Disynergy
Страница материала ↗
EXQT
Готовые ответы — на EXQT
Доступ к ответам приобретается отдельно.
Найти ответы ↗

Методические рекомендации

Теоретические материалы по информационным системам, архитектуре, администрированию, мониторингу, обновлению и документированию, которые могут быть использованы для составления технического задания и организации сопровождения ИС.

1. Теоретические основы информационных систем

Описание:
Данные материалы могут быть использованы для составления технического задания и понимания роли информационных систем в деятельности организации.

1.1. Понятие информационной системы

Информационная система (ИС) — это комплекс взаимосвязанных элементов, предназначенных для сбора, хранения, обработки, передачи и предоставления информации пользователям в соответствии с установленными целями.

ИС обеспечивает автоматизацию деятельности организации, поддержку принятия управленческих решений, повышение эффективности и прозрачности бизнес-процессов.

Ключевыми признаками любой ИС являются:

- наличие входной информации (данных);
- выполнение процедур обработки (алгоритмы, программные модули);
- получение выходной информации (отчёты, документы, результаты вычислений);
- взаимодействие пользователей с системой.

1.2. Структура информационной системы

Современная ИС представляет собой многоуровневую архитектуру, включающую технические, программные, организационные и информационные компоненты.

Типовая структура включает:

1.2.1. Технические средства (аппаратное обеспечение)

- серверы приложений и баз данных;
- рабочие станции пользователей;
- сетевое оборудование (маршрутизаторы, коммутаторы, точки доступа);
- устройства хранения данных (NAS, SAN).

1.2.2. Программное обеспечение

Помогу с учебным заданием без предоплаты. Оплата после выполнения
Нажмите на мессенджер, чтобы написать
Telegram ↗ WhatsApp ↗ MAX ↗
+7 (924) 305-23-08 · Пишите в мессенджер
disynergy.ru

- системное ПО (операционные системы Windows/Linux);
- серверы приложений и веб-серверы;
- СУБД (MySQL, PostgreSQL, MS SQL Server);
- клиентские приложения, веб-интерфейсы, мобильные клиенты.

1.2.3. Информационные ресурсы

- базы данных и таблицы;
- электронные документы;
- файлы конфигурации, настройки;
- аналитические отчёты и хранилища данных.

1.2.4. Организационная инфраструктура

- пользователи и роли;
- регламенты работы;
- правила доступа и безопасности.

1.3. Жизненный цикл информационной системы

Жизненный цикл ИС — это совокупность этапов от возникновения идеи автоматизации до вывода системы из эксплуатации.

Классическая модель жизненного цикла включает следующие стадии:

1. **Инициация и анализ требований.**
   Определение целей автоматизации, сбор требований, выявление проблем в текущих процессах.
2. **Проектирование системы.**
   Разработка архитектуры, проектирование базы данных и модулей, моделирование процессов (UML, BPMN).
3. **Разработка и интеграция.**
   Создание программных модулей, настройка оборудования, интеграция подсистем.
4. **Тестирование и внедрение.**
   Проверка качества, обучение пользователей, запуск системы в рабочей среде.
5. **Эксплуатация и сопровождение.**
   Обновления, исправление ошибок, техническая поддержка.
6. **Вывод из эксплуатации.**
   Архивация данных, переход на новую ИС, демонтаж оборудования.

1.4. Типы информационных систем

На практике ИС классифицируют по назначению, масштабу и функциям. Основные типы:

1.4.1. Корпоративные информационные системы (КИС)

Помогу с учебным заданием без предоплаты. Оплата после выполнения
Нажмите на мессенджер, чтобы написать
Telegram ↗ WhatsApp ↗ MAX ↗
+7 (924) 305-23-08 · Пишите в мессенджер
disynergy.ru

Предназначены для комплексной автоматизации деятельности предприятия: ERP, CRM, SCM, HRM, бухгалтерские системы.

Характеризуются высокой масштабируемостью, распределённой структурой и модульностью.

1.4.2. Учётные информационные системы

Решают задачи регистрации, учёта, контроля и формирования отчётности. Примеры: складской учёт, бухгалтерский учёт, учёт кадров.

1.4.3. Сервисные и поддерживающие системы

Предоставляют услуги пользователям: электронные очереди, сервисы технической поддержки, системы мониторинга.

1.4.4. Обучающие и образовательные системы

Используются в обучении и развитии персонала: LMS (Moodle), системы тестирования, тренажёры, виртуальные лаборатории.

1.4.5. Web-системы

Системы, доступные через интернет-браузер: порталы, интернет-магазины, личные кабинеты, облачные сервисы.

1.5. Состав информационной системы

Структура ИС включает комплекс взаимосвязанных компонентов.

1.5.1. Аппаратное обеспечение

- Серверы приложений — выполняют логику системы.
- Серверы баз данных — обеспечивают работу СУБД.
- Рабочие станции — компьютеры пользователей.
- Сетевое оборудование — маршрутизаторы, коммутаторы, кабельная инфраструктура.
- Системы хранения данных — дисковые массивы, облачные хранилища.

1.5.2. Программное обеспечение

Состоит из:

1. **Системного ПО:**
   Windows Server, Linux, драйверы, средства виртуализации.
2. **Прикладного ПО:**
   сервер приложений (Apache, Nginx);
   сервер баз данных (PostgreSQL, MySQL, Oracle);
   клиентские программы (толстый, тонкий, мобильный клиент).

Помогу с учебным заданием без предоплаты. Оплата после выполнения
Нажмите на мессенджер, чтобы написать
Telegram ↗ WhatsApp ↗ MAX ↗
+7 (924) 305-23-08 · Пишите в мессенджер disynergy.ru

3. **Инструментального ПО:**
   средства разработки, тестирования, мониторинга.

1.5.3. Информационные ресурсы

- реляционные базы данных;
- документы и файлы, используемые в процессе работы;
- отчёты и аналитические выгрузки;
- справочники и классификаторы (номенклатура, контрагенты, пользователи).

