1. **ГОСТ Р ИСО/МЭК 12207. Основные процессы и взаимосвязь между документами в информационной системе согласно стандартам.**

*Цель теории информационных процессов и систем состоит в том, чтобы представить имеющиеся знания в едином комплексе понятий, определений и положений, основываясь на сущности и закономерностях проектирования, внедрения и сопровождения информационных систем.*

ГОСТ Р ИСО/МЭК 12207 "Процессы жизненного цикла программных систем" (ГОСТ 12207, 1999) - один из самых известных и распространенных процессно-ориентированных стандартов в области управления ИТ.

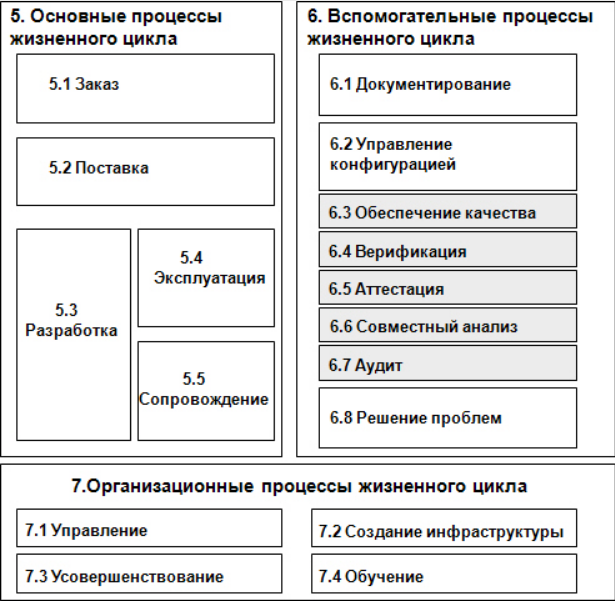
ISO (International Organization for Standardization) –это система мировых стандартов качества, разработанных Международной организацией по стандартизации. Она включает семейство нормативов, которые разрабатываются для разных областей деятельности.

Модель жизненного цикла стандарт определяет как *"структуру, состоящую из процессов, работ и задач, включающих в себя разработку, эксплуатацию и сопровождение программного продукта, охватывающую жизнь системы от установления требований к ней до прекращения ее использования".*

Методологическая основа ГОСТ Р ИСО/МЭК 12207 - разбиение процессов на группы, а именно:

1. Основные. Это процессы, непосредственно относящиеся к *жизненному циклу информационной системы*. Можно считать, что это производственные процессы организации.
2. Вспомогательные. Это процессы, предназначенные для поддержки основных процессов. Сами по себе эти процессы организации не нужны - только в связи с основными процессами, которые они обслуживают. Несколько процессов из этой группы связано с *управлением качеством*.
3. Организационные. Это общекорпоративные процессы, такие как "Обучение" или "Управление". Эти процессы существуют в организации независимо от того, как организовано производство и как устроены вспомогательные процессы.

Структура процессов жизненного цикла программных систем по ГОСТ Р ИСО/МЭК 12207



Процесс обеспечения качества

Этот процесс реализует общие принципы *управления качеством*. Общие принципы состоят в том, что,

во-первых, должен быть разработан и выполнен план работ и задач процесса обеспечения качества.

Во-вторых, должны быть выполнены работы по обеспечению качества продукта (т. е. программной системы), всех задействованных процессов и системы качества (если это предусмотрено договором).

Процесс верификации

Этот процесс обеспечивает, то, "что программные продукты функционируют в полном соответствии с требованиями или условиями, реализованными в предшествующих работах". Например, для процессов существует четыре критерия:

a) соответствие и своевременность установления проектных требований к планированию;

b) пригодность, реализуемость, выполнимость в соответствии с планом и условиями договора выбранных для проекта процессов;

с) применимость стандартов, процедур и условий к процессам проектирования;

d) укомплектованность и обученность персонала в соответствии с условиями договора".

Процесс аттестации

Процесс аттестации — это "процесс определения полноты соответствия установленных требований (*к процедуре испытаний или тестированию системы*), созданной системы или программного продукта их функциональному назначению".

Для выполнения этого процесса можно привлекать независимого исполнителя, который подготовит контрольные примеры, тестовые данные, специально отобранных пользователей для испытаний системы.

Разница между верификацией программы и аттестацией системы. Верификация обеспечивает соответствие программы технологиям и стандартам программирования, условиям договора, требованиям устойчивости к ошибкам и т. п. Аттестация же регламентирует деятельность по тестированию программного продукта.

Процесс совместного анализа

Процесс совместного анализа включает две стороны, участвующие в договоре: анализирующую и анализируемую. Анализируются управление проектом (состояние проекта, предложения по возможным изменениям в проекте) и технические объекты, т. е. создаваемые программные продукты и услуги.

Процесс аудита является процессом определения соответствия требованиям, планам и условиям договора

Процесс может выполняться двумя любыми сторонами, участвующими в договоре, когда одна сторона (ревизующая) проверяет другую сторону (ревизуемую).

Критерии завершения, взаимные обязательства и результаты согласуются сторонами.

Процесс адаптации

Процессом адаптации называется "процесс применения положений настоящего стандарта к условиям реализации конкретного программного проекта". *Адаптация* - обязательная *деятельность* в ходе применения стандарта на практике.

Наличие процесса адаптации подразумевает, что ГОСТ Р ИСО/МЭК 12207 сначала внедряется в организации в целом, а затем для каждого проекта из него "выкраивается" *подмножество* необходимых процессов.

(Недостатками ГОСТ Р ИСО/МЭК 12207-2010 являются отсутствие поддержки процесса документирования, отсутствие требований к архитектуре системы и отсутствие поддержки этапа обследования объекта автоматизации.)

1. **Виды внедрения, план внедрения.**

Виды внедрения информационных систем

Комплексная автоматизация.

- Разработка ИС производится для обеспечения максимального охвата средствами автоматизации всех участков обработки информации и выполняется в сжатые сроки. Наиболее предпочтительный, но и самый дорогостоящий вариант.

Поэтапная автоматизация.

- Автоматизация деятельности отдельных участков с возможностью их дальнейшего объединения в единую систему. Позволяет распределить финансовые затраты во времени и постепенно приспособить производственный цикл к требованиям АСУ.

Процесс внедрения стратегии организации включает ряд этапов:

1. Подготовительный этап

* Анализ существующей системы менеджмента компании, определе­ние ее соответствия внедряемой стратегии. Разработка мер по устранению выявленных дефектов.
* Решение проблемы мотивации менеджеров и специалистов, внедряю­щих стратегию.
* Разработка системы управления внедрением стратегии.
* Разработка системы стратегического контроля.
* Подбор управленческого персонала и принятие решений о расстанов­ке менеджеров, связанных с внедрением стратегии.
* Формирование рабочих групп и команд по решению проблем внедре­ния стратегии.

1. Внедрение стратегии

* Проведение рабочих совещаний групп и команд по внедрению стра­тегии.
* Установление прямых коммуникаций и взаимодействия между работника­ми, внедряющими стратегию, в том числе по каналам телекоммуникаций.
* Финансирование внедрения стратегии, рабочих групп и команд.
* Контроль результатов внедрения стратегии.
* Определение экономической эффективности и результатов стратегии. Сравнение их с планом.

1. Этап завершения реализации основных этапов стратегии

* Оценка общих результатов внедрения стратегии в плановых периодах (месяцы, кварталы, годы).
* Определение причин отклонений и принятие корректирующих ре­шений.
* Анализ перспектив развития стратегических процессов

Управление в процессе реализации стратегии проблем значительно уп­рощается, если используются механизмы политик, процедур и правил.

* Политика определяет общие ориентиры действий.
* Процедуры указыва­ют обязательную последовательность действий при возникновении проблем.
* Правила определяют обязательные действия менеджеров в условиях стан­дартных и часто возникающих ситуаций.

Сервер – это техническое оборудование, оснащенное программным обеспечением, выполняющее поставленные задачи для группы электронных устройств, которые объединены с ним в одну сеть.

Можно считать, что это специальный компьютер для хранения информации или обеспечения работоспособности необходимой программы.

1. **Стратегии, цели и сценарии внедрения.**

Под сце­нари­ями внед­ре­ния ПОпо­нима­ет­ся ком­плекс действий и за­дач, вы­пол­ня­емых и ре­ша­емых за­каз­чи­ком, при­об­ре­та­ющим ПО.

Пос­ле­дова­тельность ша­гов ти­пово­го сце­нария под­ра­зуме­ва­ет:

1) под­го­тов­ку за­яв­ки на при­об­ре­тения ПО;

2) под­го­тов­ку до­гово­ра на при­об­ре­тение ПО;

3) кон­троль за де­ятельностью пос­тавщи­ка;

4) сос­тавле­ние ак­тов при­ем­ки ра­бот.

Первым, самым важным шагом, предшествующим внедрению новой стра­тегии*,* должен бытьанализ системы менеджмента компании, ее резервов, дефектов и их устранение до начала процесса внедрения стратегии*.*Для это­го следует провести управленческую диагностику фирмы, определить силь­ные и слабые стороны системы управления.

Стратегии внедрения:

1. Обучение: Обучение сотрудников или клиентов может быть эффективной стратегией внедрения. Обучение помогает людям освоить новые навыки и привычки, которые позволяют им эффективно использовать новый продукт или технологию.

2. Интеграция и партнерство: Сотрудничество с другими компаниями или организациями может быть полезным при внедрении. Партнерство позволяет объединить ресурсы и экспертизу для достижения общих целей.

3. Пилотирование и масштабирование: Пилотное внедрение — это пробное внедрение нового продукта или идеи в ограниченной области или среде. После тестирования и анализа результатов пилотного проекта можно приступить к масштабированию внедрения.

Цели внедрения:

Цели внедрения могут быть различными и зависят от конкретной ситуации. Однако, некоторые общие цели могут включать:

1. Увеличение прибыли и рыночной доли.

2. Улучшение эффективности и производительности.

3. Создание конкурентного преимущества.

1. **Функции менеджера сопровождения и менеджера развертывания.**

Соп­ро­вож­де­ние ПО — это про­цесс из­ме­нения прог­рам­мной сис­те­мы пос­ле ее пос­тавки за­каз­чи­ку.

- На этом эта­пе ис­прав­ля­ют­ся нез­на­чительные ошиб­ки прог­рамми­рова­ния и (или) кор­ректи­ру­ют­ся от­дельные не­дора­бот­ки, не свя­зан­ные с из­ме­нени­ем об­щей ар­хи­тек­ту­ры сис­те­мы.

- При этом ли­бо из­ме­ня­ют­ся су­щес­тву­ющие, ли­бо до­бав­ля­ют­ся но­вые ком­по­нен­ты ПО.

Соп­ро­вож­де­ние в це­лях ис­прав­ле­ния оши­бок(кор­ректи­ру­ющее соп­ро­вож­де­ние) вы­пол­ня­ет­ся в слу­чае по­яв­ле­ния в ПО оши­бок в прог­рамми­рова­нии, ли­бо оши­бок про­ек­ти­рова­ния, ли­бо оши­бок в сис­темных тре­бова­ни­ях.

* Оши­бки в прог­рамми­рова­нии— дос­та­точ­но лег­ко ус­тра­нимые ошиб­ки.
* Ис­прав­ле­ния оши­бок про­ек­ти­рова­нияоб­хо­дят­ся весьма до­рого, пос­кольку тре­бу­ют кор­ректи­ров­ки и (или) пе­реп­рограм­ми­рова­ния от­дельных ком­по­нен­тов.
* Ошиб­ки в сис­темных тре­бова­ни­яхтре­бу­ют пе­реп­ро­ек­ти­рова­ния всей сис­те­мы и по­это­му яв­ля­ют­ся са­мыми до­рогос­то­ящи­ми.

Соп­ро­вож­де­ние в це­лях адап­та­ции ПО к ус­ло­ви­ям экс­плу­ата­ции (адап­тивное соп­ро­вож­де­ние) вы­пол­ня­ет­ся при из­ме­нении ого­ворен­ных в тех­ни­чес­ком за­дании ус­ло­вий и зак­лю­ча­ет­ся в до­пол­ни­тельной мо­дифи­кации ПО.

