппаратурного контроля и не влияет на исполнение программ. Однако некоторая часть аппаратурных сбоев может приводить к искажениям исполнения программ или к искажениям переменных. Причинами таких сбоев и отказов являются преимущественно внешние воздействия на аппаратуру ВС, влияющие на нарушение контактов и пропадание сигналов, или индустриальные электрические помехи. Это приводит к тому, что обнаруживаемые тестами сбои и самоустраняющиеся отказы происходят на один-два порядка чаще, чем устойчивые отказы.
3. *Невыявленные ошибки*. Эти ошибки в ПО являются основной причиной ненадежности функционирования [Липаев_2] . В процессе отладки основная часть ошибок в программах обнаруживается и устраняется, однако всегда есть риск пропуска нескольких ошибок. Любая отладка может показать наличие ошибок, но не может доказать их отсутствие. В процессе тестирования и отладки ПО программ практически невозможно выполнение абсолютно полных проверок, гарантирующих отсутствие непроверенных компонент программы и полное выявление всех возможных ошибок. В результате в сложных программах всегда существует некоторое количество невыявленных ошибок. В зависимости от *структурного построения* ПО последствия ошибки могут быть локализованы в некотором небольшом участке программ и данных либо распространиться на значительно большее расстояние от места расположения ошибки [Липаев_2]. Строгое иерархическое построение крупных комплексов программ на базе единообразно оформленных законченных программных модулей обеспечивает снижение вероятности ошибки в каждой команде программы и снижает возможность распространения последствий ошибок за пределы программного модуля и используемых в нем массивов данных.
В программах существуют конструкции потенциально ненадежные и способные приводить к отказам при небольших искажениях вычислительного процесса или массивов данных. Например, циклы, выход из которых организован при строгом сравнении с некоторым параметром, могут зацикливаться при искажениях этого параметра и шага цикла. Значительно более устойчивым является цикл, имеющий организацию выхода при достижении числа прохождений циклом заданного значения. Не отличаются устойчивостью к искажениям вычислений программные конструкции, содержащие безусловный переход по значению переменной (оператор *GO TO*). Любое искажение этой переменной может явиться причиной отказа вследствие переадресации в произвольные программы или массивы данных. На потенциальную надежность ПО влияет также структурное распределение оперативной памяти для локальных и глобальных переменных. Для обеспечения устойчивости при произвольном вызове подпрограмм и снижения вероятности разрушения данных целесообразно выделять зоны локальной памяти для каждой подпрограммы либо для подпрограммы одного иерархического уровня.
Таким образом, при формализации правил структурного построения подпрограмм и всего комплекса, выборе языка программирования и основных допустимых конструкций в

программах следует учитывать кроме эффективности программирования потенциальную склонность рекомендуемых конструкций к локализации и снижению вредных последствий любой ошибки или аппаратурного сбоя. Строгое модульно-иерархическое построение комплексов программ и максимально возможная автономизация зон оперативной памяти значительно повышают надежность функционирования таких комплексов.

Надежность восстанавливаемых объектов определяется длительностью наработки на отказ и длительностью восстановления [Липаев_2]. В программах отсутствуют компоненты, которые разрушаются физически и требуют для восстановления замены и ремонта. Поэтому время восстановления не зависит от объема технологических работ, ремонтных бригад, от их занятости, от наличия инструмента, комплектующих изделий и других факторов. При искажении вычислительного процесса или данных задача состоит в максимально быстром обнаружении искажения, в точной классификации возможных последствии искажения, а также в проведении мероприятий, обеспечивающих быстрое восстановление нормального функционирования. Средства контроля и помехозащиты программ и данных должны обеспечивать оперативное обнаружение и локализацию искажения, предотвращая расширение их последствий до уровня длительных отказов в функционировании ПО.

ведение средств контроля и помехозащиты в программы позволяет скомпенсировать их неполную отлаженность, а также снизить влияние возмущений других типов. Однако только средствами контроля и обеспечения программной помехозащиты невозможно достигнуть высокой надежности функционирования комплексов программ. Возникает оптимизационная задача распределения ресурсов на отладку и помехозащиту, обеспечивающих заданную надежность функционирования комплексов программ при минимальных суммарных затратах. Таким образом, так же как в аппаратурных комплексах, заданная надежность может быть достигнута повышением надежности компонент (отладкой подпрограмм), введением избыточности для контроля и резервирования (контроль и помехозащита комплекса программ) либо совместным сбалансированным применением этих методов повышения надежности.

### 1.5.6. Тестирование и отладка программного обеспечения

ПО как объект ***тестирования*** имеет ряд *особенностей* [Липаев_3]:

*отсутствие полностью определенного эталона* (программы), которому должны соответствовать все результаты тестирования проверяемой программы;

*высокая сложность* программ и принципиальная невозможность построения тестовых наборов, достаточных для их исчерпывающей проверки;

*невысокая степень формализации критериев качества* процесса тестирования и достигаемого при этом качества объектов тестирования;

*наличие в программах вычислительных и логических компонент*, а также компонент, характеризующихся стохастическим и динамическим поведением.

Тестирование является основным методом обнаружения ошибок при отладке программ. При этом затраты на тестирование являются наибольшими, достигают 30 - 40% общих затрат на разработку программ и в значительной степени определяют качество созданного программного продукта. Высокая доля затрат на тестирование приводит к необходимости создания методов и средств, позволяющих достигать максимального качества программ при реальных ограничениях на длительность тестирования и на связанные с этим затраты. Создаются различные методы систематического и регламентированного тестирования, обеспечивающие наилучшее использование ресурсов проектирования с учетом особенностей создаваемых программ.

Для определения *задач тестирования* целесообразно выделить три стадии:

I. *Тестирование для обнаружения ошибок в программе*.

Основной целью тестирования для обнаружения ошибок [Липаев_3] является выявление всех отклонений результатов функционирования реальной программы от заданных эталонных значений. При этом задача состоит в обнаружении максимального числа ошибок, в качестве которых принимается любое отклонение от эталонов. На этой стадии успешным является тестирование, которое приводит к обнаружению ошибок. Если в результате тестирования ошибки не выявлены, то проведенные операции не дали сведений, позволяющих повысить качество программ и тем самым не оправдали затрат. Таким образом, эффективными являются операции тестирования, обладающие высокой способностью по обнаружению ошибок в программе. Чем больше ошибок выявляется на этой стадии при каждой операции тестирования, тем выше их эффективность и обоснованность затрат на их выполнение. С этих позиций тесты, не способствующие обнаружению ошибок и только подтверждающие корректность функционирования программ, являются неэффективными.