1.5.4. Пользователи и роли

Информационная система всегда ориентирована на определённую группу пользователей. Основные категории:

- Администраторы — настройка системы, безопасность, резервное копирование.
- Операторы — ввод данных, обработка запросов, работа с документами.
- Конечные пользователи — использование функций ИС для решения профессиональных задач.
- Руководители — контроль, анализ, принятие управленческих решений.

Роли определяют:

- уровень доступа;
- разрешённые действия;
- ответственность пользователя.

1.6. Значение информационных систем в современном обществе

ИС являются ключевым элементом цифровой экономики. Они обеспечивают:

- повышение эффективности труда;
- ускорение принятия решений;
- прозрачность процессов;
- снижение ошибок;
- автоматизацию рутинных задач;
- анализ и прогнозирование на основе данных.

Использование ИС стало обязательным элементом деятельности государственных организаций, коммерческих предприятий, образовательных учреждений и сервисных компаний.

2. Архитектура и компоненты информационных систем

2.1. Понятие архитектуры информационной системы

Помогу с учебным заданием без предоплаты. Оплата после выполнения
Нажмите на мессенджер, чтобы написать
Telegram ↗ WhatsApp ↗ MAX ↗
+7 (924) 305-23-08 · Пишите в мессенджер
disynergy.ru

Архитектура информационной системы — это структурная организация всех её компонентов и способы их взаимодействия. Архитектура определяет, как распределяются функции приложения, где выполняется обработка данных, какими средствами обеспечивается доступ пользователей к функционалу и как компоненты обмениваются информацией.

Архитектура влияет на:

- надёжность и отказоустойчивость;
- масштабируемость и производительность;
- удобство сопровождения;
- безопасность и защиту данных.

2.2. Уровни архитектуры информационной системы

Современные ИС используют многоуровневую (многослойную) модель. Основные уровни:

2.2.1. Клиентский уровень (Frontend)

Выполняется на устройстве пользователя — ПК, планшете, смартфоне, терминале. Сюда относятся:

- браузеры (Chrome, Firefox, Edge);
- desktop-клиенты;
- мобильные приложения;
- интерфейсы операторов.

Основные функции клиента:

- отображение графического интерфейса (GUI);
- обработка пользовательского ввода;
- формирование запросов к серверу;
- визуализация результатов (страницы, таблицы, отчёты).

2.2.2. Серверный уровень (Backend)

Расположен на сервере приложения и отвечает за:

- выполнение бизнес-логики;
- валидацию данных;
- обработку HTTP-запросов;
- управление процессами, очередями, уведомлениями;
- работу с внешними сервисами (почта, платежи, API).

Используемые технологии:

- веб-серверы: Nginx, Apache;
- серверы приложений: Node.js, .NET Core, Java Spring;
- API: REST, GraphQL, SOAP.

Помогу с учебным заданием без предоплаты. Оплата после выполнения
Нажмите на мессенджер, чтобы написать
Telegram ↗ WhatsApp ↗ MAX ↗
+7 (924) 305-23-08 · Пишите в мессенджер
disynergy.ru

2.2.3. Уровень данных (база данных)

Представляет собой центральное хранилище обработанной информации. Основные типы:

- реляционные СУБД: MySQL, PostgreSQL, Oracle;
- нереляционные СУБД: MongoDB, Redis;
- хранилища данных: Data Warehouse, OLAP-кубы.

Функции уровня данных:

- выполнение SQL-запросов;
- хранение структурированной и неструктурированной информации;
- обеспечение целостности, безопасности, резервного копирования.

2.3. Принципы взаимодействия компонентов системы

Для функционирования ИС необходимо организованное взаимодействие между её уровнями.

2.3.1. HTTP-запросы

Применяются в web-системах для обмена данными по протоколу HTTP/HTTPS. Типовые методы:

- GET — получение данных;
- POST — отправка, создание;
- PUT/PATCH — обновление;
- DELETE — удаление.

2.3.2. API (Application Programming Interface)

API — это интерфейс взаимодействия между программами. Разновидности:

- REST API — передача данных в формате JSON;
- GraphQL — запрос только нужных полей;
- SOAP — XML-сообщения и строгие схемы;
- внутренние API — взаимодействие микросервисов.

Роль API:

- стандартизирует обмен данными;
- упрощает интеграцию систем;
- позволяет использовать сервис как модуль в других приложениях.

2.3.3. SQL-запросы

SQL — язык управления данными в реляционных базах. Основные типы операций:

- DML (манипуляции данными): SELECT, INSERT, UPDATE, DELETE;
- DDL (структура данных): CREATE, ALTER, DROP;
- DCL (доступ): GRANT, REVOKE.

Помогу с учебным заданием без предоплаты. Оплата после выполнения
Нажмите на мессенджер, чтобы написать
Telegram ↗ WhatsApp ↗ MAX ↗
+7 (924) 305-23-08 · Пишите в мессенджер
disynergy.ru

Сервер приложения формирует SQL-запросы и обращается к БД через драйверы и ORM.

2.4. Виды архитектур информационных систем

Архитектура информационного приложения может быть построена по разным моделям.

2.4.1. Монолитная архитектура

Все модули системы объединены в одном приложении:

- общий код;
- единая база данных;
- запуск как одного процесса.

Плюсы:

- простота разработки и развертывания;
- логическая целостность;
- единая точка управления.

Минусы:

- сложность масштабирования отдельных частей;
- трудность внесения изменений в больших системах;
- зависимость всех модулей друг от друга.

Используется в небольших проектах и учебных системах.

2.4.2. Клиент-серверная архитектура

Система разделена на:

1. клиент (GUI),
2. сервер приложений,
3. базу данных.

Характеристики:

- сервер хранит и обрабатывает данные;
- клиент исполняет интерфейс и отправляет запросы;
- сервер авторизует пользователя и формирует ответы.