Соп­ро­вож­де­ние в це­лях из­ме­нения фун­кци­ональных воз­можнос­тей сис­те­мы про­водит­ся, нап­ри­мер, для под­дер­жки ап­па­рат­ных средств, опе­раци­он­ной сис­те­мы (ОС) и (или) ПО.

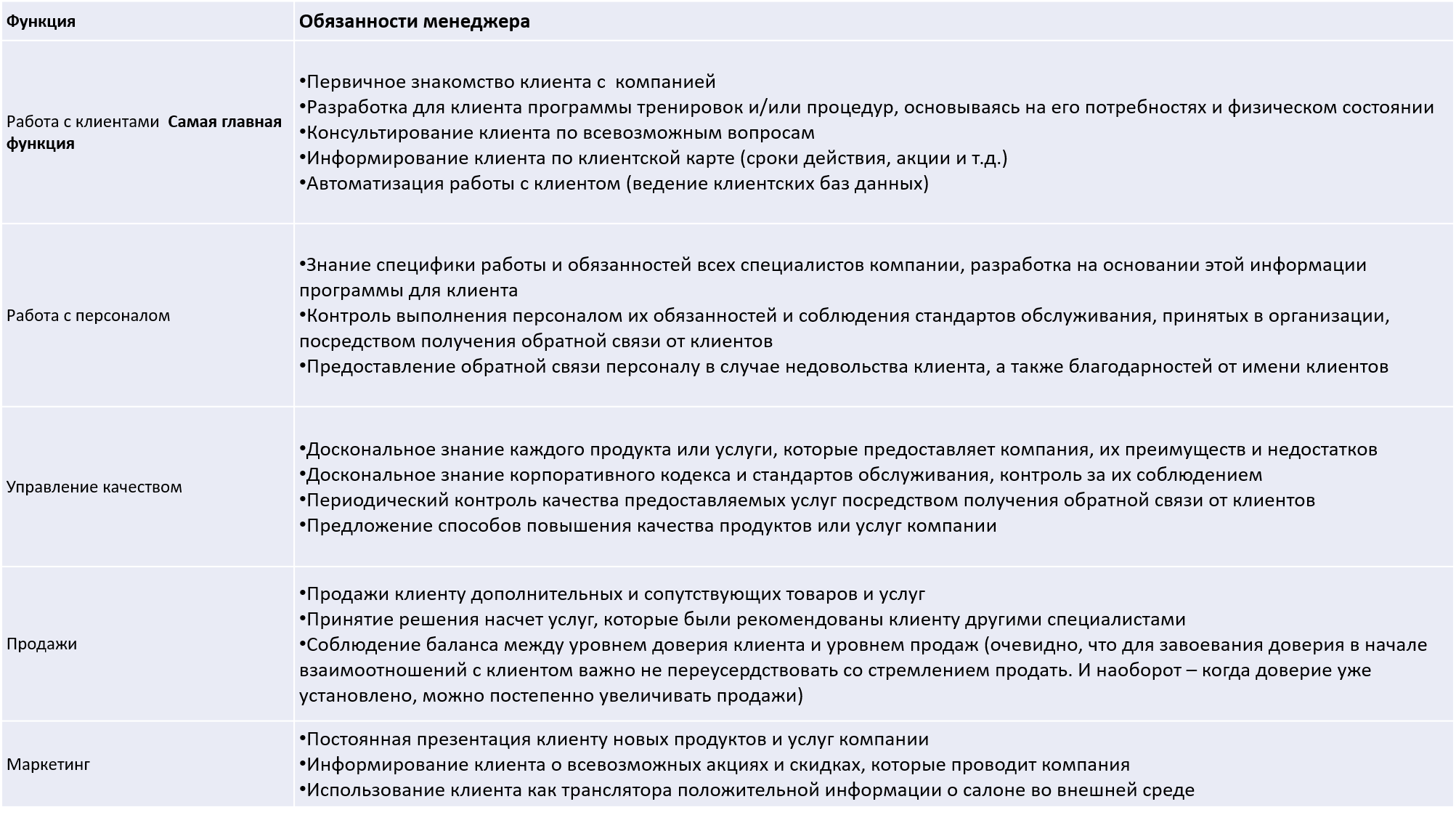
* Рас­ши­рение фун­кци­ональных воз­можнос­тей сис­те­мы, об­легча­ет ра­боту с ней опе­рато­ра и об­легча­ет ее адап­та­цию к но­вому ра­боче­му ок­ру­жению.

Ис­пользо­вание ПО в ор­га­низа­ции сто­ит на вто­ром мес­те по зат­ра­там, ус­ту­пая пер­вое мес­то имен­но соп­ро­вож­де­нию

Менеджер сопровождения – это сотрудник, который замыкает на себе основные функции работы с клиентом и оказывает ему поддержку – от момента первого визита и взаимодействия с администратором и на протяжении всего периода использования клиентом услуг салона или центра.

Менеджер сопровождения старается как можно дольше продлить сотрудничество своей компании с клиентом, постоянно расширяя и углубляя диапазон услуг

Одна из задач менеджера сопровождения – повышение рентабельности обслуживания клиентов и повышение прибыли предприятия.



Раз­верты­вание ПО

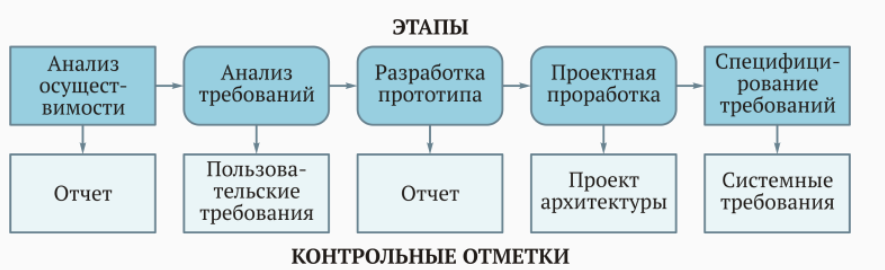
* Раз­верты­вание ПОяв­ля­ет­ся частью ЖЦ ПО и под­ра­зуме­ва­ет все действия, ко­торые де­ла­ют прог­рам­мную сис­те­му го­товой к ис­пользо­ванию.
* Раз­верты­вание мо­жет осу­щест­вляться в про­цес­се раз­ра­бот­ки ПО са­мим прог­раммис­том.
* Ме­нед­же­ры про­ек­тов приз­ва­ны спла­ниро­вать все эта­пы раз­ра­бот­ки прог­рам­мно­го про­дук­та.
* Они так­же дол­жны кон­тро­лиро­вать ход вы­пол­не­ния ра­бот и соб­лю­дение всех тре­бу­емых стан­дартов.
* Пос­то­ян­ный кон­троль за хо­дом вы­пол­не­ния ра­бот не­об­хо­дим для то­го, что­бы про­цесс раз­ра­бот­ки не вы­ходил за вре­менны́е и бюд­жетные ог­ра­ниче­ния.

Раз­верты­вание ПО в ор­га­низа­ции яв­ля­ет­ся тру­до­ем­ким про­цес­сом, сос­то­ящим из ря­да вза­имос­вя­зан­ных эта­пов ко­торые на­зыва­ют­ся кон­трольны­ми про­ек­тны­ми эле­мен­та­ми.

* план раз­верты­вания ПО
* раз­ра­бот­ка спе­цифи­каций (мо­тиви­рован­ных спис­ков) тре­бова­ний
* ус­та­нов­ка/ти­ражи­рова­ние от­дельной под­систе­мы.

На це­левом пред­при­ятии весь этап раз­верты­вания раз­би­ва­ет­ся на ряд по­дэта­пов, за­вер­ша­ющих­ся пред­став­ле­ни­ем ад­ми­нис­тра­тору раз­верты­вания внут­ренних про­ек­тных ре­зульта­тов (кон­трольны­ми от­метка­ми).

Этапы разработки спецификации требований

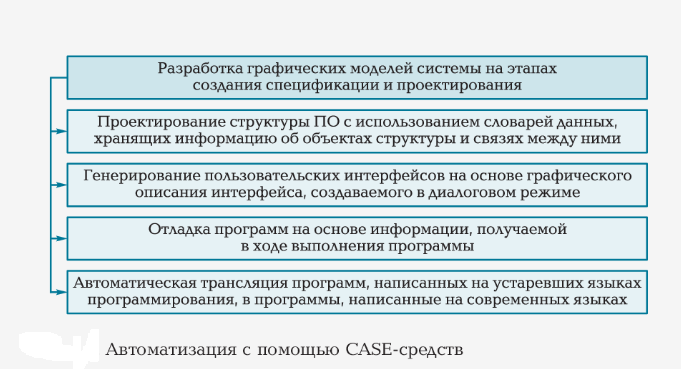


1. **Типовые функции инструментария для автоматизации процесса внедрения информационной системы.**

Ав­то­мати­зиро­ван­ные средс­тва раз­ра­бот­ки ПО— это спе­ци­альный тип ПО, пред­назна­чен­но­го для под­дер­жки про­цес­сов соз­да­ния са­мих прог­рам­мных средств, та­ких как раз­ра­бот­ка тре­бова­ний к ПО, про­ек­ти­рова­ние ПО, ко­диро­вание и тес­ти­рова­ние прог­рамм и т. д.

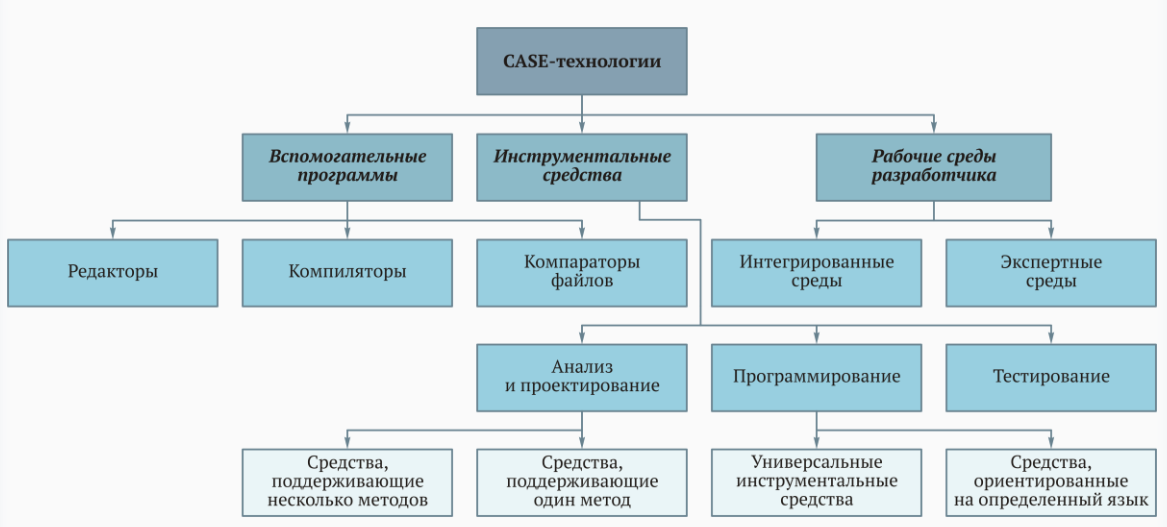
По­нятие ав­то­мати­зиро­ван­ных средств раз­ра­бот­ки ПО эк­ви­вален­тно по­нятию Computer-Aided Software Engineering (CASE), под ко­торым по­нима­ет­ся на­бор инс­тру­мен­тов и ме­тодов прог­рам­мной ин­же­нерии для про­ек­ти­рова­ния ПО и (или) ин­форма­ци­он­ных сис­тем.

Тех­но­логия CASE пред­ла­га­ет под­дер­жку про­цес­са соз­да­ния ПО пу­тем ав­то­мати­зации по­дав­ля­юще­го большинс­тва эта­пов его раз­ра­бот­ки. При­меры CASE-про­цес­сов, ко­торые мож­но ав­то­мати­зиро­вать с по­мощью CASE-средств



Кро­ме улуч­ше­ния ка­чес­тва соз­да­ва­емых прог­рамм при­мене­ние тех­но­логии CASE поз­во­ля­ет на по­рядок по­высить про­из­во­дительность тру­да раз­ра­бот­чи­ков ПО.