II. *Тестирование для диагностики и локализации причин обнаруженных искажений результатов.*
Применяется после тестирования для обнаружения ошибок [Липаев_3]. На этой стадии важнейшая задача - точно установить место искажения программы или данных, явившегося причиной отклонения результатов от эталонных при тестировании для обнаружения ошибок. Тем самым определяется часть программы, подлежащая корректировке. Эффективными являются тесты, способствующие быстрой и точной локализации первичных ошибок. На этой стадии затраты оправданы и тестирование можно считать успешным, если оно приводит к полной локализации ошибки, подлежащей исправлению.

III. *Тестирование для контроля выполненных корректировок программ и данных (контрольное тестирование).*
Контрольное тестирование [Липаев_3] применяется после локализации и устранения обнаруженных ошибок, его задача состоит в подтверждения правильности выполненной корректировки программы и в отсутствии проявления ранее обнаруженных ошибок. В этом случае успешность тестирования определяется отсутствием проявления ранее обнаруженной. локализованной и устраненной ошибки, а также отсутствием вторичных ошибок, которые могут появиться при корректировке.
Для тестирования применяются методы, предусматривающие *упорядочение и систематизацию тестов по различным стратегиям и параметрам*, и методы *неупорядоченного тестирования*.
Основное внимание при упорядоченном тестировании сосредоточивается на обнаружении ошибок при исходных данных и условиях функционирования, заданных требованиями технического задания. Однако в реальных условиях на вход программы могут попадать сильно искаженные или ложные данные. Программы должны сохранять свою работоспособность при последующем поступлении данных, изменяющихся в заданных пределах. Для этого тестирование необходимо проводить не только при корректных исходных данных, но и при искаженных.
При неупорядоченном тестировании [Липаев_3] исходные данные, имитирующие внешнюю среду, случайным образом генерируются во всем диапазоне возможного изменения параметров, производится случайный перебор значений в произвольных сочетаниях различных величин. При этом многие значения исходных данных характеризуются малой вероятностью обнаружения ошибок и не оправдывают затраты на выполнение тестирования. Кроме того, возможно появление логически противоречивых данных. В то же время данные, наиболее важные с позиции реального использования программ и возможности обнаружения ошибок, могут оказаться не охваченными в

процессе тестирования. При реально существующих ограничениях на объемы тестирования его неупорядоченное применение оказывается малоэффективным и почти не находит применения.

Под ***отладкой*** понимается процесс, позволяющий получить программу, функционирующую с требующимися характеристиками в заданной области изменения входных данных. Таким образом, в результате отладки программа должна соответствовать некоторой фиксированной совокупности правил и показателей качества, принимаемой за эталонную для данной программы.

Процесс отладки включает в себя:

*создание совокупности тестовых эталонных значений и правил*, которым должна соответствовать программа по выполняемым функциям, структуре, правилам описания, значениям исходных и соответствующих им результирующих данных;

*статическое тестирование текстов разработанных программ и данных* на выполнение всех заданных правил построения и описания без исполнения объектного кода;

*тестирование программы с ее исполнением* в объектном коде и с разными уровнями детализации: детерминированное, стохастическое и тестирование в реальном масштабе времени;

*диагностику и локализацию причин отклонения результатов тестирования* от заданных эталонных значений и правил;

*разработку изменения программы* с целью исключения причин отклонения результатов от эталонных;

*реализацию корректировки программы*, обеспечивающую соответствие программы заданному эталону.

Контроль правил построения и описания программ и данных предполагает точную формализацию этих правил и проверку степени их выполнения. Относительно небольшое число используемых правил описания и построения программ и данных, а также четкая их формализация позволяют построить высокоавтоматизированные методы и средства контроля, и автоматически выявлять отклонения от таких эталонов.

Стремление к рациональному использованию ограниченных ресурсов приводит к систематизации процесса и методов тестирования [Липаев_3]. Методы упорядоченного тестирования базируются на выделении факторов и параметров, позволяющих эффективно распределять ресурсы тестирования с учетом их влияния на качество программ. Систематизация может значительно изменяться в зависимости от этапов тестирования, однако можно выделить несколько общих принципов, на базе которых строятся основные методы тестирования. Для упорядочения операции тестирования используется информация о структуре программы и процессе обработки информации, о характере изменения и взаимосвязи переменных, о наиболее вероятных и важных сочетаниях исходных данных, о характеристиках ошибок н вероятности их проявления и т. д. В результате ограниченные ресурсы В результате ограниченные ресурсы тестирования используются прежде всего для обнаружения наиболее опасных ошибок в наиболее важных режимах функционирования программ. С этой целью последовательно применяются методы тестирования: статический, детерминированный, стохастический и в реальном масштабе времени.

*Статическое тестирование* [Липаев_3] является наиболее формализованным и автоматизируемым методом проверки корректности программ. В качестве *эталонов* применяются правила структурного построения программных модулей и обработки данных, конкретизированные для проекта ПО в целом. Кроме того, могут использоваться некоторые частные правила обработки данных, зафиксированные в спецификациях на отдельные компоненты программ. Проверка степени выполнения этих правил проводится без исполнения объектного кода программы путем формального анализа текста программы на языке программирования. Операторы и операнды текста программ при

этом анализируются в символьном виде, поэтому такой метод называют также *символическим тестированием*. Развитие и углубление символического тестирования может доводиться до уровня формальной верификации программы на соответствие ее текста детальной спецификации совокупности утверждений, полностью определяющей связи между входными и выходными данными этой программы.

Наиболее трудоемкими и детализирующими являются методы *детерминированного тестирования* [Липаев_3]. При детерминированном тестировании контролируется каждая комбинация исходных эталонных данных и соответствующая ей комбинация результатов функционирования программы. Это позволяет выявлять отклонение результатов от эталона с конкретным фиксированием всех значений исходных и результирующих данных, при которых это отклонение обнаружено.

*Стохастическое тестирование* [Липаев_3] применяется в случаях, когда невозможно перебрать все комбинации исходных данных и проконтролировать результаты функционирования программы на каждой из них (в сложных программах). При этом виде тестирования исходные тестовые данные задаются множествами случайных величин с соответствующими распределениями и для сравнения полученных результатов используются также распределения случайных величин. В результате при стохастическом тестировании возможно более широкое варьирование исходных данных, хотя отдельные ошибки могут быть не обнаружены, если они мало искажают средние статистические значения или распределения. Стохастическое тестирование применяется в основном для обнаружения ошибок, а для диагностики и локализации ошибок приходится переходить к детерминированному тестированию с использованием конкретных значений параметров из области изменения ранее использовавшихся случайных величин.