Используется в большинстве корпоративных и web-систем.

2.4.3. Микросервисная архитектура

Система разделена на независимые модули — микросервисы. Каждый микросервис:

- выполняет одну бизнес-функцию;
- имеет свой код;

Помогу с учебным заданием без предоплаты. Оплата после выполнения
Нажмите на мессенджер, чтобы написать
Telegram ↗ WhatsApp ↗ MAX ↗
+7 (924) 305-23-08 · Пишите в мессенджер disynergy.ru

- может иметь собственную БД;
- общается с другими сервисами через API.

Преимущества:

- горизонтальное масштабирование;
- независимые релизы и обновления;
- отказоустойчивость.

Недостатки:

- сложность настройки инфраструктуры;
- высокий уровень интеграции;
- необходимость DevOps-практик.

Используется в крупных цифровых платформах.

2.5. Интерфейсы взаимодействия

Система предоставляет пользователю доступ к функционалу через различные интерфейсы.

2.5.1. Графический пользовательский интерфейс (GUI)

Используется в desktop-приложениях. Состоит из окон, кнопок, форм, справочников.

2.5.2. Web-интерфейс

Работает в браузере. Особенности:

- не требует установки программы;
- доступен с любых устройств;
- использует HTML, CSS, JavaScript.

2.5.3. Программный интерфейс (API)

Предоставляет доступ к функциям приложения другим программам:

- мобильным приложениям;
- внешним сервисам;
- микросервисам системы.

API обеспечивает автоматизацию и интеграцию.

3. Администрирование и конфигурирование информационных систем

Администрирование информационных систем — это совокупность действий по настройке, поддержке, мониторингу и обеспечению стабильной работы операционных систем, программных сервисов и инфраструктуры предприятия.

Помогу с учебным заданием без предоплаты. Оплата после выполнения
Нажмите на мессенджер, чтобы написать
Telegram ↗ WhatsApp ↗ MAX ↗
+7 (924) 305-23-08 · Пишите в мессенджер
disynergy.ru

Специалист по администрированию отвечает за доступность ресурсов, безопасность данных, корректную конфигурацию систем и устранение неисправностей.

3.1. Основы настройки операционных систем

3.1.1. Настройка Windows

Администрирование ОС Windows включает:

- управление локальными и доменными пользователями (Local Users & Groups, Active Directory);
- настройку сетевых параметров: IP-адрес, DNS, маршрутизация;
- конфигурацию политик безопасности (Local Security Policy, Group Policy);
- установку и обновление драйверов и приложений;
- мониторинг работоспособности через «Диспетчер задач», «Просмотр событий», «Службы» (services.msc);
- управление автозапуском, службами, правами доступа к файловой системе (NTFS ACL).

Основные административные инструменты:

- PowerShell, cmd (консольные команды);
- MMC-консоли (eventvwr.msc, compmgmt.msc, gpedit.msc);
- Server Manager (в серверных редакциях).

3.1.2. Настройка Linux

Администрирование Linux основано на файловой структуре, текстовых конфигурациях и терминале. Ключевые аспекты:

- управление пакетами: apt, yum, dnf, snap;
- конфигурация сети (файлы /etc/network, nmcli, ip addr);
- настройка прав доступа (chmod, chown, groups);
- обновление системы (apt upgrade, dnf update);
- управление системными сервисами (systemctl, service);
- мониторинг журналов (journalctl, /var/log/syslog).

Важные каталоги:

- /etc — конфигурационные файлы;
- /var — журналы, временные данные;
- /home — домашние каталоги пользователей.

3.2. Управление пользователями и правами доступа

3.2.1. Роли пользователей

Администратор ИС должен уметь:

Помогу с учебным заданием без предоплаты. Оплата после выполнения
Нажмите на мессенджер, чтобы написать
Telegram ↗ WhatsApp ↗ MAX ↗
+7 (924) 305-23-08 · Пишите в мессенджер disynergy.ru

- создавать, изменять и удалять учётные записи;
- назначать роли и принадлежность к группам;
- настраивать уровни доступа.

Пользователи делятся на:

- администраторов — обслуживают системы;
- операторов — выполняют операции;
- обычных пользователей — работают в приложениях;
- сервисных пользователей — созданы для работы служб.

3.2.2. Права доступа и безопасность

В Windows:

Используется модель Discretionary Access Control (DAC) с ACL:

- право чтения (Read), записи (Write), выполнения (Execute);
- наследование прав;
- политики паролей, ограничений входа, контроль учетных записей (UAC).

В Linux:

Используются права:

- r (read), w (write), x (execute);
- три категории: owner, group, others;
- изменение — chmod, chown, usermod, groupadd.

Примеры:

```
chmod 755 file.sh
chown admin:staff report.txt
```

Безопасность обеспечивается:

- политиками SELinux/AppArmor,
- настройкой sudo-доступа,
- аудитом входов и действий пользователей.

3.3. Назначение и настройка служб и демонов

Службы (Windows Services) и демоны (Linux daemons) обеспечивают непрерывную работу серверных приложений.

3.3.1. Web-серверы

Примеры:

- Apache HTTP Server;

Помогу с учебным заданием без предоплаты. Оплата после выполнения
Нажмите на мессенджер, чтобы написать
Telegram ↗ WhatsApp ↗ MAX ↗
+7 (924) 305-23-08 · Пишите в мессенджер
disynergy.ru

- Nginx;
- IIS (Windows).

Функции:

- обработка HTTP-запросов;
- раздача веб-страниц;
- работа со скриптами (PHP, Python, ASP.NET).

Основные параметры настройки:

- виртуальные хосты;
- корневой каталог;
- SSL-сертификаты;
- журналы ошибок и доступа.

3.3.2. Сервера баз данных

Примеры: MySQL, PostgreSQL, MS SQL Server, Oracle, MongoDB.