Ос­новные ти­пы CASE-технологии.



Оп­ре­деле­ние тре­бова­ний — это про­цесс раз­ра­бот­ки сис­темной спе­цифи­кации.

Про­ек­ти­рова­ние и ре­али­зация — это про­цес­сы пре­об­ра­зова­ния сис­темной спе­цифи­кации в сис­те­му ис­полня­емых прог­рамм.

Ат­теста­ция ПО — это про­цесс про­вер­ки со­от­ветс­твия раз­ра­ботан­ной сис­те­мы ее спе­цифи­кации и пот­ребнос­тям пользо­вате­лей.

Эво­люция ПО — это мо­дер­ни­зация су­щес­тву­ющих прог­рам­мных сис­тем в со­от­ветс­твии с но­выми тре­бова­ни­ями.

1. **Оценка качества функционирования информационной системы.**

Управление качеством ПО

Под ка­чес­твом ПО по­нима­ет­ся его спо­соб­ность при за­дан­ных ус­ло­ви­ях удов­летво­рять ус­та­нов­ленным или пред­по­лага­емым пот­ребнос­тям.

Оцен­ку ка­чес­тва обыч­но про­из­во­дят по ря­ду пред­став­ленных ха­рак­те­рис­тик ПО.

Фун­кци­ональность— спо­соб­ность ПО ре­шать за­дачи, ко­торые со­от­ветс­тву­ют за­фик­си­рован­ным и пред­по­лага­емым пот­ребнос­тям пользо­вате­ля при за­дан­ных ус­ло­ви­ях ис­пользо­вания ПО.

На­деж­ность*—* спо­соб­ность ПО вы­пол­нять тре­бу­емые за­дачи в обоз­на­чен­ных ус­ло­ви­ях на про­тяже­нии за­дан­но­го про­межут­ка вре­мени или ука­зан­ное ко­личес­тво опе­раций.

Удобс­тво ис­пользо­вания — воз­можность ком­фор­тно­го изу­чения ПО пользо­вате­лем.

Эф­фектив­ность — спо­соб­ность ПО обес­пе­чивать тре­бу­емый уро­вень про­из­во­дительнос­ти в со­от­ветс­твии с вы­делен­ны­ми ре­сур­са­ми, вре­менем и дру­гими обоз­на­чен­ны­ми ус­ло­ви­ями.

Удобс­тво соп­ро­вож­де­ния — лег­кость, с ко­торой ПО мо­жет ана­лизи­роваться, тес­ти­роваться, из­ме­няться для ис­прав­ле­ния де­фек­тов, для ре­али­зации но­вых тре­бова­ний, для об­легче­ния дальнейше­го об­слу­жива­ния и адап­ти­роваться к име­юще­муся ок­ру­жению.

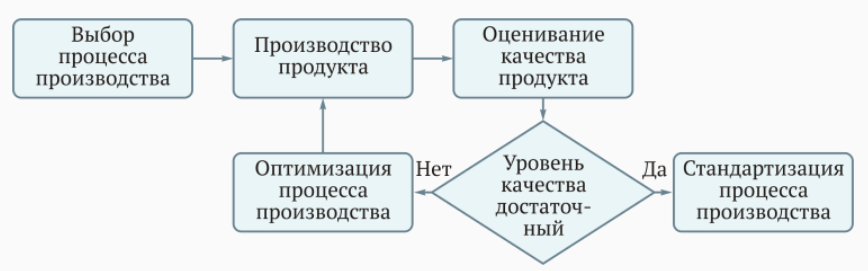
Пор­та­тив­ность— ха­рак­те­ризу­ет ПО с точ­ки зре­ния мо­бильнос­ти его пе­рено­са из од­но­го ок­ру­жения в дру­гое.

Те­оре­тичес­ки уп­равле­ние ка­чес­твом ос­но­выва­ет­ся на прин­ци­пе оп­ре­деле­ния стан­дартов и про­цедур­ных норм, в со­от­ветс­твии с ко­торы­ми дол­жно раз­ра­баты­ваться ПО, а так­же на про­вер­ке вы­пол­не­ния этих норм все­ми раз­ра­бот­чи­ками.

Про­цесс уп­равле­ния ка­чес­твом сос­то­ит из трех ви­дов де­ятельнос­ти



Пра­вило ус­пешно­го уп­равле­ния ка­чес­твом сос­то­ит в том, что ка­чес­тво про­цес­са про­из­водс­тва ПО вли­яет и на ка­чес­тво го­тово­го прог­рам­мно­го про­дук­та.



Уп­равле­ние ка­чес­твом ПО вклю­ча­ет в се­бя сле­ду­ющие про­цес­сы:

1) оп­ре­деле­ние стан­дартов, рег­ла­мен­ти­ру­ющих про­цесс раз­ра­бот­ки ПО;

2) кон­троль за про­цес­сом раз­ра­бот­ки (для обес­пе­чения вы­пол­не­ния стан­дартов);

3) соз­да­ние от­четнос­ти о хо­де про­цес­са раз­ра­бот­ки для ме­нед­же­ра про­ек­та и за­каз­чи­ка ПО.

Су­щес­тву­ет два вза­имо­допол­ня­ющих под­хо­да к про­цес­су кон­тро­ля ка­чес­тва:

1) про­вер­ки ка­чес­тва, ког­да прог­рам­мный про­дукт, соп­ро­вож­да­ющая до­кумен­та­ция и про­цесс раз­ра­бот­ки ана­лизи­ру­ют­ся груп­пой про­веря­ющих.

2) ав­то­мати­зиро­ван­ная оцен­ка ПО, ког­да прог­рам­мный про­дукт и его до­кумен­та­ция про­веря­ют­ся спе­ци­альной компьютер­ной прог­раммой, ко­торая со­пос­тавля­ет их со стан­дарта­ми дан­но­го про­ек­та.

1. **CALS-технологии.**

CALS (Continuous Acquisition and Lifecycle Support — неп­ре­рыв­ная ин­форма­ци­он­ная под­дер­жка пос­та­вок и ЖЦ) оз­на­ча­ет со­вокуп­ность прин­ци­пов и тех­но­логий ин­форма­ци­он­ной под­дер­жки ЖЦ про­дук­ции на всех его ста­ди­ях.

CALS-технологии — это процесс создания единого информационного пространства в отдельно взятой системе обеспечения жизненного цикла продукции.

CALS-тех­но­логии приз­ва­ны слу­жить средс­твом, ин­тегри­ру­ющим про­мыш­ленные ав­то­мати­зиро­ван­ные сис­те­мы в еди­ную мно­гофун­кци­ональную сис­те­му.

Целью ин­тегра­ции яв­ля­ет­ся по­выше­ние эф­фектив­ности соз­да­ния и ис­пользо­вания слож­ных прог­рам­мных сис­тем.

При­мене­ние CALS-тех­но­логии поз­во­ля­ет рез­ко по­высить эф­фектив­ность раз­ра­бот­ки ПО по ря­ду ос­новных при­чин.

Во-пер­вых, обос­но­ван­ность ре­шений, при­нима­емых в ав­то­мати­зиро­ван­ной сис­те­ме уп­равле­ния пред­при­яти­ем (АСУП), бу­дет вы­ше, ес­ли со­от­ветс­тву­ющие прог­раммы АСУП име­ют опе­ратив­ный дос­туп не только к ба­зе дан­ных АСУП, но и к ба­зам дан­ных дру­гих ав­то­мати­зиро­ван­ных сис­тем *(сис­тем ав­то­мати­зиро­ван­но­го про­ек­ти­рова­ния, ав­то­мати­зиро­ван­ных сис­тем тех­но­логи­чес­кой под­го­тов­ки про­из­водс­тва и ав­то­мати­зиро­ван­ных сис­тем уп­равле­ния тех­но­логи­чес­ким про­цес­сом)*

Во-вто­рых, при­мене­ние CALS-тех­но­логий поз­во­ля­ет су­щес­твен­но сок­ра­тить объемы про­ек­тных ра­бот, так как опи­сания ра­нее вы­пол­ненных удач­ных раз­ра­боток ком­по­нен­тов и ус­тройств, сос­тавных час­тей обо­рудо­вания, ма­шин и сис­тем, про­ек­ти­ровав­шихся ра­нее, хра­нят­ся в ба­зах дан­ных се­тевых файл-сер­ве­ров, дос­тупных лю­бому пользо­вате­лю CALS-тех­но­логии.

Дос­тупность к ин­форма­ции обес­пе­чива­ет­ся сог­ла­сован­ностью фор­ма­тов, спо­собов, ру­ководств об­щей ин­тегри­рован­ной сис­те­мы.

В-третьих, су­щес­твен­но об­легча­ет­ся ре­шение проб­лем ре­мон­топри­год­ности, ин­тегра­ции про­дук­ции в раз­лично­го ро­да сис­те­мы и сре­ды, адап­та­ции к ме­ня­ющим­ся ус­ло­ви­ям экс­плу­ата­ции и др.

Что­бы дос­тичь дол­жно­го уров­ня вза­имо­действия про­мыш­ленных ав­то­мати­зиро­ван­ных сис­тем, тре­бу­ет­ся соз­да­ние еди­ного ин­форма­ци­он­но­го прос­транс­тва в рам­ках как от­дельных пред­при­ятий, так и в рам­ках объеди­нения пред­при­ятий

Основные принципы CALS-технологий базируются на контроле и организации этапов существования продукции. К ним относят:

* обеспечение системного управления минимизацию затрат на всех стадиях;
* использование стандартных механизмов описания управляемых объектов дифференциацию программных элементов на основе использования общих стандартов (данных и интерфейсов доступа) и применение платформ на коммерческой основе;
* представление информации на безбумажной основе с приоритетом использования электронной подписи;
* сопутствующий инжиниринг всех процессов;
* непрерывное корректирование и усовершенствование с целью создания оптимальной модели управления.

Примеры использования

CALS-технологии в России используются на многих отечественных предприятиях, как гражданского, так и военного сектора.

* Электронная документация используется для многих изделий. К примеру, в авиации для самолетов, вертолетов, авиационных двигателей и комплектующих.
* Помимо этого, ведутся разработки систем навигации, телефонной и радио связи, управления.
* Применяются при проектировании и разработке автотракторной техники.
* Элементы системы используются на Воронежском механическом заводе, в государственной корпорации "Росатом", НПП "Аэросила", ОАО "Российские железные дороги" и др.

1. **Организация процесса обновления в информационной системе.**

Об­новле­ни­е ПО – это до­пол­не­ния к прог­рам­мно­му обес­пе­чению, пре­дот­вра­ща­ющие или ус­тра­ня­ющие не­полад­ки, по­выша­ющие бе­зопас­ность ли­бо улуч­ша­ющие про­из­во­дительность компьютер­ной сис­те­мы.

На­боры об­новле­ний, ис­прав­ле­ний и (или) улуч­ше­ний компьютер­ной прог­раммы, по­став­ля­емые в ви­де еди­ного ус­та­новоч­но­го дис­три­бути­ва, при­нято на­зывать па­кета­ми об­новле­ний. Па­кеты об­новле­ния обыч­но ну­меру­ют­ся и крат­ко ука­зыва­ют­ся как SP1, SP2, SP3 и т. д.

Патчи

Патч(или «зап­латка») — это не­большой прог­рам­мный код, пред­назна­чен­ный для за­мены оши­боч­ной и (или) не­оп­ти­мизи­рован­ной час­ти дру­гой прог­раммы.

Ис­прав­ле­ние с по­мощью пат­чей мо­жет при­меняться к уже ус­та­нов­ленной прог­рамме ли­бо к ее ис­ходным ко­дам. Сю­да вхо­дят ис­прав­ле­ние оши­бок, из­ме­нение внеш­не­го ви­да, улуч­ше­ние про­из­во­дительнос­ти прог­рамм, а так­же лю­бые дру­гие из­ме­нения, ко­торые раз­ра­бот­чик по­желал сде­лать.