Последующее расширение области изменения исходных данных возможно при применении *тестирования в реальном масштабе времени* [Липаев_3]. В процессе такого тестирования проверяется исполнение программ и обработка исходных данных с учетом времени их поступления, длительности и приоритетности обработки, динамики использования памяти и взаимодействия с другими программами и т.д. При обнаружении отклонений результатов исполнения программ от предполагавшихся эталонных для локализации ошибки приходится фиксировать время и переходить к детерминированному тестированию.

### 1.5.7. Ошибки программного обеспечения

Неоднократно экспериментально установлено, что в любом сложном комплексе программ в процессе эксплуатации обнаруживаются *ошибки*[Липаев_3], даже если проведено самое тщательное тестирование. Важной особенностью тестирования программ является невозможность получения полностью определенной абсолютно корректной программы, которую можно было бы использовать в качестве эталона без ошибок для сравнения с ней создаваемой программы. Поэтому при тестировании первоначально обнаруживаются *вторичные ошибки*, которые являются отклонениями некоторых результатов функционирования программ от предполагаемых эталонных значений. Тестирование для локализации ошибки позволяет установить причину вторичной ошибки и выявить *первичную ошибку*, подлежащую исправлению.

*Первичные ошибки* в программах в различной степени искажают результаты, которые первоначально обнаруживаются как *вторичные ошибки*, в процессе тестирования. Каждая первичная ошибка k-го типа отражается как вторичная ошибка j-го результата с некоторым коэффициентом пропорциональности $D_{kj}$. Если в некоторый момент тестирования имеющиеся первичные ошибки характеризуются вероятностями проявления $Q_k$, то они будут приводить к интегральным вторичным ошибкам с интенсивностью (весом), рассчитываемой как

$$\sigma = \sum_{j=1}^{q} \sum_{k=1}^{m} \Delta_{kj} Q_k \ ,$$

где $m$ — полное число возможных типов ошибок в программах;

$q$ — полное число контролируемых результатов, в которых могут проявляться вторичные ошибки.

Однако прямыми измерениями невозможно установить коэффициент влияния первичных ошибок на вторичные $D_{k\ j}$, а также вероятность первичных ошибок $Q_k$. Поэтому исследованы, в основном, обобщенные характеристики вторичных ошибок и некоторые общие закономерности проявления первичных ошибок в процессе тестирования программ.

В результате анализа и обобщения экспериментальных данных предложено несколько *математических моделей*, описывающих основные закономерности изменения суммарного числа вторичных ошибок в программах. Эти модели предназначены для оценки:

*возможного изменения надежности* функционирования в процессе отладки, испытаний и эксплуатации;

*числа ошибок*, оставшихся *невыявленными* в тестируемых программах;

*времени, требующегося для обнаружения следующей ошибки* в функционирующей или тестируемой программе;

*времени, необходимого для выявления всех ошибок* с заданной вероятностью.

Точное определение полного числа невыявленных ошибок в ПО прямыми методами измерения невозможно, поскольку в противном случае их можно было бы все зафиксировать и устранить. Однако имеются косвенные пути для приближенной статистической оценки их полного числа или вероятности ошибки в каждой команде программы. Такие оценки базируются на построении математических моделей в предположении жесткой корреляции между общим числом ошибок и их проявлениями в некотором ПО после его отладки в течение времени $t$, т.е. между следующими параметрами:

*суммарным числом первичных ошибок* в ПО ($n_0$) или вероятностью ошибки в каждой команде программы ($p_0$):

*числом вторичных ошибок*, выявляемых в единицу времени в процессе тестирования и отладки при постоянных усилиях на ее проведение ($dn/dt$);

*интенсивностью искажений результатов* в единицу времени ($l$) на выходе программы (вследствие невыявленных первичных ошибок) при функционировании системы в типовых условиях.

В результате может быть построена *экспоненциальная математическая модель распределения ошибок* в программах и установлена связь между интенсивностью обнаружения вторичных ошибок при тестировании $dn/dt$, интенсивностью $l$ проявления ошибок при нормальном функционировании ПО и числом выявленных первичных ошибок $n$. При этом учитываются все виды ошибок независимо от источников их происхождения (технологические, программные, алгоритмические, системные).

При постоянных усилиях на тестирование интенсивность обнаружения искажений и вычислительного процесса, программ или данных вследствие еще невыявленных ошибок пропорциональна числу $n_0$ оставшихся первичных ошибок в ПО. Тогда

$dn/dt = K'l = Kn_0 = K(N_0 - n)$,

где $N_0$ — число ошибок в ПО в начале отладки, а коэффициенты $K$ и $K'$ учитывают: масштаб времени, используемого для описания процесса обнаружения ошибок, быстродействие ЭВМ, распределение тестовых значений на входе проверяемого

комплекса программ и другие параметры. Значение коэффициента $K'$ можно, в принципе, определить как изменение темпа проявления ошибок при переходе от функционирования программ на специальных тестах к функционированию на нормальных типовых исходных данных. Так как предполагается, что в начале отладки при $t = 0$ отсутствуют обнаруженные ошибки, то $n = N_0 [1 – exp(-Kt)]$.

Число оставшихся первичных ошибок в ПО $n_0 = N_0 exp(-Kt)$, пропорционально интенсивности обнаружения $dn/dt$ с точностью до коэффициента $K$.

Длительность функционирования программ (наработка) между проявлением ошибок, которые рассматриваются как обнаруживаемые искажения программ, данных или вычислительного процесса, равна величине, обратной интенсивности обнаружения ошибок

$$T = 1 / \frac{dn}{d\tau} = \frac{1}{KN_0} \exp(K\tau) .$$

Если известны все моменты обнаружения ошибок $t_i$ и каждый раз в эти моменты обнаруживается и достоверно устраняется одна первичная ошибка, то, используя метод максимального правдоподобия, можно получить уравнение для определения значения начального числа первичных ошибок $N_0$

$$\sum_{i=1}^{n} \frac{1}{N_0 - (i-1)} = \frac{n \sum_{i=1}^{n} t_i}{N_0 \sum_{i=1}^{n} t_i - \sum_{i=1}^{n} (i-1) t_i} ,$$

а также выражение для расчета коэффициента пропорциональности

$$K = n / (N_0 \sum_{i=1}^{n} t_i - \sum_{i=1}^{n} (i-1) t_i) .$$

В результате можно рассчитать число оставшихся в программе первичных ошибок и среднюю наработку $T$ до обнаружения следующей ошибки. С помощью преобразований можно получить затраты времени на тестирование, которые позволяют устранить $dn$ ошибок и соответственно повысить наработку между очередными обнаружениями ошибок от значения $T_1$ до $T_2$

$$\Delta\tau = \frac{N_0 T_0}{K} \ln(T_2 / T_1) .$$

Необходимо подчеркнуть статистический характер приведенных соотношений. Неравномерность выбора маршрутов исполнения программы при нормальной эксплуатации, разное влияние конкретных типов ошибок в программах на проявление их при функционировании, а также сравнительно небольшие значения $n$ и $dn$, особенно на заключительных этапах отладки приводят к тому, что затраты времени на тестирование могут быть весьма значительными.