Администратор должен уметь:

- конфигурировать параметры БД (порты, пользователя root, каталоги данных);
- создавать базы и пользователей;
- настраивать резервное копирование и восстановление;
- контролировать производительность (индексы, блокировки, запросы).

3.3.3. Почтовые сервисы

Примеры:

- Postfix, Exim (Linux);
- Microsoft Exchange Server (Windows).

Функции:

- отправка и приём почты;
- маршрутизация сообщений;
- управление почтовыми ящиками и доменами;
- защита (SPF, DKIM, DMARC).

3.3.4. Управление службами

В Windows:

- просмотр: services.msc;
- консоль: sc query, net start, net stop.

В Linux (systemd):

Помогу с учебным заданием без предоплаты. Оплата после выполнения
Нажмите на мессенджер, чтобы написать
Telegram ↗ WhatsApp ↗ MAX ↗
+7 (924) 305-23-08 · Пишите в мессенджер disynergy.ru

```
systemctl status nginx
systemctl start mysql
systemctl enable ssh
systemctl restart apache2
```

3.4. Принципы работы с конфигурационными файлами

Конфигурационные файлы определяют параметры работы сервисов, модулей и приложений. Основные форматы:

3.4.1. INI (.ini)

Структура: разделы + параметры. Применение: Windows-приложения, лёгкие настройки.

```
[server]
port=8080
mode=prod
```

3.4.2. JSON (.json)

Структурированный формат данных, часто используется в web-приложениях.

```
{
  "database": {
    "host": "localhost",
    "port": 5432
  }
}
```

3.4.3. YAML (.yaml)

Удобен для DevOps и конфигурации сервисов: Docker, Kubernetes, Ansible.

```
server:
  port: 8000
  debug: false
```

3.4.4. XML (.xml)

Строгий, расширяемый, используется в крупных системах и стандартах.

```
<config>
    <service enabled="true">
        <port>443</port>
```

Помогу с учебным заданием без предоплаты. Оплата после выполнения
Нажмите на мессенджер, чтобы написать
Telegram ↗ WhatsApp ↗ MAX ↗
+7 (924) 305-23-08 · Пишите в мессенджер disynergy.ru

```
    </service>
</config>
```

3.4.5. Общие принципы работы

- хранение конфигураций в /etc (Linux) или ProgramData (Windows);
- разделение рабочих и шаблонных файлов;
- резервное копирование перед изменениями;
- использование комментариев;
- проверка синтаксиса (nginx -t, jsonlint, yamllint).

3.5. Базовые команды для проверки состояния системы

Администратор должен оперативно оценивать состояние ОС, сети и служб.

3.5.1. Сетевые команды

Универсальные:

- ping — проверка доступности узла;
- tracert / traceroute — маршрут до узла;
- nslookup / dig — DNS-запросы.

Windows:

- ipconfig, netstat, arp -a, net use, tasklist.

Linux:

- ip addr, ifconfig;
- netstat, ss;
- route, iptables;
- journalctl -u service.

3.5.2. Команды управления процессами

Windows:

- tasklist, taskkill.

Linux:

- ps aux, top, htop, kill, pkill.

3.5.3. Управление службами

Windows:

- services.msc, sc stop, sc start.

Linux:

Помогу с учебным заданием без предоплаты. Оплата после выполнения
Нажмите на мессенджер, чтобы написать
Telegram ↗ WhatsApp ↗ MAX ↗
+7 (924) 305-23-08 · Пишите в мессенджер disynergy.ru

- systemctl status, systemctl restart, systemctl enable.

4. Мониторинг и диагностика

Мониторинг и диагностика являются ключевыми процессами обеспечения стабильной, безопасной и предсказуемой работы информационных систем. Они позволяют своевременно выявлять неисправности, оценивать нагрузку, анализировать работоспособность инфраструктуры и предотвращать сбои.

4.1. Основные методы контроля состояния системы

Современные ИС требуют постоянного наблюдения за ресурсами и состоянием сервисов. Контроль осуществляется через:

- системные журналы (логи);
- ресурсные метрики производительности;
- диагностику сетевой доступности;
- проверку работоспособности сервисов и компонентов.

4.2. Использование логов

Логи — это текстовые файлы, в которых фиксируются события операционной системы, сервисов и приложений. Они — первый источник информации при диагностике ошибок.

4.2.1. Система логирования в Windows

Имеет централизованный инструмент: **Event Viewer (Просмотр событий)**. Журналы событий:

- System — работа драйверов, служб, сети;
- Application — ошибки приложений;
- Security — попытки входа, аудит;
- Setup — установка и обновления системы.

Администратор должен уметь:

- фильтровать события по уровню критичности (Error, Warning, Information);
- искать причину сбоя по EventID;
- отслеживать поведение служб.

4.2.2. Логи в Linux

Журналы файловой системы:

- /var/log/syslog — общая системная информация;
- /var/log/auth.log — входы пользователей;
- /var/log/kern.log — работа ядра;
- /var/log/nginx/ — логи веб-сервера;

Помогу с учебным заданием без предоплаты. Оплата после выполнения
Нажмите на мессенджер, чтобы написать
Telegram ↗ WhatsApp ↗ MAX ↗
+7 (924) 305-23-08 · Пишите в мессенджер disynergy.ru

- /var/log/mysql/ — логи БД.

Современный Linux использует systemd-журналы: `journalctl`

```
journalctl -xe       # последние ошибки
journalctl -u nginx  # логи конкретной службы
journalctl --since "1 hour ago"
```

4.2.3. Syslog

Syslog — универсальная система логирования, применяемая:

- сетевыми устройствами (коммутаторы, маршрутизаторы);
- многими серверами Linux;
- SIEM-системами.

Позволяет хранить логи централизованно и анализировать поведение всей сети.

4.3. Метрики производительности

Мониторинг ресурсов позволяет понять, насколько эффективно работает система, и выявить узкие места.

4.3.1. CPU (центральный процессор)

Контролируются:

- загрузка процессора (%);
- количество активных процессов;
- время процессорных очередей.