Раз­мер пат­чей мо­жет варьиро­ваться от нес­кольких ки­лобайт до со­тен ме­габайт.

Хот­фикс(Hotfix) — тер­мин, при­меня­ющийся для пат­чей, ко­торые ус­та­нав­ли­ва­ют­ся на ра­бота­ющую сис­те­му без пе­реза­пус­ка.

Час­то хот­фиксы пред­назна­чены для ре­шения кон­крет­ных проб­лем той или иной ор­га­низа­ции или кон­крет­ных пользо­вате­лей и не вы­ходят за ее пре­делы.

Так­же хот­фиксы мо­гут пре­дос­тавляться для кон­крет­ной кон­фи­гура­ции обо­рудо­вания и не ра­ботать с дру­гими сис­те­мами.

Процесс управления обновлениями ПО

Про­цесс уп­равле­ния об­новле­ни­ями ПО на­зыва­ет­ся Patch Management.

Раз­ли­ча­ют раз­нос­тные и сум­марные па­кеты об­новле­ний.

Раз­нос­тный па­кет со­дер­жит только те об­новле­ния, ко­торых не бы­ло в пре­дыду­щих па­кетах об­новле­ния, тог­да как сум­марный па­кет вклю­ча­ет в се­бя со­дер­жи­мое всех пре­дыду­щих об­новле­ний.

Виды обновлений

Раз­ли­ча­ют об­новле­ния ОС и прик­ладно­го ПО.

Об­новле­ния для ОС и сер­верно­го ПО при­меня­ют­ся в це­лях под­дер­жки над­ле­жаще­го уров­ня бе­зопас­ности и ус­тра­нения «дыр» в за­щите.

Об­новле­ния прик­ладно­го ПО (нап­ри­мер, Microsoft Office, Adobe Acrobat, кли­ент­ские час­ти биз­нес-при­ложе­ний) не­об­хо­димы для ре­шения воз­никших проб­лем с час­то ис­пользу­емы­ми или важ­ны­ми биб­ли­оте­ками и дру­гими час­тя­ми ис­ходно­го ко­да.

Метод управления обновлениями

Ме­тод уп­равле­ния об­новле­ни­ями яв­ля­ет­ся ком­би­наци­ей под­хо­да к тес­ти­рова­нию об­новле­ний и под­хо­да к раз­верты­ванию ре­лизов с об­новле­ни­ями.

Раз­ли­ча­ют два под­хо­да к тес­ти­рова­нию об­новле­ний:

1) на ло­кальных вир­ту­альных ма­шинах;

2) в пол­но­цен­ной тес­то­вой сре­де.

В пер­вом слу­чае для соз­да­ния вир­ту­альных ма­шин и се­тей ис­пользу­ют­ся ли­бо про­дукт VMware, ли­бо про­дукт Oracle VirtualBox.

Как пра­вило, тех­но­логия вир­ту­али­зации ис­пользу­ет­ся для не­больших се­тей с чис­лом ра­бочих стан­ций 45…70, не бо­лее.

Достоинства виртуализации процесса тестирования

* соз­да­ние са­мой вир­ту­альной ма­шины не тре­бу­ет больших зат­рат по вре­мени;
* вир­ту­альных ма­шин на од­ной фи­зичес­кой плат­форме мо­жет быть нес­колько;
* каж­дая вир­ту­альная ма­шина име­ет свои собс­твен­ные вир­ту­альные ап­па­рат­ные ком­по­нен­ты (па­мять, про­цес­сор, жес­ткий диск, се­тевые адап­те­ры);
* воз­можность сде­лать «сни­мок» те­куще­го сос­то­яния сис­те­мы и со­дер­жи­мого дис­ков од­ним кли­ком мы­ши, а за­тем в те­чение очень ко­рот­ко­го про­межут­ка вре­мени вер­нуться в ис­ходное сос­то­яние, что мо­жет быть очень по­лез­ным при ус­ло­вии об­на­руже­ния об­новле­ния, ус­та­нов­ка ко­торо­го вы­зыва­ет кри­тичес­кие не­ис­прав­ности в сис­те­ме.

Вто­рой ва­ри­ант — тес­ти­рова­ние с ис­пользо­вани­ем пол­но­цен­ной тес­то­вой сре­ды — при­меним для больших про­мыш­ленных се­тей.

Этот ва­ри­ант га­ран­ти­ру­ет вы­сокую чис­то­ту тес­ти­рова­ния, пос­кольку ис­пользу­ет те же под­хо­ды к ус­та­нов­ке об­новле­ний и инс­тру­мен­ты для уп­равле­ния об­новле­ни­ями, что и в про­мыш­ленной сре­де, при­чем про­цесс тес­ти­рова­ния мож­но уп­ростить, ис­пользуя все­го па­ру кор­по­ратив­ных компьюте­ров (так на­зыва­емых тес­то­вых кли­ен­тов).

Ста­дии про­цес­са Patch Management.

Про­цесс уп­равле­ния об­новле­ни­ями сос­то­ит из нес­кольких эта­пов (ста­дий).

1. Под­го­тов­ка тес­то­вых кли­ен­тов.

На ма­шину для тес­ти­рова­ния об­новле­ний ус­та­нав­ли­ва­ет­ся об­раз ОС, вклю­ча­ющий при­ложе­ния, а так­же ут­вер­жден­ные ра­нее про­тес­ти­рован­ные об­новле­ния. При сле­ду­ющей заг­рузке ОС про­ис­хо­дит ав­то­мати­чес­кая ус­та­нов­ка дан­ных ут­вер­жден­ных об­новле­ний, а это оз­на­ча­ет, что тес­то­вая сре­да под­го­тов­ле­на.

2. Соз­да­ние лис­тов об­новле­ний или патч-лис­тов, вклю­ча­ющих об­новле­ния, вы­шед­шие в те­кущем ме­сяце и под­хо­дящие под оп­ре­деле­ние «тре­бу­емые об­новле­ния».

По­доб­ные патч-лис­ты поз­во­лят еже­месяч­но ав­то­мати­чес­ки об­новлять при­ложе­ния, для ко­торых они соз­да­ны.

3. Раз­верты­вание в тес­то­вой сре­де.

При до­бав­ле­нии па­кета к кол­лекции тес­то­вых кли­ен­тов ав­то­мати­чес­ки на­чина­ет­ся раз­верты­вание. Дан­ная про­цеду­ра за­нима­ет все­го нес­колько ми­нут, по ис­те­чении ко­торых в об­ласти уве­дом­ле­ний па­нели за­дач по­яв­ля­ет­ся со­от­ветс­тву­ющий зна­чок, пос­ле двойно­го на­жатия на ко­торый от­кры­ва­ет­ся ок­но *Configuration Manager*, где пос­ле вы­бора об­новле­ний на­чина­ет­ся ус­та­нов­ка, за­кан­чи­ва­юща­яся зап­ро­сом на пе­резаг­рузку сис­те­мы.

4. Раз­верты­вание на пи­лот­ных пользо­вате­лях — ста­дия *Pre Deployment*.

На этой ста­дии го­товит­ся спи­сок про­тес­ти­рован­ных об­новле­ний, ко­торый от­прав­ля­ет­ся на кли­ен­ты пи­лот­ных пользо­вате­лей и до­бав­ля­ет­ся к кол­лекции пи­лот­ных кли­ен­тов. Как пра­вило, в ка­чес­тве пи­лот­ных кли­ен­тов вы­бира­ют­ся те пользо­вате­ли, ко­торые хо­рошо раз­би­ра­ют­ся в ПО, за ко­торое они от­ве­ча­ют. Их чис­ло варьиру­ет­ся в за­виси­мос­ти от чис­ла ра­бочих стан­ций се­ти. Ста­дия Pre Deployment про­дол­жа­ет­ся в те­чение че­тырех дней, по ис­те­чении ко­торых фор­ми­ру­ет­ся окон­ча­тельный спи­сок об­новле­ний, ко­торые бу­дут ус­та­нов­ле­ны на все ра­бочие стан­ции про­мыш­ленной се­ти.

1. **Регламенты обновления.**

Регламенты обновления информационной системы (ИС) представляют собой детально разработанные инструкции и правила, регулирующие процесс обновления системы. Они включают в себя процедуры, которые должны быть выполнены до, во время и после обновления, а также определяют роли и ответственность участников процесса.

Основные элементы регламента обновления

1. Цели и задачи обновления:
   * Определение целей обновления (например, улучшение функциональности, повышение безопасности, исправление ошибок).
   * Конкретизация задач, которые необходимо выполнить для достижения этих целей.
2. Периодичность обновлений:
   * Определение частоты плановых обновлений (ежемесячные, квартальные, ежегодные и т.д.).
   * Установление условий для проведения внеплановых обновлений (например, в случае обнаружения критической уязвимости).
3. Процедура подготовки:
   * Анализ текущего состояния системы и определение необходимости обновления.
   * Создание резервных копий данных и системных конфигураций.
   * Тестирование обновлений на тестовых или пилотных системах для выявления возможных проблем.
4. Планирование обновления:
   * Составление детального плана обновления, включающего сроки, этапы, необходимые ресурсы и ответственных лиц.
   * Определение временных окон для проведения обновлений с минимальным влиянием на пользователей.
5. Процесс выполнения обновления:
   * Уведомление пользователей о планируемом обновлении и возможных перерывах в работе системы.
   * Пошаговая инструкция по выполнению обновления, включая последовательность действий и меры предосторожности.
   * Реализация обновления в соответствии с установленным планом и мониторинг его выполнения.
6. Проверка после обновления:
   * Функциональное тестирование обновленной системы для подтверждения корректной работы всех функций.
   * Проверка производительности системы и выявление возможных проблем.
   * Проведение тестов безопасности для удостоверения отсутствия новых уязвимостей.
7. Документирование обновлений:
   * Ведение журнала изменений, в котором фиксируются все проведенные обновления, их результаты и выявленные проблемы.
   * Обновление технической и пользовательской документации в соответствии с внесенными изменениями.
8. Откат и восстановление:
   * План действий на случай неудачного обновления, включая процедуры отката к предыдущей версии.
   * Тестирование механизма восстановления для обеспечения его надежности.
9. Обратная связь и доработка:
   * Сбор и анализ обратной связи от пользователей относительно проведенного обновления.
   * Исправление выявленных после обновления ошибок и внедрение улучшений на основе полученной обратной связи.

Пример регламента обновления

1. Цель и задачи обновления

* Цель: Обновление системы для улучшения безопасности.
* Задачи: Установка патча безопасности, обновление антивирусного ПО, проверка системы на уязвимости.

2. Периодичность обновлений

* Плановые обновления: Раз в месяц, каждое второе воскресенье.
* Внеплановые обновления: При обнаружении критической уязвимости.

3. Процедура подготовки

* Создание резервной копии всех данных и конфигураций за день до обновления.
* Тестирование обновления на тестовой системе за неделю до плановой даты.

4. Планирование обновления

* Составление плана обновления с указанием временного окна с 02:00 до 04:00.
* Назначение ответственных лиц: системный администратор, инженер по безопасности.

5. Процесс выполнения обновления

* Уведомление пользователей за три дня до обновления.
* Отключение системы в 02:00, выполнение обновления по инструкции.
* Мониторинг системы в процессе обновления.

6. Проверка после обновления

* Проведение функциональных тестов с 04:00 до 05:00.
* Проверка производительности системы в течение первых двух часов после обновления.
* Тестирование безопасности в течение дня после обновления.

7. Документирование обновлений

* Запись изменений в журнал изменений.
* Обновление документации системы в течение недели после обновления.

8. Откат и восстановление

* План отката: В случае проблем восстановление резервной копии в течение двух часов.
* Тестирование процедуры восстановления каждый квартал.

9. Обратная связь и доработка

* Сбор обратной связи от пользователей в течение недели после обновления.
* Внедрение необходимых исправлений и улучшений в течение месяца.

Заключение

Регламент обновления является важным инструментом для организации процесса обновления информационной системы. Он обеспечивает четкость и последовательность действий, минимизирует риски и помогает поддерживать высокое качество работы системы.