### 1.5.8. Математические модели надежности программного обеспечения

Одной из предпосылок использования математической статистики в проведении анализа надежности ПО является наличие данных [Шураков]. В начальной стадии развития

подходов к оценке качества исполнения ПО обсуждались модели теории массового обслуживания, имитационные и другие модели, описывающие этот процесс. Эти подходы привели к созданию специального программного обеспечения, имеющего своей основной целью сбор и обработку статистических данных. На основании полученных статистических данных возможно вычисление следующих показателей надежности: функции надежности $R(t)$;
среднего времени между ошибками программного обеспечения $t_{ср}$.
Стандартными функциями вероятности здесь выступают:

$$R(t) = P(t' > t);$$
$$F(t) = 1 - R(t);$$

где $t'$ – случайное переменное время сбоя;
$t$ – частное значение случайной переменной;
$R(t)$ – функция надежности, порождающая вероятность отсутствия сбоев в интервале времени от 0 до $t$;
$F(t)$ – кумулятивная функция распределения, порождающая вероятность сбоя в интервале времени от 0 до $t$;
$P(t < t')$ – вероятность того, что время прохождения сбоя лежит вне рассматриваемого интервала.
В области надежности ПО определяют также другую функцию условной вероятности, называемую количеством (скоростью, темпом) ошибок, или *функцию риска* $Z(t)$. Эта функция определяется в терминах вероятности того, что ошибка произойдет в интервале от $t$ до $t + dt$, т.е. вероятность появления ошибки: $P(t < t' < t+dt) = f(t)dt$.
Вероятность ошибки в интервале от $t$ до $t+dt$ при условии, что ошибка не произошла до $t$: $P(t < t' < t+dt / t' > t) = Z(t)dt$.
Из этих определений следует, что

$$Z(t) = \frac{f(t)}{R(t)} = -\frac{1}{R(t)} \times \frac{dR(t)}{dt}.$$

Решая это дифференциальное уравнение относительно $R(t)$ при начальных условиях $R(0) = 1$, получим:

$$R(t) = e^{-\int_0^t Z(x)dx}$$

Среднее время проявления ошибки задается следующей зависимостью:

$$t_{ср} = \int_0^\infty R(t)dt$$

Для простого случая, когда $Z(t)$ принимается постоянной на всем интервале исследования, уравнения принимают вид:

$$Z(t) = \lambda$$

$$R(t) = e^{-\lambda t}$$

$$t_{ср} = \frac{1}{\lambda}$$

Одним из способов оценки $t_{ср}$ является наблюдение за поведением программы в определенный временной период и на участке между двумя последующими ошибками. Время между обнаружением двух последовательных ошибок имеет тенденцию к возрастанию по мере обнаружения и корректировки ошибок. Экстраполируя этот ряд величин в будущее, можно с определенной вероятностью определить общее количество ошибок в разработанной системе. Гораздо лучшим приемом, требующим меньшее количество точек данных для той же точности прогноза, является постулирование модели для удаления ошибок и использования тестовых данных для оценки каждой модели.

*В моделях, базирующихся на теории надежности технических систем* [Шураков], количественными мерами для использования в модели описания качества ПО являются функция риска $Z(t)$, функция надежности $R(t)$ и среднее время между ошибками $t_{ср}$. Количество выявленных ошибок ПО в рассматриваемых моделях предполагается пропорциональным числу остаточных ошибок, при этом число остаточных ошибок принимается равным числу начальных ошибок за вычетом числа исправленных ошибок (последнее значение считается известным, если аккуратно ведутся записи отладки). Таким образом, модель содержит две неизвестные константы: число первоначальных ошибок и константу пропорциональности, для оценки которых используется информация функционального теста.

Необходимая информация, фиксируемая по каждому прогону программы тестирования, включает *длительность теста, наличие ошибки и ее классификацию* как аппаратурную, программную, ошибку оператора или неизвестную ошибку.

Каждый из $r$ последовательных прогонов представляет $T_1, T_2, \ldots, T_r$ часов успешного выполнения. При $n$ общих прогонов каждый $(n - r)$ неуспешный прогон представляется $t_1, t_2, \ldots, t_{n-r}$ часами успешного прохождения до появления ошибки. Общее количество часов успешного прогона $H$ определяется как:

$$H = \sum_{i=1}^{r} T_i + \sum_{i=1}^{n-r} t_i .$$

Предполагая, что количество ошибок постоянно, можно вычислить его как приведенное к одному часу работы:

$$\lambda = \frac{n - r}{H} . \qquad (1)$$

На основании соотношения для $t_{ср}$ величина среднего времени между сбоями определяется как:

$$t_{ср} = \frac{1}{\lambda} = \frac{H}{n - r} . \qquad (2)$$

На основании (1) и (2) определяется количество ошибок и среднее время между двумя смежными ошибками. Так как в расчет принимаются только ошибки ПО, то дальнейшему рассмотрению подлежат ошибки $x = n - r$.