Высокая нагрузка может указывать на:

- зависший процесс;
- проблемы в бизнес-логике;
- перегруз веб-сервера.

4.3.2. RAM (оперативная память)

Следят за:

- объёмом используемой памяти;
- свободной памятью;
- использованием swap (подкачки).

Переполнение RAM приводит к:

- замедлению системы;
- росту использования SWAP;
- сбоям в программах.

Помогу с учебным заданием без предоплаты. Оплата после выполнения
Нажмите на мессенджер, чтобы написать
Telegram ↗ WhatsApp ↗ MAX ↗
+7 (924) 305-23-08 · Пишите в мессенджер disynergy.ru

4.3.3. Disk (диски и файловая система)

Метрики:

- скорость чтения/записи;
- доступное место;
- задержка дисковых операций (IO wait).

Высокий IO wait = медленные диски → задержки в работе БД.

4.3.4. Network (сеть)

Контролируются:

- пропускная способность;
- количество сетевых ошибок;
- задержка и потери пакетов;
- скорость входящего/исходящего трафика.

4.4. Понятия uptime, latency и availability

4.4.1. Uptime (время непрерывной работы)

Показывает, сколько система работает без перезагрузки. В Linux:

```
uptime
```

4.4.2. Latency (задержка запроса)

Время отклика системы:

- измеряется ping-тестами, API-бенчмарками;
- критична для web-приложений и БД.

4.4.3. Availability (доступность)

Показатель того, насколько часто сервис доступен пользователям. Вычисляется как процент времени, в течение которого система работала.

Примеры:

- 99% — 3,65 дня простоя в год;
- 99.9% — 8,7 часа простоя;
- 99.99% — 52 минуты.

4.5. Методы поиска и устранения неисправностей

Процесс диагностики включает несколько последовательных этапов.

Помогу с учебным заданием без предоплаты. Оплата после выполнения
Нажмите на мессенджер, чтобы написать
Telegram ↗ WhatsApp ↗ MAX ↗
+7 (924) 305-23-08 · Пишите в мессенджер disynergy.ru

4.5.1. Анализ логов

Первый и основной способ выяснить:

- причину аварии;
- время возникновения;
- какой модуль вызвал проблему.

4.5.2. Трассировка (Tracing)

Позволяет отслеживать путь сетевого пакета или запросов. Инструменты:

- tracert (Windows) / traceroute (Linux);
- трассировка API через Postman / curl;
- трассировка зависимостей в микросервисах (Jaeger, Zipkin).

Используется, если:

- сервис недоступен;
- реквест «не доходит» до сервера;
- проблема в маршрутизации.

4.5.3. Тестирование соединений

Команды:

- ping — базовая доступность;
- telnet host port — проверка порта;
- curl / wget — тестирование HTTP;
- nslookup, dig — проверка DNS.

4.5.4. Локализация проблемы

При диагностике учитывают:

1. это ошибка ОС?
2. ошибка сети?
3. ошибка приложения?
4. ошибка пользователя?

Используются инструменты:

- Диспетчер задач (Windows), top/htop (Linux);
- systemctl для проверки статуса сервиса;
- du, df для проверки дисков.

4.5.5. Устранение неисправностей

Помогу с учебным заданием без предоплаты. Оплата после выполнения
Нажмите на мессенджер, чтобы написать
Telegram ↗ WhatsApp ↗ MAX ↗
+7 (924) 305-23-08 · Пишите в мессенджер disynergy.ru

Типовые действия:

- перезапуск службы;
- удаление зависшего процесса;
- освобождение дискового пространства;
- восстановление из бэкапа;
- проверка конфигурационных файлов;
- откат обновления.

4.6. Тестирование работоспособности после обновлений

Обновления ПО могут нарушать функциональность, поэтому требуется проверка.

4.6.1. Smoke testing

Быстрая проверка базовой работоспособности системы:

- запускается ли сервис?
- доступны ли ключевые функции?
- нет ли критических ошибок?

Цель — убедиться, что система «дышит».

4.6.2. Regression testing

Проверка того, что после внесения изменений:

- не нарушилась старая функциональность;
- все ранее работающие функции работают корректно.

Регрессионные тесты предотвращают повторное появление ошибок.

5. Обновление и сопровождение информационных систем

Сопровождение и обновление программного обеспечения — один из ключевых этапов жизненного цикла информационной системы. Этот этап обеспечивает корректную работу, стабильность, безопасность и актуальность программных решений.

5.1. Этапы жизненного цикла программного обеспечения

После разработки система не считается завершённой — она должна поддерживаться, обновляться, адаптироваться к изменениям инфраструктуры, законодательства и бизнес-процессов.

Классическая схема завершающих этапов жизненного цикла ПО:

5.1.1. Внедрение

Этап ввода системы в эксплуатацию:

Помогу с учебным заданием без предоплаты. Оплата после выполнения
Нажмите на мессенджер, чтобы написать
Telegram ↗ WhatsApp ↗ MAX ↗
+7 (924) 305-23-08 · Пишите в мессенджер disynergy.ru

- установка и настройка программного обеспечения;
- обучение пользователей и администраторов;
- перенос данных (миграция);
- первичное тестирование в рабочей среде;
- адаптация под реальные процессы организации.

5.1.2. Сопровождение

Период, в котором система используется, контролируется и корректируется. Основные задачи сопровождения:

- мониторинг работоспособности и производительности;
- диагностика и исправление ошибок;
- техническая поддержка пользователей;
- установка обновлений и патчей;
- регулярное резервное копирование данных;
- поддержка безопасности.

5.1.3. Модернизация

Комплекс мероприятий по развитию и расширению возможностей ИС. Включает:

- переработку архитектуры;
- добавление новых функций;
- оптимизацию производительности;
- улучшение интерфейсов и UX;
- переход на новые версии библиотек, платформ и оборудования.

Модернизация направлена на повышение эффективности, масштабируемости и конкурентоспособности системы.