1. **Тестирование программного обеспечения в процессе внедрения и эксплуатации.**

Тестирование программного обеспечения

Тес­ти­рова­ние ис­пользу­ет­ся для об­на­руже­ния оши­бок и де­фек­тов в ПО. Целью тес­ти­рова­ния яв­ля­ет­ся вы­яв­ле­ние в ПО скры­тых де­фек­тов до то­го, как оно бу­дет сда­но за­каз­чи­ку.

Тестирование сборки

Пос­ле то­го как про­тес­ти­рова­ны все от­дельные прог­рам­мные мо­дули, вы­пол­ня­ет­ся сбор­ка сис­те­мы, в ре­зульта­те че­го соз­да­ет­ся час­тичная или пол­ная прог­рам­мная сис­те­ма.

При ее тес­ти­рова­нии вы­яв­ля­ют­ся проб­ле­мы, воз­ни­ка­ющие при вза­имо­действии прог­рам­мных мо­дулей сис­те­мы.

Тес­ты, про­веря­ющие сбор­ку, раз­ра­баты­ва­ют­ся на ос­но­ве сис­темной спе­цифи­кации, при­чем тес­ти­рова­ние сбор­ки сле­ду­ет на­чинать сра­зу пос­ле соз­да­ния ра­ботос­по­соб­ных вер­сий прог­рам­мных мо­дулей сис­те­мы.

В про­цес­се тес­ти­рова­ния сбор­ки воз­ни­ка­ет проб­ле­ма ло­кали­зации вы­яв­ленных оши­бок.

Меж­ду ком­по­нен­та­ми сис­те­мы су­щес­тву­ют слож­ные вза­имо­от­но­шения, и при об­на­руже­нии ано­мальных вы­ход­ных дан­ных бы­ва­ет труд­но ус­та­новить ис­точник ошиб­ки.

Что­бы об­легчить ло­кали­зацию оши­бок, сле­ду­ет ис­пользо­вать по­шаго­вый ме­тод сбор­ки и тес­ти­рова­ния сис­те­мы.

Сна­чала сле­ду­ет соз­дать ми­нимальную кон­фи­гура­цию сис­те­мы и ее про­тес­ти­ровать. За­тем в ми­нимальную кон­фи­гура­цию нуж­но до­бавить но­вые ком­по­нен­ты и сно­ва про­тес­ти­ровать (и так до пол­ной сбор­ки сис­те­мы).

Тестирование интерфейса

Дан­ное тес­ти­рова­ние вы­пол­ня­ет­ся в тех слу­ча­ях, ког­да мо­дули или под­систе­мы ин­тегри­ру­ют­ся в большие сис­те­мы.

Цель тес­ти­рова­ния ин­терфейса — вы­явить ошиб­ки, воз­ни­ка­ющие в сис­те­ме вследс­твие оши­бок в ин­терфейсах.

Типы интерфейсов

* па­рамет­ри­чес­кие ин­терфейсы, в ко­торых ссыл­ки на дан­ные и фун­кции пе­реда­ют­ся в ви­де па­рамет­ров от од­но­го ком­по­нен­та к дру­гому;
* ин­терфейсы раз­де­ля­емой па­мяти, в ко­торых ка­кой-ли­бо блок па­мяти сов­мес­тно ис­пользу­ет­ся раз­ны­ми под­систе­мами. Од­на под­систе­ма по­меща­ет дан­ные в па­мять, а дру­гие под­систе­мы ис­пользу­ют эти дан­ные;
* про­цедур­ные ин­терфейсы, в ко­торых од­на под­систе­ма ин­капсу­лиру­ет (вклю­ча­ет в се­бя) на­бор про­цедур, вы­зыва­емых из дру­гих под­систем. Та­кой тип ин­терфейса име­ют объек­ты и абс­трак­тные ти­пы дан­ных;
* ин­терфейсы пе­реда­чи со­об­ще­ний**,** в ко­торых од­на под­систе­ма зап­ра­шива­ет сер­вис у дру­гой под­систе­мы пос­редс­твом пе­реда­чи ей со­об­ще­ния, при­чем от­ветное со­об­ще­ние со­дер­жит ре­зульта­ты вы­пол­не­ния сер­ви­са (при­мер: так ра­бота­ют сис­те­мы «кли­ент-сер­вер»).

Классы ошибок

Не­кор­рек­тное ис­пользо­вание ин­терфейса име­ет мес­то, ког­да при вы­зове мо­дуля вы­зыва­ющий мо­дуль пе­реда­ет вы­зыва­емо­му па­рамет­ры, на­рушая по­рядок их сле­дова­ния и (или) ука­зывая не­вер­ное их чис­ло и (или) тип пе­реда­ва­емых дан­ных.

Неп­ра­вильное по­нима­ние ин­терфейсов име­ет мес­то, ког­да по­веде­ние вы­зыва­емо­го мо­дуля не сов­па­да­ет с ожи­да­емым. В этом слу­чае по­веде­ние вы­зыва­юще­го мо­дуля ста­новит­ся неп­ред­ска­зу­емым. Нап­ри­мер, ес­ли прог­рамма би­нар­но­го по­ис­ка вы­зыва­ет­ся для по­ис­ка за­дан­но­го эле­мен­та в не­упо­рядо­чен­ном мас­си­ве, то в ра­боте прог­раммы про­изойдет сбой.

Ошиб­ки син­хро­низа­ции ин­терфейсов встре­ча­ют­ся в сис­те­мах ре­ально­го вре­мени, где ис­пользу­ют­ся ин­терфейсы раз­де­ля­емой па­мяти или пе­реда­чи со­об­ще­ний. Под­систе­ма-ис­точник и под­систе­ма-при­ем­ник дан­ных мо­гут ра­ботать с раз­ной ско­ростью. Ес­ли при про­ек­ти­рова­нии ин­терфейса не учи­тывать этот фак­тор, при­ем­ник мо­жет по­лучить дос­туп к ус­та­рев­шим дан­ным, по­тому что ис­точник к то­му мо­мен­ту еще не ус­пел об­но­вить сов­мес­тно ис­пользу­емые дан­ные.

Ошиб­ки син­хро­низа­ции ин­терфейсо***в*** встре­ча­ют­ся в сис­те­мах ре­ально­го вре­мени, где ис­пользу­ют­ся ин­терфейсы раз­де­ля­емой па­мяти или пе­реда­чи со­об­ще­ний.

*Под­систе­ма-ис­точник и под­систе­ма-при­ем­ник дан­ных мо­гут ра­ботать с раз­ной ско­ростью*. Ес­ли при про­ек­ти­рова­нии ин­терфейса не учи­тывать этот фак­тор, при­ем­ник мо­жет по­лучить дос­туп к ус­та­рев­шим дан­ным, по­тому что ис­точник к то­му мо­мен­ту еще не ус­пел об­но­вить сов­мес­тно ис­пользу­емые дан­ные.

За­дачу тес­ти­рова­ния ин­терфейсов ус­ложня­ет тот факт, что не­кото­рые ошиб­ки мо­гут про­явиться только в не­обыч­ных ус­ло­ви­ях.

Объектно-ориентированные системы (ООС)

Сис­те­мы, раз­ра­ботан­ные по фун­кци­ональной мо­дели, и объек­тно-ори­ен­ти­рован­ные сис­те­мы (О­ОС) име­ют сле­ду­ющие су­щес­твен­ные от­ли­чия:

* объек­ты, как от­дельные прог­рам­мные ком­по­нен­ты, пред­став­ля­ют со­бой неч­то большее, чем от­дельные под­прог­раммы или фун­кции;
* объек­ты, ин­тегри­рован­ные в под­систе­мы, обыч­но сла­бо свя­заны меж­ду со­бой, и по­это­му слож­но оп­ре­делить са­мый вер­хний уро­вень сис­те­мы;
* при ана­лизе пов­торно ис­пользу­емых объек­тов их ис­ходный код мо­жет быть не­до­ступ­ным для ис­пы­тате­лей.

При­мени­тельно к О­ОС мож­но оп­ре­делить че­тыре уров­ня тес­ти­рова­ния (тер­ми­ноло­гия объек­тно-ори­ен­ти­рован­но­го прог­рамми­рова­ния):

1) тес­ти­рова­ние от­дельных ме­тодов, ас­со­ци­иро­ван­ных с объек­та­ми;

2) тес­ти­рова­ние от­дельных клас­сов объек­тов;

3) тес­ти­рова­ние клас­те­ров (спе­ци­альных групп) объек­тов — обыч­но ис­пользу­ет ме­тоды, ос­но­ван­ные на сце­нари­ях;

4) тес­ти­рова­ние сис­те­мы — ве­рифи­кация и ат­теста­ция О­ОС вы­пол­ня­ет­ся так же, как и для лю­бых дру­гих ти­пов сис­тем.

Три подхода при тестировании сборки ООС

При тес­ти­рова­нии сбор­ки О­ОС ис­пользу­ет­ся три под­хо­да:

1) тес­ти­рова­ние мак­си­мально­го чис­ла сце­нари­ев и ва­ри­ан­тов их ис­пользо­вания, со­вокуп­ность ко­торых опи­сыва­ет один ре­жим ра­боты сис­те­мы;

2) тес­ти­рова­ние по­токов, про­веря­ющее сис­темные от­кли­ки на ввод­ные воз­му­щения, и пос­кольку О­ОС яв­ля­ют­ся со­бытийно-уп­равля­емы­ми, то для них осо­бен­но под­хо­дит дан­ный вид тес­ти­рова­ния;

3) тес­ти­рова­ние пос­ле­дова­тельнос­ти вза­имо­действий меж­ду объек­та­ми.

Инструментальные средства тестирования

В об­щем цик­ле раз­ра­бот­ки ПО тес­ти­рова­ние яв­ля­ет­ся са­мым до­рогос­то­ящим и тру­до­ем­ким про­цес­сом, по­это­му в нас­то­ящее вре­мя соз­дан и ши­роко ис­пользу­ет­ся це­лый ряд ав­то­мати­зиру­ющих этот про­цесс инс­тру­мен­тальных средств.

Ком­па­ратор файлов по ре­зульта­там срав­не­ния сос­тавля­ет от­чет об об­на­ружен­ных раз­ли­чи­ях. Ком­па­рато­ры осо­бен­но важ­ны при срав­не­нии раз­личных вер­сий прог­раммы. Раз­ли­чия в ре­зульта­тах ука­зыва­ют на воз­можные проб­ле­мы, су­щес­тву­ющие в но­вой вер­сии сис­те­мы.

Ге­нера­тор от­че­тов фор­ми­ру­ет от­че­ты по ре­зульта­там тес­ти­рова­ния.

Ди­нами­чес­кий ана­лиза­тор под­счи­тыва­ет, сколько раз вы­пол­ня­ет­ся каж­дый опе­ратор тес­ти­ру­емой прог­раммы и вы­да­ет от­чет по ре­зульта­там сче­та.

Ими­татормо­дели­ру­ют ма­шину, на ко­торой бу­дет вы­пол­няться прог­рамма (ес­ли это це­левой ими­татор) или мо­дели­ру­ет вза­имо­действия с ин­терфейсом пользо­вате­ля (ес­ли это ими­татор пользо­вательско­го ин­терфейса) или, на­конец, ге­нери­ру­ет пос­ле­дова­тельнос­ти пов­то­ря­ющих­ся тран­закций, т. е. зап­ро­сов к ба­зе дан­ных (ес­ли это ими­татор вво­да-вы­вода).

1. **Эксплуатационная документация.**

Го­сударст­вен­ный стан­дарт ГОСТ 19.101—77 «Еди­ная сис­те­ма прог­рам­мной до­кумен­та­ции (ЕСПД).

Ви­ды прог­рамм и прог­рам­мных до­кумен­тов пре­дус­матри­ва­ет соз­да­ние ря­да ос­новных до­кумен­тов, ка­са­ющих­ся ПО.