На основе изложенного ошибки классифицируются на $x_n$ аппаратных ошибок, $x_s$ ошибок ПО, $x_o$ ошибок в работе оператора и $x_н$ неизвестных ошибок.
Так, количество ошибок ПО и среднее время между их проявлениями могут быть вычислены на основании:

$$\lambda_s = \frac{x_s}{H}$$

$$t_{ср} = \frac{H}{x_s}$$

В результате можно построить графики приведенных зависимостей по времени проведения отладки $t$ и провести на их основе количественную оценку прогресса в улучшении качества ПО путем экстраполяции значений показателей за пределы анализируемого временного интервала.
Среди *моделей, не базирующихся на теории надежности технических систем*, основная модель не требует предположения о постоянстве функции риска $Z(t)$ и может быть названа моделью, сеющей предварительные ошибки [Шураков].
Данная модель включает в себя средство, обеспечивающее рассеивание в тестируемую программу некоторого количества известных ошибок. Эти ошибки случайным образом вставляются в программу, а затем предполагается, что ошибки, имевшиеся в программе, и рассеянные ошибки должны быть обнаружены с равной вероятностью в результате последовательных тестовых прогонов. На основании тестирования программы в течение некоторого периода времени окажутся обнаруженными исходные и частично рассеянные ошибки.
Пусть в процессе тестирования обнаружено $n$ исходных ошибок и $v$ из $S$ рассеянных ошибок. Тогда оценка вероятности оставшихся исходных ошибок в программе составит:

$$N = \frac{S \times n}{v}.$$

При таком подходе оценка может быть получена практически после каждой обнаруженной исходной ошибки.
Вторая часть данной модели касается гипотез выражения и тестирования $N$. Пусть программа содержит $K$ исходных ошибок и $S$ рассеянных ошибок. Программа тестируется до тех пор, пока не будут обнаружены все рассеянные ошибки. В это же время количество обнаруженных исходных ошибок накапливается и запоминается. Далее вычисляется оценка надежности модели:

$$C = \begin{cases} 1, \text{ если } n > k; \\ \dfrac{S}{S+k+1}, \text{ если } n \le k, \end{cases}$$

как вероятность того, что она будет правильно отвергать ложные утверждения. Например, если утверждается, что программа не содержит ошибок ($k$ = 0), а в ней были рассеяны четыре ошибки, и все они обнаружены, но при этом не обнаружено ни одной исходной ошибки, то $C$ = 0,8. Для достижения значения вероятности 0,95 в такой ситуации необходимо рассеять и обнаружить 19 ошибок.

Приведенные зависимости для $N$ и $C$ составляют полезную модель ошибок, так как первая из них позволяет прогнозировать степень завершенности тестирования, а вторая может использоваться для оценки меры достоверности прогноза. Одним из недостатков этой модели является то, что $C$ не может быть предсказано до тех пор, пока не будут обнаружены все рассеянные ошибки.
Для преодоления этого недостатка зависимость для $C$ может быть модифицирована таким образом, что оценка может быть выполнена после обнаружения $j$ из $S$ рассеянных ошибок:

$$C = \begin{cases} 1, \text{ если } n > k; \\ \left(\frac{S}{j-1}\right) / \left(\frac{S+k+1}{k+j+1}\right), \text{ если } n \le k. \end{cases}$$

Рассмотренная модель рассеивания ошибок математически достаточно проста и позволяет получить статистически приемлемые результаты оценок.
Одним из подходов к оценке *сложности ПО* является представление программ как последовательности узлов, дуг и петель (циклов) в виде ориентированного графа (*имитационная модель*).
В ориентированном графе узлы представляют собой точки, в которых части программы могут объединяться или разъединяться, а дуги представляют собой последовательность линейных участков типа вычислений по заданным формулам, ввод-вывод и т.д. Инструкции размещаются в дугах, а ошибки имеют место в некоторых из инструкций. Ввод определяет путь от начального узла до узла выхода. Начинаясь в начальном узле, ввод вызывает выполнение инструкций на своем пути, расходуя тестовое время до момента встречи ошибки. После обнаружения ошибки она исправляется, при этом расходуется некоторое время на исправление. Имеется определенный риск, что исправление внесет новую ошибку. после повторного пуска в начальном узле выполнение программы начинается с тем же самым вводом. Этот процесс повторяется до тех пор, пока на данном пути не будет встречено ошибок.

### 1.5.9. Литература по надежности программного обеспечения


1. Липаев В.В. Качество программного обеспечения. – М.: Финансы и статистика, 1983. – 263 с.: ил.
2. Липаев В.В. Надежность программного обеспечения АСУ. – М.: Энергоиздат, 1981. – 240 с.: ил.
3. Липаев В.В. Тестирование программ. – М.: Радио и связь, 1986. – 296 с.: ил.
4. Шураков В.В. Надежность программного обеспечения систем обработки данных: Учебник. – 2-е изд., перераб. и доп. – М.: Финансы и статистика, 1987. – 272 с.: ил.
5. Экономическая информатика: Учебник для вузов / Под ред. д.э.н. проф. В.В. Евдокимова. - СПб.: Питер, 1997. - 592 с.: ил.


## 1.6. Экспериментальная оценка надежности

### 1.6.1. Понятие экспериментальной оценки надежности

Под экспериментальной оценкой надежности понимается определение и контроль различных показателей по результатам испытаний или наблюдений в процессе эксплуатации.
В общем комплексе мероприятий то обеспечению надежности экспериментальные оценки

играют существенную роль, в частности, позволяют оценить фактические значения показателей надежности и обосновать необходимость мероприятий по повышению надежности. Результаты экспериментальной оценки показателей надежности типовых элементов и узлов служат исходными данными при априорных оценках надежности вновь разрабатываемых изделий.

Точность и достоверность экспериментальных оценок определяют эффективность мероприятий по обеспечению надежности на всех этапах цикла «проектирование - производство - эксплуатация».

Экспериментальные оценки показателей надежности могут быть получены по результатам либо испытаний - специальных или совмещенных, либо наблюдений за функционированием изделий в условиях эксплуатации.

Специальными называются испытания, организуемые специально с целью определения (контроля) показателе надежности.

Совмещенными называются испытания, при которых определение (контроль) показателей надежности совмещаются с экспериментальным исследованием других параметров изделия.

Особенностью специальных испытаний является то, что объем их обычно заранее планируется, а условия функционирования изделий устанавливаются исходя из требований оценки конкретных показателей надежности. Такие испытания, как правило, организуются для изделий, выпускаемых в достаточно большом количестве. Проводить специальные испытания для сложных изделий и систем во многих случаях не представляется возможным, так как объем выпуска обычно ограничен единицами экземпляров, а процесс изготовления, отладки, проверки функционирования и довод и занимает слишком много времени. Показатели надежности таких изделий оцениваются в основном по результатам либо совмещенных испытаний, либо наблюдений на этапе эксплуатации.

При обработке экспериментальных данных отмеченные различия не существенны, поэтому ниже во всех случаях термины «испытания» и «наблюдения» используются как синонимы.

При экспериментальных оценках могут быть использованы прямые либо косвенные методы.

Прямыми называются методы, при которых показатели надежности изделия оцениваются непосредственно по результатам наблюдения за функционированием изделия как целого.