5.2. Различие между обслуживанием и модернизацией

В практике эксплуатации ИС часто путают два процесса — обслуживание (maintenance) и модернизацию (upgrade/enhancement). Их различие принципиально.

5.2.1. Обслуживание (сопровождение)

Цель: поддерживать корректную работу уже существующей функциональности.

Характерные виды работ:

- устранение ошибок (bugfix);
- обновление справочников и конфигураций;
- обеспечение совместимости;
- оптимизация без изменения логики;

Помогу с учебным заданием без предоплаты. Оплата после выполнения
Нажмите на мессенджер, чтобы написать
Telegram ↗ WhatsApp ↗ MAX ↗
+7 (924) 305-23-08 · Пишите в мессенджер disynergy.ru

- техническое обслуживание оборудования;
- установка мелких корректировок.

Обслуживание не меняет архитектуру и логику работы системы, а только поддерживает стабильность.

5.2.2. Модернизация

Цель: улучшить, изменить или расширить систему.

Типичные изменения:

- добавление новых модулей;
- переработка интерфейса;
- переход с монолита на микросервисы;
- обновление СУБД или языка программирования;
- интеграция с новыми сервисами;
- изменение бизнес-процессов.

Модернизация изменяет функциональность или архитектуру и приводит к появлению новой версии ПО.

5.3. Принципы установки патчей, обновлений и исправлений

Обновления — это важнейший инструмент повышения безопасности и стабильности программного обеспечения.

5.3.1. Типы обновлений

5.3.1.1. Release (релиз)

Крупное обновление:

- новые функции;
- улучшения архитектуры;
- смена версии (например: 2.0 → 2.1).

Обычно сопровождается документацией, тестированием и обновлением пользовательских инструкций.

5.3.1.2. Hotfix

Экстренное исправление критической ошибки:

- устанавливается оперативно;
- не несёт новых функций;
- направлено на устранение сбоя или уязвимости.

5.3.1.3. Patch (патч)

Небольшое исправление или улучшение:

- корректировка ошибок;

Помогу с учебным заданием без предоплаты. Оплата после выполнения
Нажмите на мессенджер, чтобы написать
Telegram ↗ WhatsApp ↗ MAX ↗
+7 (924) 305-23-08 · Пишите в мессенджер disynergy.ru

- улучшение стабильности;
- обновление модулей или библиотек.

Может распространяться автоматически (Windows Update, Linux repositories).

5.3.2. Принципы безопасного обновления

Перед обновлением:

- создавать резервные копии данных и конфигураций;
- проверять совместимость новой версии с зависимостями;
- изучать changelog и документацию;
- тестировать на стенде (pre-production).

Во время обновления:

- соблюдать порядок операций (сначала БД, потом сервер);
- использовать режимы обслуживания (maintenance mode);
- вести логи операций.

После обновления:

- выполнять smoke-тестирование;
- проверять журнал ошибок;
- информировать пользователей о возможных изменениях.

5.4. Документирование процесса обновления

Правильное документирование позволяет отслеживать изменения, выявлять проблемы и поддерживать прозрачность эксплуатации ИС. Документы, которые должен вести администратор:

5.4.1. Журнал изменений (Changelog)

Содержит информацию:

- что было изменено;
- какие ошибки исправлены;
- какие модули обновлены;
- версии компонентов;
- дата и время обновления.

Пример:

```
Версия: 3.1.4
Дата: 14.11.2025
Ответственный: Иванов И.И.
Изменения:
- Исправлен сбой авторизации (bug #458)
```

Помогу с учебным заданием без предоплаты. Оплата после выполнения
Нажмите на мессенджер, чтобы написать
Telegram ↗ WhatsApp ↗ MAX ↗
+7 (924) 305-23-08 · Пишите в мессенджер disynergy.ru

```
- Обновлена библиотека React до 18.2
- Оптимизирована работа модуля отчетов
```

5.4.2. Учёт версий

Используется семантическое версионирование:
`MAJOR.MINOR.PATCH`

- MAJOR — крупные изменения, несовместимости;
- MINOR — новые функции, совместимые изменения;
- PATCH — исправления без изменения логики.

Пример: `2.5.13`

5.4.3. Документация обновлений

Обязательно фиксируются:

- причина обновления;
- описание выполненных действий;
- версии до и после обновления;
- инструкции по откату;
- сведения о тестировании после обновления.

5.4.4. Ответственный за обновление

Назначается сотрудник:

- который выполнял работы;
- фиксирует результаты в журнале;
- подписывает документацию;
- предоставляет отчёт руководителю или заказчику.

6. Документирование и отчётность при сопровождении информационных систем

Документирование — ключевой элемент сопровождения информационных систем. Грамотно оформленная документация позволяет обеспечивать стабильную эксплуатацию, передавать систему в поддержку новым специалистам, фиксировать изменения и соблюдать требования безопасности и регламентов организации.

Документы сопровождения ИС описывают структуру системы, порядок её обслуживания, правила обновления, план восстановления и результаты выполненных работ.

6.1. Структура технической документации по сопровождению ИС

Помогу с учебным заданием без предоплаты. Оплата после выполнения
Нажмите на мессенджер, чтобы написать
Telegram ↗ WhatsApp ↗ MAX ↗
+7 (924) 305-23-08 · Пишите в мессенджер disynergy.ru

Техническая документация оформляется согласно ГОСТам, внутренним регламентам организации и требованиям безопасности. К основным документам относятся:

1. паспорт системы;
2. журнал сопровождения;
3. инструкция администратора;
4. план резервного копирования;
5. отчёты об обновлениях, работах и модернизациях.

Рассмотрим каждый документ подробнее.

6.2. Паспорт системы

Паспорт информационной системы — это документ, содержащий сводные сведения о системе, её назначении, составе, конфигурации и правилах эксплуатации.