Прог­рам­мная и экс­плу­ата­ци­он­ная до­кумен­та­ция дол­жна со­дер­жать опи­сание всех ва­ри­ан­тов при­мене­ния пер­со­нально­го компьюте­ра, ре­жимов фун­кци­они­рова­ния и по­луча­емых ре­зульта­тов при лю­бых (кор­рек­тных и оши­боч­ных) вход­ных дан­ных и действи­ях пользо­вате­ля, вклю­чая со­об­ще­ния об ошиб­ках, действия пос­ле них и спо­собы их ус­тра­нения.

Прог­рам­мная и экс­плу­ата­ци­он­ная до­кумен­та­ция дол­жна так­же ха­рак­те­ризо­ваться пол­но­той, неп­ро­тиво­речи­востью, ссы­лоч­ной це­лос­тностью, удобс­твом ис­пользо­вания и ха­рак­те­ризо­ваться дос­та­точ­ностью опи­сания про­цедур ин­стал­ля­ции (на­чальной ус­та­нов­ки), кон­фи­гури­рова­ния и ис­пользо­вания пер­со­нально­го компьюте­ра.

Прог­рам­мная до­кумен­та­ция:

текст прог­раммы — за­пись прог­раммы с не­об­хо­димы­ми ком­мента­ри­ями;

опи­сание прог­раммы — све­дения о ло­гичес­кой струк­ту­ре и по­ряд­ке ее ра­боты;

программа и ме­тоди­ка ис­пы­таний — тре­бова­ния, под­ле­жащие про­вер­ке при ис­пы­тании прог­раммы, а так­же по­рядок и ме­тоды их кон­тро­ля;

тех­ни­чес­кое за­дание— спе­цифи­кация тре­бова­ний к соз­да­ва­емо­му ПК с точ­ки зре­ния за­каз­чи­ка;

по­яс­ни­тельная за­пис­ка — схе­ма ал­го­рит­ма, об­щее опи­сание ал­го­рит­ма и (или) фун­кцио­ни­рова­ния прог­раммы, а так­же обос­но­вание при­нятых тех­ни­чес­ких и эко­номи­чес­ких ре­шений.

В до­кумен­те «Текст прог­раммы»при­водит­ся сим­во­личес­кая за­пись прог­рамм ком­плек­са на ис­ходных язы­ках, соп­ро­вож­да­емая не­об­хо­димы­ми ком­мента­ри­ями, ко­торые мо­гут от­ра­жать, нап­ри­мер, фун­кци­ональное наз­на­чение и струк­ту­ру.

В раз­де­лах до­кумен­та «Опи­сание прог­раммы»ука­зыва­ет­ся обоз­на­чение, на­име­нова­ние и фун­кци­ональное наз­на­чение прог­раммы, сос­тав ее ПО, язы­ки прог­рамми­рова­ния, на ко­торых на­писа­на прог­рамма, и све­дения о фун­кци­ональных ог­ра­ниче­ни­ях на при­мене­ние прог­раммы.

В раз­де­ле «Опи­сание ло­гичес­кой струк­ту­ры» дол­жны быть ука­заны ал­го­ритм прог­раммы, ис­пользу­емые ме­тоды, струк­ту­ра прог­раммы с опи­сани­ем фун­кций сос­тавных час­тей и свя­зи меж­ду ни­ми, а так­же свя­зи прог­раммы с дру­гими прог­рамма­ми ПК.

В раз­де­ле «Ис­пользу­емые тех­ни­чес­кие средс­тва» ука­зыва­ют­ся ти­пы и па­рамет­ры ус­тройств, ко­торые ис­пользу­ют­ся при ра­боте прог­раммы.

В раз­де­ле «Вы­зов и заг­рузка» ука­зыва­ют­ся спо­соб вы­зова прог­раммы с со­от­ветс­тву­юще­го но­сите­ля дан­ных и точ­ки вхо­да в прог­рамму.

В раз­де­лах «Вход­ные дан­ные» и «Вы­ход­ные дан­ные» да­ют­ся све­дения о под­го­тов­ке дан­ных, опи­сыва­ет­ся их фор­мат и спо­соб ко­диро­вания.

В раз­де­лах до­кумен­та **«**Программа и ме­тоди­ка ис­пы­таний**»** ука­зыва­ют­ся на­име­нова­ние, об­ласть при­мене­ния и обоз­на­чение ис­пы­ту­емо­го ПК, цель про­веде­ния ис­пы­таний, при­водит­ся пе­речень до­кумен­та­ции, предъяв­ля­емой при про­веде­нии ис­пы­таний, тех­ни­чес­кие тре­бова­ния к прог­рам­мной до­кумен­та­ции и тех­ни­чес­ким ха­рак­те­рис­ти­кам.

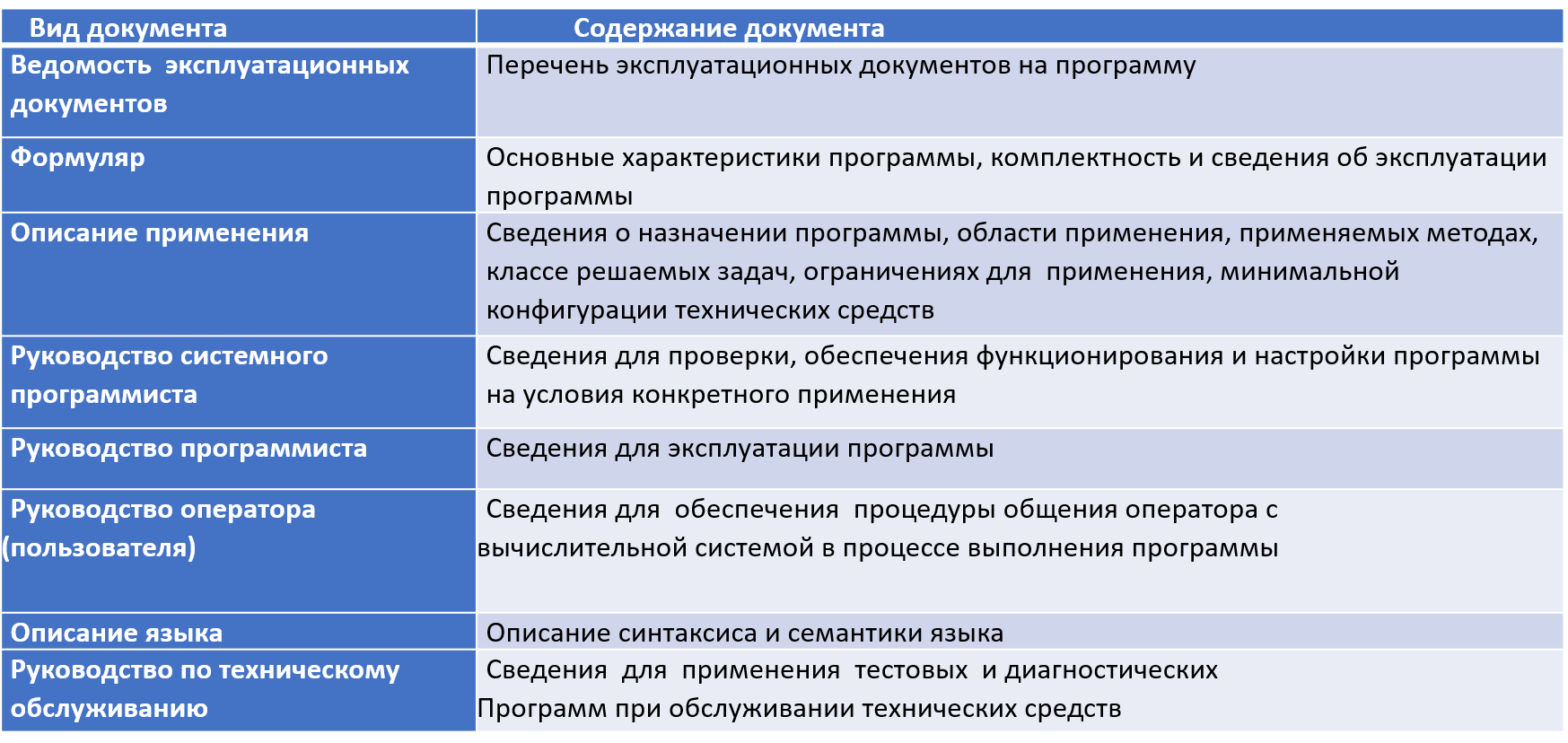
От­дельным пун­ктом опи­сан по­рядок про­веде­ния ис­пы­таний, со­дер­жа­щий пос­ле­дова­тельность ис­пы­таний, сос­тав и струк­ту­ра тех­ни­чес­ких средств.

В зак­лю­чительной час­ти при­водят­ся опи­сания ис­пользу­емых ме­тодов ис­пы­таний по от­дельным по­каза­телям, ко­торые рас­по­лага­ют­ся в по­ряд­ке их сле­дова­ния в раз­де­ле «Тех­ни­чес­кие тре­бова­ния».

В раз­де­лах до­кумен­та **«**По­яс­ни­тельная за­пис­ка**»** кро­ме све­дений, со­дер­жа­щих­ся в дру­гих раз­де­лах, при­водит­ся пе­речень до­кумен­тов, на ос­но­вании ко­торых ве­дет­ся раз­ра­бот­ка, из­ла­га­ет­ся пос­та­нов­ка за­дачи на раз­ра­бот­ку прог­раммы.

Кро­ме то­го, там да­ет­ся опи­сание при­меня­емых ма­тема­тичес­ких ме­тодов и ал­го­рит­ма с обос­но­вани­ем схе­мы его вы­бора, опи­сыва­ют­ся вза­имо­действия прог­раммы с дру­гими прог­рамма­ми, а так­же да­ет­ся обос­но­вание вы­бора ме­тода ор­га­низа­ции вход­ных и вы­ход­ных дан­ных и сос­та­ва тех­ни­чес­ких средств.

Са­мос­то­ятельную груп­пу сос­тавля­ет экс­плу­ата­ци­он­ная до­кумен­та­ция



В до­кумен­те **«**Опи­сание при­мене­ния**»** при­водят­ся раз­де­лы, опи­сыва­ющие наз­на­чение прог­раммы, ус­ло­вия ее при­мене­ния, не­об­хо­димые для ее вы­пол­не­ния, да­ет­ся опи­сание пос­та­нов­ки за­дачи и ме­тодов ее ре­шения, а так­же опи­сыва­ют­ся вход­ные и вы­ход­ные дан­ные.

Ори­гинальная часть до­кумен­та **«**Ру­ководс­тво сис­темно­го прог­раммис­та**»** со­дер­жит све­дения о нас­тройке и про­вер­ке прог­раммы, со­об­ще­ния сис­темно­му прог­раммис­ту и до­пол­ни­тельные воз­можнос­ти прог­раммы. При не­об­хо­димос­ти при­водят­ся по­яс­ня­ющие при­меры и да­ет­ся опи­сание спо­собов про­вер­ки, поз­во­ля­ющих дать об­щее зак­лю­чение о ра­ботос­по­соб­ности прог­раммы (кон­трольные при­меры, ме­тоды про­гона, ре­зульта­ты).

Ори­гинальная часть до­кумен­та **«**Ру­ководс­тво прог­раммис­та**»** со­дер­жит све­дения об об­ра­щении к прог­рамме (про­цеду­ры вы­зова и спо­собы пе­реда­чи уп­равле­ния и па­рамет­ров) и тек­сты ад­ре­сован­ных прог­раммис­ту со­об­ще­ний.

Ру­ководс­тво пользо­вате­ля со­дер­жит све­дения о наз­на­чении прог­раммы, об об­ласти ее при­мене­ния, о при­меня­емых ме­тодах и ог­ра­ниче­ни­ях, кон­фи­гура­ции тех­ни­чес­ких средств и све­дения, не­об­хо­димые для об­ще­ния пользо­вате­ля с пер­со­нальным компьюте­ром в про­цес­се вы­пол­не­ния про­грам­мы.

Дан­ные ру­ководс­тва соз­да­ют­ся на ос­но­ве до­кумен­тов ЕСПД «Опи­сание при­мене­ния» и «Ру­ководс­тво опе­рато­ра».