Косвенными называются методы, при которых требуемые показатели надежности выражаются через другие показатели надежности изделия или его элементов, а затем определяются расчетным путем. Весьма широко распространено другое название этих методов - «расчетно-экспериментальные методы» (РЭМ).

Прямые методы обладают большей достоверностью, однако для изделий, имеющих структурную избыточность, использование косвенных методов позволяет существенно уменьшить требуемый объем испытаний (наблюдений). Для сложных изделий, испытываемых практически всегда в неполном составе, косвенные методы являются единственно приемлемыми.

Экспериментальная оценка показателей надежности требует значительных затрат времени. Сокращение времени (ускорение) испытаний может быть достигнуто применением либо специальных методов планирования и обработки, либо формированных режимов испытаний.

Ускоренными называются любые испытания, при которых используются те или иные методы сокращения времени испытаний.

Форсированными называются ускоренные испытания, при которых ускорение достигается ужесточением (формированием) режимов с целью набора необходимого количества

статистической информации за более короткое время.
Применение формированных испытаний требует большой предварительной, подготовки: выбора эффективных ускоряющих факторов, исследования степени изменения показателей надежности при различных уровнях ускоряющего фактора. Испытания в форсированных режимах целесообразны прежде всего для контроля надежности серийных изделий, выпускаемых по неизменной технологии в течение длительного времени.

Экспериментальные оценки надежности преследуют одну из следующих целей:

- определение фактических значений показателей надежности;
- контроль соответствия изделия заданному требованию.

Определительная и контрольная постановки задачи имеют существенные отличия. При сопоставимы требованиях к точности и достоверности требуемый объем испытаний при контрольной постановке может быть существенно меньшее чем при определительной в случае, если истинное значение показателя надежности изделия существенно отличается от требуемого уровня.
Для контрольной и определительной процедур, кроме того, существенно различны этапы планирования.
Планирование контрольной процедуры опирается на требуемое значение показателя надежности. В результате планирования определяется необходимыми объем испытаний, и оценочный норматив - решающее правило, по которому принимается решение о соответствии или несоответствии изделия заданному требованию. Следовательно, ошибка в планировании контрольной процедуры в принципе не может быть выявлена в результате испытаний, и, таким образом корректность планирования непосредственно определяет достоверность заключения о соответствии или несоответствии изделия заданному требованию.
При планировании определительной процедуры принципиально невозможно однозначно указать необходимый объем испытаний, так как точность оценок показателей надежности при заданной достоверности зависит не от объема испытаний, а от объема получав ой при испытаниях информации. Исходя из требуемой точности и достоверности оценок, в результате планирования определительной процедуры получают не объем испытаний, а минимально необходимое число информативных реализаций.

Требуемый объем испытаний - число изделий, или число опытов, и продолжительность испытаний - зависит от фактической надежности изделия, которая до испытаний неизвестна. Следовательно, необходимый объем испытаний при планировании определительной процедуры может быть определен лишь ориентировочно, исходя из предполагаемого уровня надежности изделий. Однако ошибки в планировании объема определительных испытаний выявляются в процессе испытаний и при обработке их результатов и могут быть скорректированы.

Информация, которая требуется для обработки от предприятия, как правило, характеризуется небольшим числом наблюдений. Под малой выборкой понимается несплошное статистическое обследование, при котором выборочная совокупность образуется из сравнительно небольшого числа единиц генеральной совокупности. Объем малой выборки обычно не превышает 30 единиц и может доходить до 4-5 единиц.

Все задачи математической статистики касаются вопросов обработки наблюдений над случайными явлениями, но в зависимости от характера решаемого практического вопроса и от объема имеющегося материала эти задачи могут принимать ту или иную форму.

Охарактеризуем вкратце некоторые типичные задачи математической статистики, часто встречающейся на практике.

1. Задача определения закона распределения случайной величины (или системы случайной величины) по статистическим данным

Доказано, что закономерности, наблюдаемые в случайных явлениях, проявляются тем точнее и отчетливее, чем больше объем статистического материала. При обработке обширных по своему объему статистических данных часто возникает вопрос об определении законов распределения тех или иных случайных величин. Теоретически, при достаточном количестве опытов свойственные этим случайным величинам закономерности будут осуществляться сколь угодно точно. На практике нам всегда приходится иметь дело с ограниченным количеством данных; в связи с этим результаты наблюдений и их обработки всегда содержат больший или меньший элемент случайности. Возникает вопрос о том, какие черты наблюдаемого явления относятся к постоянным, устойчивым и действительно присущи ему, а какие являются случайными и проявляются только за счет ограниченного объема экспериментальных данных. Естественно, к методике обработки экспериментальных данных следует предъявить такие требования, чтобы она, по возможности, сохраняла типичные, характерные черты наблюдаемого явления и отбрасывала все несущественное, второстепенное, связанное с недостаточным объемом опытного материала. В связи с этим возникает характерная для математической статистики задача сглаживания или выравнивания статистических данных, представления их в наиболее компактном виде с помощью простых аналитических зависимостей.

Практически подавляющее большинство статистических распределений, встречающихся при исследовании систем автоматического управления химико-технологическими процессами по надежности и безопасности, можно описать одним из следующих пяти стандартных распределений:

Нормальным:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-m_x)^2}{2\sigma^2}}$$

Логарифмически-нормальным:

$$f(x) = \frac{1}{x b_1 \sqrt{2\pi}} e^{-\frac{(\ln x - a_1)^2}{2b_1^2}}$$

Гамма-распределением:

$$f(x) = \frac{b_2^{a_2}}{\Gamma(a_2)} x^{a_2-1} e^{-b_2 x}$$

Распределением Вейбулла:

$$f(x) = a_3 b_3^{a_3} x^{a_3-1} e^{-(x b_3)^{a_3}}$$

Равномерным распределением:

$$f(x) \quad \begin{cases} 0 \text{ при } x<b_4 \text{ и } x>a_4 \end{cases}$$

$$\frac{1}{a_4 - b_4} \text{ при } b_4 \le x \le a_4$$

2. Задача проверки правдоподобия гипотез
Эта задача тесно связана с предыдущей; при решении такого рода задач мы обычно не располагаем настолько обширным статистическим материалом, чтобы выявляющиеся в нем статистические закономерности были в достаточной мере свободны от элементов случайности. Статистический материал может с большим или меньшим правдоподобием подтверждать или не подтверждать справедливость той или иной гипотезы. Например, может возникнуть вопрос: согласуются ли результаты эксперимента с гипотезой о том, что данная случайная величина подчинена закону распределения $F(x)$? Другой подобный вопрос: указывает ли наблюдаемая в опыте тенденция к зависимости между двумя случайными величинами на наличие действительной объективной зависимости между ними или же она объясняется случайными причинами, связанными с недостаточным объемом наблюдений? Для решения подобных вопросов математическая статистика выработала ряд специальных приемов (будут рассмотрены далее).