Типовое содержание паспорта ИС:

- наименование системы и её назначение;
- владелец ИС и ответственные лица;
- состав аппаратного обеспечения;
- версия программного обеспечения и его компоненты;
- архитектурная схема (логическая и физическая);
- перечень подсистем и ключевых модулей;
- сведения о базе данных (структура, местоположение, СУБД);
- интеграции с другими сервисами;
- используемые порты, протоколы и сетевые параметры;
- правила эксплуатации и ограничения;
- дата ввода в эксплуатацию;
- сведения о текущей версии и последних обновлениях.

Паспорт используется:

- администраторами;
- аудиторами;
- специалистами при передаче системы в сопровождение;
- в составе документации при аттестации и проверках безопасности.

6.3. Журнал сопровождения

Журнал сопровождения — документ, фиксирующий все операции, связанные с обслуживанием ИС.

В журнал заносится:

- дата выполнения работы;
- время начала и завершения;
- описание операции (настройка, исправление, обновление);

Помогу с учебным заданием без предоплаты. Оплата после выполнения
Нажмите на мессенджер, чтобы написать
Telegram ↗ WhatsApp ↗ MAX ↗
+7 (924) 305-23-08 · Пишите в мессенджер disynergy.ru

- ответственное лицо;
- результат и примечания;
- ссылки на сопутствующие документы (отчёт, тикет, скриншоты).

Журнал может вестись:

- в Excel / Google Sheets;
- в корпоративных системах (Jira, ServiceNow, 1С:ИТС);
- в виде физического документа.

Пример записи в журнале сопровождения

Дата	Операция	Описание	Исполнитель	Результат
12.11.2025	Устранение ошибки	Перезапуск MySQL, очистка блокировок	Петров П.П.	Работа восстановлена

Журнал позволяет:

- отслеживать изменения;
- контролировать качество сопровождения;
- диагностировать повторяющиеся проблемы.

6.4. Инструкция администратора

Документ описывает порядок администрирования системы, её компонентов и инфраструктуры.

Содержание инструкции администратора:

- требования к оборудованию и программному окружению;
- порядок запуска и остановки сервисов;
- управление пользователями и ролями;
- правила изменения конфигурационных файлов;
- порядок резервного копирования и восстановления;
- мониторинг состояния (service status, логи, метрики);
- периодические операции (архивация, очистка логов);
- типовые сценарии устранения неисправностей;
- чек-листы обновлений и проверок работоспособности.

Инструкция должна быть актуальной и регулярно обновляться.

6.5. План резервного копирования

Резервное копирование — неотъемлемая часть сопровождения ИС. План определяет:

- что резервируется (БД, конфиги, документы, логи);
- где хранятся копии (локально, облако, сетевые NAS/SAN);
- как часто создаются копии (ежедневно, еженедельно, инкрементно);
- методы копирования (полное, инкрементальное, дифференциальное);

Помогу с учебным заданием без предоплаты. Оплата после выполнения
Нажмите на мессенджер, чтобы написать
Telegram ↗ WhatsApp ↗ MAX ↗
+7 (924) 305-23-08 · Пишите в мессенджер disynergy.ru

- процедура восстановления системы;
- проверка целостности бэкапов;
- ответственные лица.

Образец структуры плана:

1. Цели и задачи резервного копирования;
2. Критически важные данные;
3. Виды и расписание копирования;
4. Описание используемого ПО (rsync, Veeam, Acronis, pg_dump, mysqldump);
5. Политика хранения копий;
6. Порядок восстановления;
7. Ответственные сотрудники;
8. Журнал выполненных резервных копий.

6.6. Отчёт об обновлении

Документ фиксирует факт выполнения обновления и результаты тестирования. Стандартное содержимое отчёта:

- дата выполнения обновления;
- система, модуль или сервер;
- старые и новые версии ПО;
- цель обновления (исправление ошибки, обновление безопасности, релиз новой версии);
- перечень выполненных действий;
- резервные копии, созданные перед обновлением;
- результаты smoke-testing и проверок;
- список возникших проблем и способов их устранения;
- фамилия и должность исполнителя.

Пример фрагмента:

```
Отчёт об обновлении №2025-11-14
Система: Сервис бронирования билетов
Версия до: 1.4.12
Версия после: 1.5.0
Изменения: внедрение нового API оплаты, исправления безопасности.
Дата: 14.11.2025
Исполнитель: Смирнова А.А.
Статус: выполнено успешно
```

Отчёт хранится вместе с журналом сопровождения и changelog.

Помогу с учебным заданием без предоплаты. Оплата после выполнения
Нажмите на мессенджер, чтобы написать
Telegram ↗ WhatsApp ↗ MAX ↗
+7 (924) 305-23-08 · Пишите в мессенджер
disynergy.ru

2. Выполнение задания

Разработка технического задания (ТЗ) на создание и сопровождение информационной системы в соответствии с требованиями ГОСТ 34.602-2020.

2.1. Описание задания

Необходимо разработать техническое задание (ТЗ) на создание и сопровождение информационной системы в соответствии с требованиями ГОСТ 34.602-2020.

- В документе следует определить назначение, цели и функции системы, описать объект автоматизации, сформулировать функциональные, информационные, технические и эксплуатационные требования, а также предусмотреть раздел, регламентирующий процессы сопровождения, обновления и администрирования системы.
- Результатом выполнения является структурированный документ ТЗ (в формате DOCX или PDF) с приложениями — схемой архитектуры системы, диаграммой взаимодействия модулей и описанием процедур сопровождения, оформленный по нормам ГОСТ 34.

2.2. Основные задачи

Студент должен:

1. Изучить структуру и содержание технического задания по ГОСТ 34.602-2020.
2. Определить назначение, цели и функции проектируемой информационной системы (ИС).
3. Описать состав, структуру и основные характеристики системы.
4. Определить требования к функциям сопровождения и администрирования.
5. Разработать текст ТЗ для выбранной предметной области.
6. Подготовить схему жизненного цикла ИС с указанием стадии сопровождения.