Эксплуатационный документ **"**Описание языка**"** включается в программную документацию, если разработанный программный продукт реализует некий язык программирования, управления заданиями, организации вычислительного процесса и т. п.

Эксплуатационный документ **"Руководство по техническому обслуживанию"** включается в программную документацию, если разработанный программный продукт требует использования тестовых или диагностических программ.

1. **Понятие совместимости программного обеспечения.**

Концепция программной совместимости

Программная совместимость (ПС) - это если программы, составленные для одной модели, могут выполняться на других моделях ряда.

Программная совместимость обеспечивается единым набором команд, единой формой представления данных и единой системой адресации для всех программных модулей.

Программная совместимость позволяет разрабатывать программы независимо от конкретной ЭВМ, применять общие для большинства ЭВМ ОС.

Кон­цепция прог­рам­мной сов­мести­мос­ти выд­ви­нула в чис­ло при­ори­тет­ных ряд но­вых тре­бова­ний к вновь раз­ра­баты­ва­емым ПО и ап­па­рат­но-прог­рам­мным ком­плек­сам. Прог­рам­мно-сов­мести­мая вы­чис­ли­тельная сре­да дол­жна:

* поз­во­лять гиб­ко ме­нять ко­личес­тво и сос­тав ап­па­рат­ных средств и ПО в со­от­ветс­твии с ме­ня­ющи­мися тре­бова­ни­ями ре­ша­емых за­дач;
* обес­пе­чивать воз­можность за­пус­ка од­них и тех же ПК на раз­личных ап­па­рат­ных плат­формах, обес­пе­чивая мо­бильность ПО;
* га­ран­ти­ровать воз­можность при­мене­ния од­них и тех же че­лове­ко-ма­шин­ных ин­терфейсов на всех компьюте­рах, вхо­дящих в не­од­но­род­ную сеть.

1. **Аппаратная и программная совместимость. Совместимость драйверов.**

Аппаратная совместимость

Аппаратная совместимость означает способность одного компьютера работать с узлами или устройствами, входящими в состав другого компьютера.

Составной частью аппаратной совместимости является электромагнитная совместимость — способность работающих технических средств не создавать взаимных электромагнитных помех, а также функционировать при наличии внешних электромагнитных полей.

Информационная совместимость

Информационная совместимость — это способность двух и более компьютеров или систем адекватно воспринимать одинаково представленные данные.

Частью информационной совместимости, а также средством ее обеспечения является совместимость форматов представления данных.

Программная совместимость

Программная совместимость — это возможность выполнения одних и тех же программ на разных компьютерах с получением одинаковых результатов.

В отличие от программной совместимости термин «совместимость программ» означает пригодность программ к взаимодействию друг с другом и, в частности, к объединению в персональном компьютере для решения более сложных задач, например в автоматизированных системах.

Полная совместимость — это аппаратная, программная и информационная совместимость двух или более компьютеров без каких-либо ограничений для их пользователей.

Аппаратная платформа

Развитием термина «аппаратная совместимость» является термин «аппаратная платформа».

В области персональных компьютеров сегодня наиболее широко распространены две аппаратные платформы — IBM РС и Аррlе Macintosh.

Обеспечение совместимости компьютерных платформ.

В настоящее время существуют два основных варианта решения этой проблемы: *аппаратные и программные*.

Аппаратные решения заключаются в использовании печатных модулей (плат), содержащих дополнительный микропроцессор и «лишнюю» оперативную и видеопамять от другой аппаратной платформы.

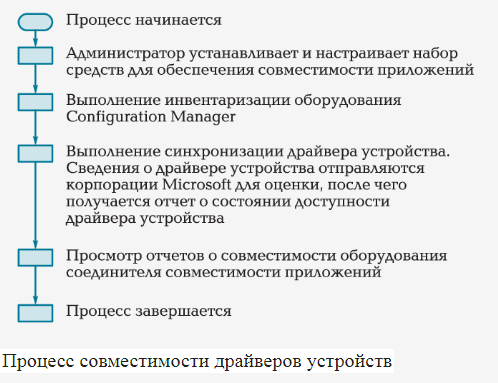
Данную конструкцию можно оснастить второй ОС (по

выбору пользователя) и осуществлять обмен файлами между установленными ОС.

Программные решения заключаются в  написании программ эмуляторов, позволяющих запускать ПО, разработанное для ПЭВМ одного типа, на другой ПЭВМ.

Программа-эмулятор представляет собой программу, выполняющую каждую команду эмулируемой программы посредством одной или нескольких команд той ПЭВМ, на которой реализуется эмуляция.

К программным решениям совместимости компьютерных платформ относится также обеспечение совместимости драйверов — частей кода ОС, отвечающих за взаимодействие ПО с физической и (или) виртуальной аппаратурой.



Раз­ли­ча­ют од­но- и мно­го­уров­не­вые драйве­ры.

У многоуровневых драйверов об­ра­бот­ка зап­ро­сов вво­да-вы­вода рас­пре­деля­ет­ся меж­ду от­дельны­ми драйве­рами, каж­дый из ко­торых вы­пол­ня­ет свою часть ра­боты.

При об­ра­бот­ке зап­ро­са дан­ные идут от драйве­ров вер­хне­го уров­ня к драйве­рам ниж­не­го, а при ис­полне­нии зап­ро­са — в об­ратном нап­равле­нии.

Для технологии Plug-n-Play существуют три уровня типа драйверов:

1. шинные драйверы — обслуживают контроллер шины, мосты, причем для каждого типа шины в системе имеется собственный шинный драйвер;
2. фильтр-драйвер — сортирует запросы ввода-вывода для шины и других шинных устройств и находится на различных уровнях (как выше, так и ниже шинного драйвера);
3. функциональный драйвер — является обязательным и основным драйвером устройства, для которого он реализует интерфейс.

По структуре драйвер представляет собой обычную динамически подключаемую DLL-библиотеку уровня ядра ОС, дополняя его набором процедур, периодически вызываемых внешними программами.

Несмотря на то, что процедуры драйверов для разных устройств разные, сохраняются общая структура их написания и последовательность выполняемых действий.

1. **Причины возникновения проблем совместимости. Методы выявления проблем совместимости ПО.**

Причины возникновения проблем совместимости программного обеспечения

Прог­рам­мные не­ис­прав­ности компьюте­ра встре­ча­ют­ся нам­но­го ча­ще, чем ап­па­рат­ные.

Ос­новные при­чины воз­никно­вения прог­рам­мных оши­бок — это не­совер­шенс­тво ПО и ОС.

Мож­но наз­вать нес­колько при­чин по­яв­ле­ния прог­рамм низ­ко­го ка­чес­тва.

Нап­ри­мер, ес­ли при сво­ей ра­боте прог­рамма зах­ва­тыва­ет ре­сур­сы, кон­тро­лиро­вать ко­торые она не спо­соб­на, по­яв­ля­ет­ся ПО, име­ющее мно­жес­тво не­доче­тов и пос­то­ян­но вы­да­ющее ошиб­ки.

Серьезные проблемы возникают также, если программа

пытается эксплуатировать установленные в ОС системные библиотеки, не учитывая, что они постоянно обновляются корпорацией, поставляющей данную ОС.

Такие программы отказываются работать с улучшенными версиями.

Опытные программисты умеют адаптировать программу для работы в разных условиях, что делает ее менее уязвимой.

Часто для успешной работы квалифицированно составленной программы требуется большее количество ресурсов, чем ей может предоставить ОС.

В этом случае для обеспечения работы программы система увеличивает файл подкачки, отбирающий оперативную память у других программ, что приводит не только к снижению производительности компьютера.

Проблему обычно решает увеличение объема оперативной памяти

Серьезно нарушить стабильность работы ОС могут непрофессиональные действия программиста с реестром ОС, которые прописывают свои файлы и ссылки в самых различных его разделах.

Ошибки в реестре негативно сказываются на всех процессах, происходящих в компьютере.

Проблема совместимости ПО чаще всего возникает,

когда приложение вольно обращается к функциям ОС.

Ведущие производители ПО стараются своевременно выпускать новые версии программного продукта, совместимые с действующими ОС, однако организации, эксплуатирующие ОС, иногда вынуждены использовать ПО, не полностью совместимое с новой версией ОС.

В лучшем случае пользователь получает предупреждения о несовместимости, в худшем — машина просто перестает реагировать на запросы пользователя.

Методы выявления проблем совместимости программного обеспечения.

Для устранения подобных проблем необходимо при установке/обновлении новой/установленной ОС выполнить на ПЭВМ предприятия следующие этапы:

*1) проверить используемые на предприятии программы на совместимость с новой ОС и определить те, у которых возникли проблемы совместимости;*

*2) обновить версии проблемных программ, а если в них предприятие больше не нуждается — удалить из ОС.*

Наиболее распространенным методом устранения проблем совместимости ПО является использование динамически загружаемых библиотек.

Динамические исполняемые библиотеки (DLL) представляют собой специальные программные компоненты, которые позволяют обращаться к одному файлу сразу нескольким приложениям. Основным предназначением таких библиотек является расширение возможностей утилит и экономия места, поскольку для использования аналогичных компонентов можно иметь всего одну версию файла, которая будет делиться между всеми программами

DLL – это файлы системных библиотек в Windows. Формально они являются исполняемыми, но запустить их по отдельности не получится.

Библиотеки используются другими программами для выполнения стандартных функций сохранения, удаления, печати.

Методы выявления проблем совместимости программного  
обеспечения

Следующим методом устранения проблем совместимости ПО

является применение группы методов специального тестирования ПО, к которым относятся:

* инсталляционное тестирование;
* регрессионное тестирование;
* тестирование функциональности;
* конфигурационное тестирование;
* тестирование совместимости;
* тестирование интерфейса человек/машина

Инсталляционное тестирование

В процессе инсталляционного тестирования программ проверяется корректность их установки и удаления в среде, максимально приближенной к эксплуатационной.

Проверка правильности установки должна быть обязательным элементом проекта по тестированию любого продукта

Регрессионное тестирование;Тестирование функциональности

Регрессионное тестирование подразумевает повторное выполнение тестов для проверки того, что текущая версия программы ведет себя идентично предыдущей, за исключением измененных областей.

В отличие от него тестирование функциональности

делает акцент на проверку выполнения программой новых функций, появившихся в конкретном выпуске программного продукта.

Конфигурационное тестирование

С помощью конфигурационных тестов проверяется совместимость продукта с различным программным и аппаратным обеспечением.

При разработке программный продукт настраивается на работу в максимально разнообразной внешней среде.

Для того чтобы выяснить реакцию продукта на окружение и соседство с другим ПО, и проводят данные тесты.

Тестирование совместимости; Тестирование интерфейса человек/машина

Тестирование совместимости помогает убедиться в функциональных возможностях и надежности работы продукта в поддерживаемых браузерах и ОС.

Также может проверяться работоспособность продукта при использовании различных аппаратных платформ.

Тестирование интерфейса человек/машина

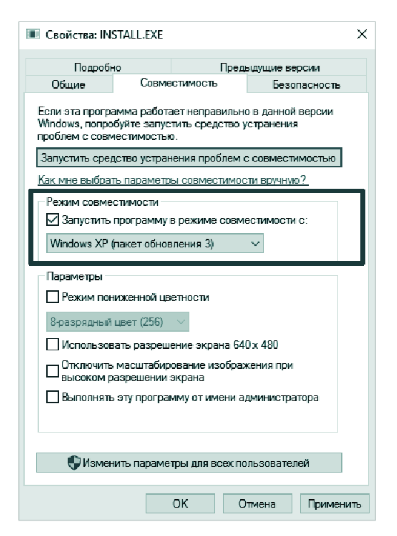
Производится в отношении вида пользовательского интерфейса и удобства навигации по различным браузерам (если навигация предусмотрена в программе).

Режим совместимости Windows10

Проблему решает «штатный» режим совместимости Windows10, позволяющий запустить старую программу, нормально работающую в одной из предыдущих версий Windows (XP, Vista или Windows 7), обеспечивая ее стабильную работу в новой версии — Windows 10.