3. Задача определения неизвестных параметров распределения
Часто при обработке статистического материала вовсе не возникает вопрос об определении законов распределения исследуемых случайных величин. Обыкновенно это бывает связано с крайне недостаточным объемом экспериментального материала. Иногда же характер закона распределения качественно известен, из теоретических соображений; например, часто можно утверждать заранее, что случайная величина подчинена нормальному закону. Тогда возникает более узкая задача обработки наблюдений — определить только некоторые параметры (числовые характеристики) системы случайных величин. При небольшом числе опытов задача более или менее точного определения этих параметров не может быть решена; в этих случаях экспериментальный материал содержит в себе неизбежно значительный элемент случайности; поэтому случайными оказываются и все параметры, вычисленные на основе этих данных. В таких условиях может быть поставлена только задача об определении так называемых «оценок» или «подходящих значений» для искомых параметров, т.е. таких приближенных значений, которые при массовом применении приводили бы в среднем к меньшим ошибкам, чем всякие другие.

### 1.6.2. Организация испытаний и сбор информации

Опросы организации испытаний, сбора и предварительной обработки информации являются общими и одинаково важными при любых видах испытаний на надежность - определительных и контрольных, нормальных и ускоряемых, специальных и совмещенных. При организации испытаний следует обратить внимание на следующие факторы:

- режим эксплуатации изделия при испытаниях (непрерывный или циклический);
- характер внешних воздействий (механические, климатические, электрические);
- объекты сбора статистики;
- состав, обязанности и ответственности членов испытательных групп;
- правила и порядок контроля работоспособности изделия;
- состав информации, которую необходимо фиксировать для анализа и оценки надежности;
- формы учетных документов для фиксации наработки и отказов;

- правило прекращения испытаний.

От степени проработки этих вопросов при подготовке испытаний зависит достоверность получаемых оценок показателей надежности.
Известно, что уровень фактической надежности изделия существенно зависит от параметров окружающей среды и режима функционирования изделия.
Если изделие предназначено для функционирования в широком диапазоне параметров среды, то целесообразно задать и проверять показатели надежности для различных (например, граничных) значений параметров среды. Если задан уровень показателя надежности и специально не оговорены соответствующие ему условия, то при испытаниях следует обеспечить наиболее характерные для данного изделия условия функционирования.

Правильный выбор объектов сбора статистики, в особенности для сложных изделий при использовании РЭМ, является не простой задачей. При слишком мелком делении изделия на самостоятельные объекты сбора статистики существенно усложняется учет, увеличивается число учетных документов, что неизбежно ведет с снижению достоверности получаемой информации. При чрезмерном укрупнении объектов сбора статистики может потеряться необходимая детализация информации о причине отказа, месте отказа и о фактической наработке отдельных элементов изделия.

Контроль работоспособности при испытаниях на надежность может быть непрерывным, периодическим, эпизодическим.
Наиболее полную информацию дает непрерывный контроль, который позволяет фиксировать моменты отказов изделий. Однако такой контроль не всегда может быть обеспечен.
Если при испытаниях осуществляется периодический контроль функционирования, то данные об отказах оказываются сгруппированными по интервалам контроля. В этом случае при выборе периодичности контроля рекомендуется руководствоваться следующими соображениями. Минимальный период контроля определяется только техническими и экономическими соображениями. Что касается ограничений «сверху», максимальная наработка между двумя последовательными проверками определяется при определительных испытаниях интересами статистической наработки: период контроля должен быть не слишком большим, чтобы на интервал испытаний приходилось не менее 10—15 межконтрольных периодов.

При контрольных испытаниях, в особенности в случае последовательных испытаний, слишком редкие проверки функционирования могут привести к существенным ошибкам. В этом случае частота контроля должна быть такой, чтобы вероятность двойного пересечения границы зон приемки или браковки за один межконтрольный период была пренебрежимо мала.

В зависимости от конкретных условий могут быть организованы испытания с заменой (восстановлением) или без замены отказавши изделий.
Для изделий с распределением времени работы до отказа, отличным от экспоненциального, целесообразная стратегия восстановления зависит также от вида определяемого показателя надежности. Если, например, определяется среднее время работы изделия до нового отказа, то после каждого отказа должно быть обеспечено полное восстановление изделия до первоначального состояния по всем параметрам.
Как правило, в процессе испытаний должны выполняться в установленные сроки все

регламентные работы, предусмотренные технической документацией. Однако если целью испытаний является определение оптимального межрегламентного (межремонтного) периода, то испытания должны проводиться при функционировании изделия до отказа без регламентного обслуживания. В результате такого эксперимента оценивают вид и параметры функции распределения времени работы до отказа, а затем определяется искомый период.

Возможности объективного анализа и обработки статистической информации, получаемой в результате испытаний, существенно зависят от полноты сведений о каждом случае нарушения функционирования (отказа, неисправности). Фиксации подлежат все случаи нарушения функционирования. Выделение из общей статистики отказов для различных оценок производится при обработке данных.

Существенной особенностью экспериментальной оценки показателей надежности является большой объем сведений, который необходимо фиксировать в каждом случае нарушения функционирования.

В процессе испытаний на надежность необходимо обеспечить фиксацию по крайней мере следующей информации:

- общая наработка изделия и время работы от момента предыдущего нарушения;
- место нарушения (завода кой и позиционный номера отказавшего элемента узла, детали);
- причина нарушения;
- последствия нарушения (полное нарушение работоспособности или частичное и по каким именно функциям );
- вид нарушения (поломка, износ детали, уход параметра и т.п.);
- способ устранения нарушения (замена элемента, регулировка, перестановка элементов и т.п.);
- данные об оперативности подключения и контролируемости резерва (для изделии, имеющих резервные компоненты);
- условия среды в момент нарушения функционирования (температура, вибрация, удары и другие сопутствующие явления, в том числе манипуляции обслуживающего персонала).

Весьма часто нарушения работоспособности сложных изделий, в особенности опытных образцов, не имеют отношения к свойству «надежность» (срабатывание автоматов защиты или перегорание предохранителей при бросках напряжения, зацикливание вычислительных устройств при некоторых определенных типах входной информации, остановы технологических линий при значительном изменении качества сырья и т. п.). Поэтому тщательная фиксация всех явлений, сопутствующих отказу, очень важна для правильной их классификации при предварительной обработке результатов испытаний.

