**Оглавление**

[1. Предмет теории надежности 3](#_Toc200388636)

[2. Основные понятия и определения теории надёжности 3](#_Toc200388637)

[3. Понятие «отказ» и классификация отказов информационных систем 4](#_Toc200388638)

[4. Зависимость надёжности от времени 5](#_Toc200388639)

[7. Показатели надёжности информационных систем 6](#_Toc200388640)

[10. Надёжность восстанавливаемых систем 7](#_Toc200388641)

[11. Надёжность невосстанавливаемых систем 7](#_Toc200388642)

[20. Классификация ошибок программного обеспечения 8](#_Toc200388643)

[26. Зависимость надёжности от времени 9](#_Toc200388644)

[29. Показатели надёжности вычислительных систем 10](#_Toc200388645)

[30. Специфика информационной системы как объекта исследования надёжности 10](#_Toc200388646)

# **1. Предмет теории надежности**

Теория надёжности — это наука, которая изучает, насколько хорошо и стабильно работает система или устройство в течение времени. Проще говоря, она отвечает на вопросы вроде: «Как долго система будет работать без сбоев?», «Какова вероятность, что она откажет в самый неподходящий момент?», и «Что можно сделать, чтобы этого не произошло?»

Предметом этой теории являются закономерности отказов, причины и последствия этих отказов, а также методы повышения надёжности систем. Причём сюда можно отнести и технические устройства, и программное обеспечение, и целые информационные системы.

Основные задачи теории надёжности:

* анализировать причины возможных отказов;
* разрабатывать методы предотвращения отказов;
* оценивать вероятность безотказной работы за определённый промежуток времени;
* оптимизировать структуру и процессы для повышения надёжности.

В случае информационных систем сюда добавляются особенности вроде:

* надёжность хранения данных,
* устойчивость к сбоям оборудования и ПО,
* защита от потери или искажения информации,
* обеспечение непрерывности работы сервисов.

Таким образом, теория надёжности помогает проектировать такие ИС, которые стабильно выполняют свои функции, даже если что-то пойдёт не так.

# **2. Основные понятия и определения теории надёжности**

Теория надёжности — это область науки и техники, которая занимается изучением того, насколько надёжно работают различные изделия, системы и устройства. Под надёжностью понимается способность объекта выполнять свои функции в течение заданного времени и в определённых условиях.

В теории надёжности выделяют несколько ключевых свойств:

1. Безотказность — это способность изделия работать без поломок или сбоев в течение определённого времени. Например, если у нас есть насос, и он должен проработать 1000 часов без поломок — это и есть показатель его безотказности.
2. Долговечность — это общее "время жизни" изделия до наступления предельного состояния, когда его уже нельзя или нецелесообразно ремонтировать. Например, автомобильный двигатель рассчитан на 300 000 км пробега.
3. Ремонтопригодность — это возможность восстановить работоспособность изделия после отказа с помощью ремонта. Чем проще и быстрее ремонт, тем выше ремонтопригодность.
4. Сохраняемость — это способность изделия сохранять свои свойства при хранении или транспортировке, когда оно не используется. Например, запасной генератор может долго лежать на складе и при этом не потерять своих характеристик.

Чтобы надёжность можно было оценить количественно, используют разные показатели:

* Вероятность безотказной работы P(t) — это вероятность того, что изделие проработает без отказов в течение времени t.
* Интенсивность отказов λ(t) — это условная вероятность отказа в единицу времени.
* Среднее время безотказной работы (MTBF) — это среднее время, которое проходит между отказами.

Важность теории надёжности трудно переоценить — ведь от надёжности зависят безопасность, экономическая эффективность и удобство эксплуатации практически всех технических систем: от бытовой техники до самолётов и атомных станций.

# **3. Понятие «отказ» и классификация отказов информационных систем**

Отказ — это событие, при котором информационная система (ИС) или её компонент перестаёт выполнять заданные функции или начинает выполнять их неправильно. Проще говоря, это ситуация, когда система не работает так, как от неё ожидается.

Важно понимать, что отказ может быть как полным (система «упала»), так и частичным (например, одна функция не работает, а остальные работают нормально).

Классификация отказов информационных систем

Отказы ИС можно классифицировать по разным признакам:

1. По степени влияния на работоспособность системы:

Полный отказ — система полностью перестаёт работать.

Частичный отказ — работает только часть функций.

1. По продолжительности:

Кратковременный (временный) отказ — возникает на короткий промежуток времени, после чего система восстанавливает работу.

Постоянный отказ — требует вмешательства для восстановления (перезагрузка, ремонт, исправление кода).

1. По причинам возникновения:

Аппаратные отказы — связаны с неисправностью оборудования (жёсткий диск, сервер, сеть и т.п.).