2.3. Содержание задания

2.3.1. Выбор предметной области

Выберите реальную или учебную информационную систему, для которой требуется сопровождение.

Примеры:

- Информационная система (ИС) учёта оборудования в лаборатории колледжа;

Помогу с учебным заданием без предоплаты. Оплата после выполнения
Нажмите на мессенджер, чтобы написать
Telegram ↗ WhatsApp ↗ MAX ↗
+7 (924) 305-23-08 · Пишите в мессенджер
disynergy.ru

- ИС бронирования билетов театра;
- ИС учёта заявок службы поддержки;
- ИС ведения базы студентов и успеваемости.

2.3.2. Изучение структуры ГОСТ 34.602-2020

ГОСТ устанавливает обязательные разделы ТЗ, включая следующие элементы.

Обязательные разделы ТЗ по ГОСТ 34.602-2020

№ раздела	Название	Краткое содержание
1	Общие сведения	Наименование системы, основания для разработки, исполнители, сроки.
2	Назначение и цели создания системы	Для чего предназначена ИС, какую задачу решает.
3	Характеристика объекта автоматизации	Описание текущих процессов, пользователей, потоков данных.
4	Требования к системе	Функциональные, информационные, технические, программные, эргономические и эксплуатационные требования, требования к защите информации.
5	Состав и содержание работ по созданию системы	Этапы, исполнители, контрольные точки.
6	Порядок контроля и приёмки системы	Виды испытаний, критерии качества, документация.
7	Требования к сопровождению системы	Периодическое обновление, резервное копирование, ведение журнала инцидентов, регламенты обслуживания.
8	Приложения	Диаграммы, схемы, таблицы, форматы данных, UML/PlantUML-модели.

Структура разделов ТЗ в текстовом виде:

1. **Общие сведения**
   (наименование системы, основания для разработки, исполнители, сроки).
2. **Назначение и цели создания системы**
   (для чего предназначена ИС, какую задачу решает).
3. **Характеристика объекта автоматизации**
   (описание текущих процессов, пользователей, потоков данных).
4. **Требования к системе**
   функциональные;
   информационные;
   технические;

Помогу с учебным заданием без предоплаты. Оплата после выполнения
Нажмите на мессенджер, чтобы написать
Telegram ↗ WhatsApp ↗ MAX ↗
+7 (924) 305-23-08 · Пишите в мессенджер disynergy.ru

программные;
эргономические и эксплуатационные;
требования к защите информации.

5. **Состав и содержание работ по созданию системы**
   (этапы, исполнители, контрольные точки).
6. **Порядок контроля и приёмки системы**
   (виды испытаний, критерии качества, документация).
7. **Требования к сопровождению системы**
   (периодическое обновление, резервное копирование, ведение журнала инцидентов, регламенты обслуживания).
8. **Приложения**
   (диаграммы, схемы, таблицы, форматы данных, UML/PlantUML-модели).

2.3.3. Разработка текстового документа

Подготовьте документ «Техническое задание на разработку и сопровождение информационной системы».

Структура (пример):

Техническое задание
на разработку и сопровождение информационной системы

(примерная структура по ГОСТ 34.602-2020)

1. Общие сведения

Наименование: «Информационная система учёта оборудования».
Основание для разработки: учебная практика ПМ.06.
Исполнители: студенты группы ___, руководитель ___.

2. Назначение и цели создания системы

Система предназначена для автоматизации учёта, обслуживания и сопровождения компьютерного оборудования в учебных лабораториях колледжа.

3. Характеристика объекта автоматизации

Объектом автоматизации является процесс регистрации, хранения и обновления сведений об оборудовании, включая данные о ремонтах и замене комплектующих.

4. Требования к системе

- **Функциональные:** регистрация оборудования, поиск, отчёты, резервное копирование.
- **Информационные:** использование JSON-файлов для хранения данных.

Помогу с учебным заданием без предоплаты. Оплата после выполнения
Нажмите на мессенджер, чтобы написать
Telegram ↗ WhatsApp ↗ MAX ↗
+7 (924) 305-23-08 · Пишите в мессенджер disynergy.ru

- **Технические:** клиентское приложение на C#, хранилище на локальном сервере.
- **Эксплуатационные:** работа под Windows 10+, автономный режим.
- **Безопасность:** ограничение доступа по ролям, резервирование данных.

5. Состав и содержание работ

- проектирование структуры данных и интерфейса;
- реализация прототипа;
- тестирование;
- подготовка документации;
- передача системы на сопровождение.

6. Порядок контроля и приёмки

- приёмочные испытания;
- проверка корректности ввода данных;
- контроль восстановления после сбоя.

7. Требования к сопровождению

- периодическое обновление программы и данных;
- ведение журнала ошибок и изменений;
- контроль резервных копий и журналов обслуживания;
- обновление инструкций пользователя и администратора.

8. Приложения

- схема архитектуры ИС;
- диаграмма классов (UML);
- диаграмма последовательности (PlantUML);
- структура JSON-файла данных;
- план резервного копирования.

2.4. Результаты выполнения

Студент представляет:

- разработанное техническое задание (в формате DOCX или PDF);
- схему архитектуры системы (PlantUML или Mermaid);
- таблицу функциональных и эксплуатационных требований;
- описание процедур сопровождения (обновления, резервирования, документирования);

Помогу с учебным заданием без предоплаты. Оплата после выполнения
Нажмите на мессенджер, чтобы написать
Telegram ↗ WhatsApp ↗ MAX ↗
+7 (924) 305-23-08 · Пишите в мессенджер
disynergy.ru

- отчёт-пояснение по разработке ТЗ.

Помогу с учебным заданием без предоплаты. Оплата после выполнения

Нажмите на мессенджер, чтобы написать

Telegram ↗ WhatsApp ↗ MAX ↗

+7 (924) 305-23-08 · Пишите в мессенджер

disynergy.ru