Достоверность первичной информации обеспечивается полнотой и регулярностью записей, а также глубиной и объективностью анализа причин отказов. Важно иметь в виду, что недостоверные первичные данные невозможно улучшить даже самой тщательной статистической обработкой.

Наиболее распространенными учетными документами при экспериментальных оценках надежности являются аппаратный журнал и карточка учета неисправности.
Целесообразно, чтобы аппаратный журнал служил не только для учета наработки,

включений, выключений и фиксации нарушений работоспособности изделия, но и рабочим дневником испытателя. Журнал является первичным документом, в котором в хронологическом порядке отражается состояние изделия, все проводимые в процессе испытаний или эксплуатации работы, а также все замечания обслуживающего персонала по качеству функционирования, удобству обслуживания и ремонта и т.п. Записи в аппаратном журнале служат, как правило, основанием для заполнения карточки учета неисправности.

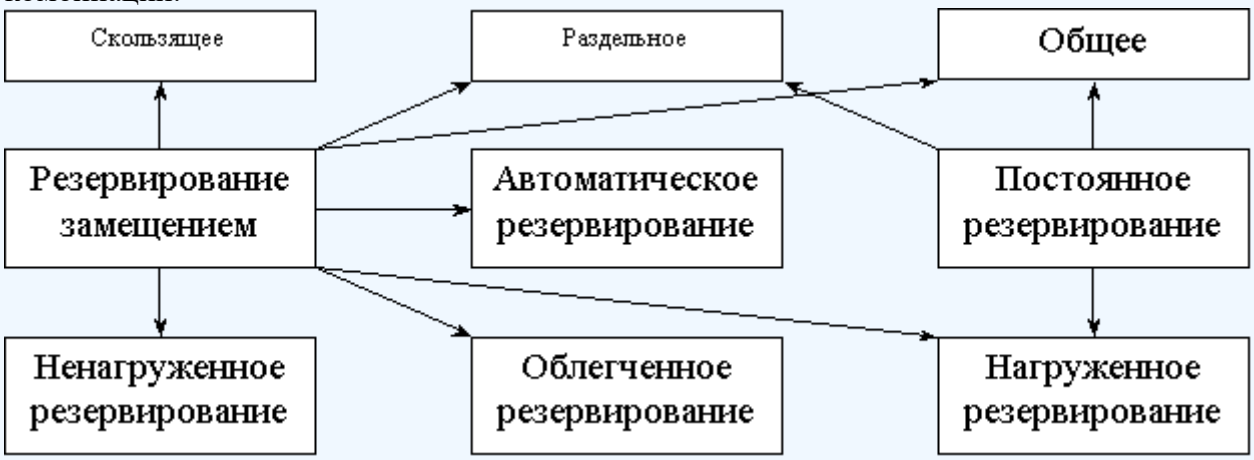
Карточка учета неисправностей заполняется по данным аппаратного журнала с привлечением другой информации (в том числе из ремонтных органов) и представляет собой своеобразный протокол по каждой неисправности. Карточки являются весьма удобной формой накопления статистической информации, учитывая в особенности необходимость последующей сортировки и классификации отказов по различным признакам при первичной обработке результатов испытаний.

## 1.7. Оптимальное резервирование

### 1.7.1. Методы и виды резервирования

**Резервирование** — способ повышения надежности системы путем включения в состав системы резерва предусмотренного на стадии проектирования этой системы или во время эксплуатации.

Ниже показана схема возможных методов и видов резервирования и возможных их комбинаций.




Резервирование замещением: при отказе элемента система перестраивается и в замен отказавшего подключается элемент из числа резервных.

Автоматическое резервирование: при отказе основного элемента, автоматически подключается резервный.

Постоянное резервирование: резервные и основные элементы находятся в одинаковых условиях и параллельно выполняют заданные функции.

Общее резервирование: резервируется вся система в целом.

Раздельное резервирование: резервируются отдельные участки системы.

Скользящее резервирование: один резервный элемент предназначен для резервирования некоторого множества основных элементов такого же типа. При отказе он заменяется.

Нагруженное резервирование: резервные элементы системы находятся во включенном состоянии, работают параллельно с основными элементами и практически одинаково расходуют свой ресурс работы.

Ненагруженное резервирование: резервные элементы находятся в выключенном состоянии и практически не расходуют свой ресурс работы.

Облегченное резервирование: резервные элементы находятся во включенном состоянии

однако расходуют свой ресурс намного меньше чем при подключении их на место основных.

### 1.7.2. Постановка задачи оптимального резервирования

ВС состоят из отдельных элементов. Эти элементы подвергаются капитальным и аварийным ремонтам, причем периодичность ремонтов неодинакова. При капитальных и аварийных ремонтах неисправную единицу удаляют из системы, а на ее место устанавливают прибор из резервного парка.

Модель должна определять оптимальный уровень резервного парка для каждого вида средств. По условию задачи требуется, чтобы система функционировала с максимальной надежностью. Если какой-нибудь элемент отказывает, то его заменяют запасным. Отказавший элемент сразу начинают ремонтировать. В системе непрерывно должно функционировать *zk* элементов *k*-го типа, причем в запасе должно иметься *xk* элементов того же *k*-го типа *(k=1, 2, …, S)*.

Сущность задачи заключается в оптимальном распределении стоимостных ресурсов *C*, выделенных на приобретение резервных единиц.

Рассматриваемая система считается отказавшей, если в момент отказа работающего элемента *k*-го типа все *xk* запасных элементов того же типа находятся в ремонте. Таким образом, надо найти количество резервных элементов

$\overline{X} = \{x_1, x_2, x_3 \ldots x_k\}$ так, чтобы не было их нехватки.

$P(\overline{X})$ — показатель надежности всей системы.

Так как модель соединений элементов с точки зрения надежности представляется как последовательное соединение элементов.

$$P(\overline{X}) = \max_{\overline{X}} \prod_{k=1}^{S} P_k(x_k)$$

Необходимо найти минимум риска нехватки элементов.

$$\min_{\overline{X}}[1 - P(\overline{X})] = \min_{\overline{X}} \prod_{k=1}^{S}[1 - P_k(x_k)]$$

На все элементы есть ограничения (вес, цена, объем и т.д.). Но мы будем пользоваться только стоимостным ограничением, как, пожалуй, наиболее часто встречающимся.

$C_0 - \sum_{i=1}^{k} C_i x_i \geq 0$, где *Ci* — стоимость одного элемента *i*-го типа.