Программные отказы — из-за ошибок в ПО (баги, утечки памяти и др.).

Человеческий фактор — ошибки пользователей или администраторов.

Внешние воздействия — например, сбои в электроснабжении, атаки злоумышленников, стихийные бедствия.

1. По способу проявления:

Открытые отказы — сразу заметны (например, сбой на экране, сообщение об ошибке).

Скрытые отказы — не проявляются сразу, но могут приводить к накоплению проблем (например, постепенная порча данных).

# **4. Зависимость надёжности от времени**

Надёжность системы — это её способность работать без отказов в течение определённого времени. Логично, что надёжность всегда *зависит от времени*: чем дольше система работает, тем выше вероятность того, что в какой-то момент она откажет.

Обычно эту зависимость описывают с помощью вероятности безотказной работы на интервале времени от начала эксплуатации до момента времени *t*. Чем больше *t*, тем ниже эта вероятность.

Чтобы объяснить проще, можно представить себе лампочку. Сразу после включения она почти наверняка будет работать. Но если оставить её гореть 1000 часов, вероятность того, что она всё ещё работает, станет гораздо меньше.

В теории надёжности принято выделять три характерных периода зависимости надёжности от времени (это часто называют "кривой "ванны" из-за её формы):

1. Начальный период (приработка)— вероятность отказов выше из-за возможных дефектов производства или ошибок настройки.— Со временем эти "детские болезни" устраняются.
2. Период нормальной эксплуатации— система работает стабильно, вероятность отказов почти постоянная и низкая.
3. Период старения— вероятность отказов начинает расти из-за износа оборудования, деградации программного обеспечения (например, накопления ошибок в данных или несовместимости с новыми компонентами).

Применительно к информационным системам:

Аппаратные компоненты (жёсткие диски, серверы) подвержены физическому износу — значит, с ростом времени вероятность их отказа растёт.

Программное обеспечение может устаревать (например, из-за изменения окружения или появления новых угроз безопасности), что тоже снижает надёжность со временем.

# **7. Показатели надёжности информационных систем**

Показатели надёжности — это числовые характеристики, которые помогают измерить и оценить, насколько система работает стабильно и без сбоев.

Основные показатели надёжности информационных систем:

1. Вероятность безотказной работы (Р)— это вероятность того, что система будет работать без отказа в течение заданного времени. Например, вероятность, что сервер не упадёт в течение суток.
2. Среднее время наработки на отказ (MTTF — Mean Time To Failure)— это среднее время, в течение которого система работает до первого отказа. Чем выше MTTF, тем надёжнее система.
3. Среднее время восстановления (MTTR — Mean Time To Repair)— это среднее время, которое требуется для восстановления работоспособности после отказа. Чем меньше MTTR, тем быстрее система возвращается в работу.
4. Среднее время между отказами (MTBF — Mean Time Between Failures)— это сумма времени работы и времени восёстановления, то есть средний интервал между двумя последовательными отказами. Часто используется для оценки общей надёжности.
5. Интенсивность отказов (λ, ламбда)— количество отказов в единицу времени. Чем меньше λ, тем надёжнее система.
6. Надёжность (R(t))— функция, показывающая вероятность безотказной работы в интервале времени t.

# **10. Надёжность восстанавливаемых систем**

Восстанавливаемая система — это такая система, которая после отказа может быть возвращена в рабочее состояние с помощью ремонта, перезагрузки или других восстановительных действий.

В отличие от невосстанавливаемых систем, где после отказа всё — система выходит из строя навсегда, у восстанавливаемых систем возможна повторная эксплуатация после устранения неисправности.

Надёжность восстанавливаемых систем оценивается не только вероятностью безотказной работы, но и скоростью и качеством восстановления.

# 11. Надёжность невосстанавливаемых систем

Невосстанавливаемая система — это система, которая после отказа считается выведенной из эксплуатации и не подлежит восстановлению или ремонту. То есть, когда такая система выходит из строя, её дальнейшая работа невозможна без полной замены или серьёзного вмешательства.

Надёжность невосстанавливаемых систем характеризуется вероятностью безотказной работы за определённый промежуток времени.

В отличие от восстанавливаемых систем, где учитывается и время восстановления, у невосстанавливаемых систем рассматривается только период безотказной работы.

Примером может служить, например, одноразовый датчик или устройство с ограниченным ресурсом, которое при отказе подлежит замене.

Для повышения надёжности таких систем применяют методы контроля качества на этапе производства, тщательный отбор компонентов и разработку резервных схем.

# **20. Классификация ошибок программного обеспечения**

Ошибки программного обеспечения — это неправильные действия или дефекты в коде, которые приводят к неправильной работе программы или её отказу.

Ошибки в ПО классифицируют по разным признакам:

1. По происхождению:

Синтаксические ошибки — нарушения правил языка программирования (например, пропущенная запятая, неправильное написание команды). Обычно выявляются на этапе компиляции.

Логические ошибки — неправильная логика работы программы, из-за чего она выполняет не те действия, которые ожидались. Эти ошибки сложно найти, они проявляются во время выполнения.

Ошибки проектирования — неправильная архитектура программы, приводящая к сбоям или неправильной работе.

1. По времени проявления:

Ошибки компиляции — обнаруживаются при сборке программы.

Ошибки выполнения (runtime errors) — проявляются во время работы программы, например, деление на ноль или выход за пределы массива.

Ошибки логики (semantic errors) — программа выполняется без сбоев, но даёт неправильный результат.

1. По влиянию на систему:

Критические ошибки — приводят к полному сбою или потере данных.

Некритические ошибки — влияют на отдельные функции, но система в целом работает.

1. По источнику:

Ошибки разработчика — допущены при написании кода.

Ошибки пользователя — вызваны неправильным использованием программы.

Ошибки окружения — вызваны проблемами в операционной системе, оборудовании или взаимодействии с другими программами.

# **26. Зависимость надёжности от времени**

Надёжность — это способность системы безотказно выполнять свои функции в течение определённого времени. Главная особенность — надёжность всегда зависит от времени.

Основная идея

* Чем дольше система работает, тем выше вероятность, что она столкнётся с отказом.
* Надёжность выражается через вероятность безотказной работы за время *t*: R(t)=P(система работает без отказа на [0,t])R(t) = P(\{система работает без отказа на } [0, t])R(t)=P(система работает без отказа на [0,t]).

Кривая отказов (или «кривая ванны»)

Надёжность системы меняется по времени в три этапа:

1. Начальный период (приработка, «детские болезни»)

Высокий уровень отказов из-за производственных дефектов или ошибок настройки.

Система проходит проверку, выявляются и устраняются слабые места.

1. Период нормальной эксплуатации

Отказы случаются редко и в основном случайным образом, вероятность отказа примерно постоянна.

Это наиболее стабильный этап работы.

1. Период старения и износа

Вероятность отказов начинает расти из-за износа компонентов, устаревания программного обеспечения, накопления ошибок.

Требуется обновление или ремонт.

Математическое описание

Надёжность R(t) обычно убывает по экспоненте или более сложным функциям, отражая рост вероятности отказа со временем.

Интенсивность отказов λ(t) показывает, насколько часто система выходит из строя в единицу времени.

# **29. Показатели надёжности вычислительных систем**

Показатели надёжности — это числовые характеристики, с помощью которых оценивают устойчивость и стабильность работы вычислительных систем (ВС).

Основные показатели надёжности вычислительных систем:

1. Вероятность безотказной работы (R(t)) — вероятность того, что система будет работать без отказа в течение времени t.
2. Среднее время наработки на отказ (MTTF — Mean Time To Failure) — среднее время работы системы до первого отказа. Чем больше, тем надёжнее система.
3. Среднее время восстановления (MTTR — Mean Time To Repair) — среднее время, необходимое для восстановления работоспособности после отказа.
4. Среднее время между отказами (MTBF — Mean Time Between Failures) — средний интервал между двумя последовательными отказами, равен сумме MTTF и MTTR.
5. Интенсивность отказов (λ — lambda) — частота отказов в единицу времени. Чем меньше λ, тем надёжнее система.
6. Наработка (работоспособность) — фактическое время работы системы без отказов.

# **30. Специфика информационной системы как объекта исследования надёжности**

Информационная система (ИС) — это сложный комплекс, включающий аппаратные средства, программное обеспечение, данные и людей, взаимодействующих для обработки и передачи информации.

Поэтому надёжность ИС имеет свои особенности:

1. Многоуровневая структура — ИС состоит из нескольких взаимосвязанных компонентов: оборудование, ПО, сети, базы данных, пользовательский интерфейс. Надёжность зависит от надёжности каждого уровня и их взаимодействия.
2. Комбинация технических и логических элементов — Отказы могут быть связаны не только с физическими неисправностями, но и с ошибками в программном обеспечении, неправильной обработкой данных, человеческим фактором.
3. Влияние программных ошибок и обновлений — ПО может содержать баги, которые проявляются со временем, а обновления системы могут как улучшать, так и снижать надёжность.
4. Значимость целостности и безопасности данных — Надёжность ИС включает сохранность и корректность данных, защиту от потерь и искажений.
5. Влияние внешних факторов — ИС зависит от условий эксплуатации: качество электропитания, сетевые атаки, человеческие ошибки при администрировании.
6. Возможность восстановления и резервирования — Важна способность ИС быстро восстанавливаться после сбоев, например, за счёт резервных копий, кластеров и аварийного переключения.