Для запуска старого приложения в режиме совместимости необходимо:

* в контекстном меню ярлыка вызываемой программы выбрать пункт Свойства, откроется форма настройки параметров приложения;
* на появившейся форме выбрать вкладку Совместимость, включить режим совместимости, установив галочку в первом поле сверху;



1. **Выполнение чистой загрузки.**

Чистая загрузка Windows

* Под термином «чистая загрузка Windows» понимается загрузка ОС Windows исключительно с теми программами и драйверами, которые поддерживают ее нормальное функционирование.
* Чистая загрузка обеспечивает бесконфликтную работу ОС и применяется в тех случаях, когда пользователь не может определить причину сбоев ОС при ее обычной загрузке.
* Данная функция предусмотрена в ОС компании Microsoft, начиная с Windows 7.

Безопасный режим

Чистая загрузка Windows отличается от режима «безопасный», используемого, например, при возникновении проблем с поврежденными драйверам, из-за которых не удается запустить компьютер в стандартном режиме.

В безопасном режиме ОС Windows отключает все сторонние драйверы, которые могут помешать ее загрузке, и администратор получает возможность восстановить нормальную систему, удалив поврежденный драйвер

Выполнение

* Чтобы произвести старт, необходимо выполнить следующие действия:
* нажать сочетание клавиш [Windows + R] — откроется строка Выполнить;
* в строке окна Выполнить ввести: msconfig — откроется окно изменения конфигураций системы
* на вкладке Общие активировать Выборочный запуск и отключить пункт Загружать элементы автозагрузки
* перейти на вкладку Службы и установить в левом нижнем углу флаг напротив пункта Не отображать службы Microsoft, чтобы случайно не выключить важные для загрузки системы службы;
* нажать кнопку Отключить все — все службы от сторонних разработчиков будут деактивированы
* для отключения программ в меню Автозагрузка перейти на вкладку Автозагрузка и нажать кнопку Открыть диспетчер задач, который загрузится на вкладке с автоматически активируемыми при включении компьютера программами;
* выделить здесь по очереди все приложения и нажать кнопку Отключить и после реального отключения закрыть диспетчер задач и нажать кнопку ОК в окне изменений конфигураций системы;

«штатными» средствами перезагрузить ОС в настроенном только что  
режиме Чистая загрузка ОС.

1. **Выявление причин возникновения проблем совместимости ПО. Выбор методов выявления совместимости.**

Проблемы совместимости

Обычно приложения и аппаратное обеспечение, работающее на Windows 7, продолжает нормально работать и на Windows 10, но иногда возникают определенные проблемы.

Основные причины того, из-за чего это происходит:

1. По­мешать ус­та­нов­ке и за­пус­ку при­ложе­ния мо­гут две при­чины:

* приложения пытаются копировать файлы и ярлыки, созданные в предыдущей ОС Windows, но не существующие в новой ОС;
* приложения пытаются получить доступ к компонентам Windows, которые были переименованы.

2. Проблемы совместимости могут возникнуть, если включен

* контроль учетной записи пользователей (UAC). UAC увеличивает безопасность Windows, ограничивая доступ к компьютеру без уровня администратора, что ограничивает запуск приложений большинству пользователей в качестве обычных пользователей.

3. Папки и файлы ОС защищены системной программой от перезаписи, поэтому установщики приложений, пытавшиеся их изменить, «натыкаются» на сообщение об ошибке, указывающем на невозможность обновления ресурса.

4. Windows 7 полностью поддерживает 64-битную архитектуру. Приложения или компоненты, использующие 16-битные исполняемые файлы, 16-битные установщики или 32-битные драйвера ядра, могут вызвать сбой при запуске или будут неправильно функционировать.

5. Если пользователь в своей системе использует предыдущую версию интерфейса прикладного программирования (API), то у него могут возникнуть сбои при работе таких приложений, как сканеры сети, антивирусы и фаерволы.

(Мастер совместимости программ

Инструментом решения проблем совместимости приложений является Мастер совместимости программ. Его графический интерфейс входит в состав Центра справки и поддержки Windows.)

Выбор методов решения проблем совместимости на основе анализа

После определения несовместимых приложений необходимо принять попытку по полному устранению проблем или, хотя бы, по уменьшению последствий.

Для обеспечения совместимости приложений необходимо выполнить:

* + Определить наиболее значимые приложения.
  + Определить проблемы совместимости.

Уменьшение проблем с совместимостью приложения обычно зависит от различных факторов, таких как тип приложения и текущей поддержки приложения.

Общие методы уменьшение проблем с совместимостью приложений включают:

1. Изменение конфигурации существующего приложения: Вы можете использовать инструменты, Compatibility Administrator или Standard User Analyzer, для обнаружения проблемы и создания исправления данного приложения, что решит проблему совместимости
2. Применение обновлений или пакетов обновлений ОС: это может помочь решить многие из проблем с совместимостью и дать возможность приложению работать в новой среде операционной системы.
3. Апгрейд приложения до совместимого релиза: если более новая, совместимая версия приложения существует, лучшее решение — обновить до более новой версии.
4. Изменение конфигурации безопасности: как пример, Защищенный режим Internet Explorer может быть смягчен, добавив сайт в список надежных сайтов или выключив Защищенный режим (что не рекомендуется).
5. Запуск приложения в виртуализированной среде: если все другие методы недоступны, для решения проблем вы можете запустить приложение в более раннем релизе Windows, используя инструменты виртуализации, такие как PC Microsoft Virtual, Microsoft Virtual Server и т.п.
6. Использование функций совместимости приложения: проблемы приложения, такие как управление версиями операционной системы, могут быть смягчены, запуском приложения в режиме эмуляции (режим совместимости с предыдущей версией ОС).
7. Выбор другого приложения, которое выполняет ту же самую функцию, но не имеет проблем с совместимостью: если другое совместимое приложение доступно, вы можете использовать его.
8. **Проблемы перехода на новые версии программ. Мастер совместимости программ.**

Проблемы перехода на новые версии программ

1.Апгрейд (Upgrade).

На данном этапе происходит обмен одного из имеющихся у пользователя продукта на новый. Это предложение действует только для обладателей лицензионной системы. «Старая» программа остается у клиента и ею можно пользоваться во время «переходного» периода.

2. Установка.

Предусматривает проведение самих работ, результат – рабочая новая версия ПО, которую можно сразу же использовать.

3. Перенос данных.

Необходимый этап в рамках, которого информация со старой версии переносится на новую.

4. Освоение.

Научиться работать с новой системой можно:

самостоятельно, для этого выпускается специализированная литература;

в специально созданных учебных центрах;

индивидуально, с квалифицированным специалистом

Безопасность

Окончание поддержки версии Windows также означает прекращение выпуска обновлений безопасности.

ПК, которые на момент окончания поддержки продолжат использовать

эту ОС, окажутся гораздо более уязвимыми к угрозам, вирусам, шпионскому ПО и другим вредоносным действиям***.***

Всего одна атака может подвергнуть опасному риску интеллектуальную собственность организации, данные клиентов, важную персональную информацию, повседневную деятельность и на долгое время привести к возбуждению исков и ущербу для репутации компании

Увеличивающийся уровень сложности

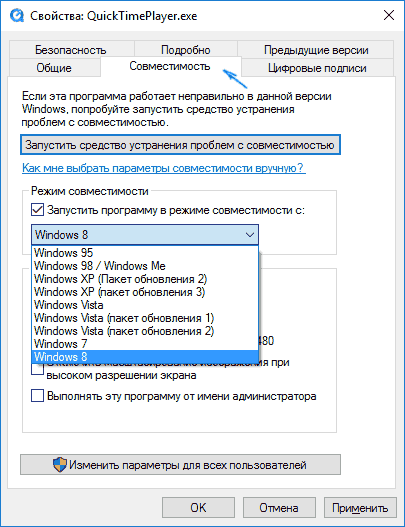
* Многочисленные зависимости от платформ и технологии других компаний, обычно используемые в среде предприятия, потенциально усложняют предстоящий переход к более поздним версиям ПО (Windows), а уровень сложности будет возрастать по мере переноса этих элементов инфраструктуры в среды, несовместимые с Windows
* Необходимость разрешения этих проблем, а также проблем, связанных с вопросами *безопасности и совместимости*, будет делать работу ИТ-отделов все более и более сложной и время затратной и отнимать как кадровые, так и финансовые ресурсы.

Мастер совместимости программ

* Режим совместимости программ Windows 10 позволяет запускать ПО на компьютере, которое нормально работало только в предыдущих версиях Windows, а в последней ОС программа не запускается или работает с ошибками. В этой инструкции о том, как включить режим совместимости с Windows 8, 7, Vista или XP в Windows 10 для исправления ошибок запуска программ.

Включение режима совместимости через свойства программы или ярлыка

* Кликните правой кнопкой мыши по ярлыку или исполняемому файлу программы, выберите пункт «Свойства» и откройте, если таковая обнаружится, вкладку «Совместимость».
* Все, что останется сделать — задать параметры режима совместимости: указать версию Windows, в которой программа запускалась без ошибок. При необходимости, включите запуск программы от имени администратора или в режиме меньшего разрешения экрана и пониженной цветности (для совсем старых программ). Затем примените сделанные настройки. В следующий раз программа будет запущена с уже измененными параметрами.



1. **Инструментарий учета аппаратных компонентов (GPT)**

Инструментарий учета аппаратных компонентов включает в себя различные методы и инструменты, предназначенные для отслеживания и управления аппаратным обеспечением в организации. Этот инструментарий помогает в поддержании актуальной информации о наличии, состоянии и расположении оборудования, а также в оптимизации его использования и обеспечения безопасности.

Ниже приведены основные элементы такого инструментария:

1.Системы управления инвентаризацией (Inventory Management Systems)

* Программное обеспечение для учета аппаратного обеспечения: такие системы позволяют автоматизировать процесс учета оборудования, обеспечивая удобный интерфейс для внесения данных, поиска и отчетности.

2.Маркировка и идентификация оборудования

* Штрих-коды: используются для быстрого сканирования и идентификации оборудования с помощью сканеров штрих-кодов.
* QR-коды: предоставляют быстрый доступ к информации об оборудовании с помощью смартфонов и планшетов.

3. Периодические проверки и инвентаризация

* Регулярные аудиты: проведение регулярных проверок для сверки фактического наличия оборудования с учетными данными.
* Системы напоминаний: использование напоминаний и планировщиков для организации периодических проверок и обслуживания оборудования.

4. Мониторинг состояния оборудования

* Сенсоры и датчики: установка сенсоров для мониторинга параметров оборудования, таких как температура, влажность и уровень износа.
* ПО для мониторинга: специализированные программные решения для анализа данных с сенсоров и создания отчетов о состоянии оборудования.

5. Управление жизненным циклом оборудования

* Реестр оборудования: ведение реестра с полной информацией о каждом компоненте, включая дату приобретения, спецификации, историю обслуживания и ремонтов.
* Планирование замены: разработка планов по замене устаревшего или изношенного оборудования для предотвращения простоев.

6. Безопасность и защита

* Контроль доступа: системы контроля доступа для предотвращения несанкционированного использования и краж оборудования.
* Шифрование и защита данных: обеспечение защиты данных на уровне оборудования, особенно для устройств хранения данных.

7. Отчетность и аналитика

* Генерация отчетов: создание регулярных отчетов о состоянии, использовании и износе оборудования для анализа и принятия решений.
* Аналитические инструменты: использование аналитических инструментов для прогнозирования потребностей в оборудовании и оптимизации его использования.

Использование данного инструментария позволяет компаниям эффективно управлять своим аппаратным обеспечением, минимизировать риски и затраты, а также повышать общую эффективность и безопасность работы.

1. **Анализ приложений с проблемами совместимости.**

Анализ приложений с проблемами совместимости

* Основная причина неработоспособности программных продуктов на разных ОС – это разные рабочие библиотеки системы.
* В каждой операционной системе они имеют разное размещение директорий (да и содержимого в конечном итоге).
* Самый распространенный метод решения проблем совместимости – это использование DLL (динамически загружаемых библиотек), которые перехватывают «проблемные» вызовы системных функций.
* Эти библиотеки нередко называют «системными заплатками».

Статические и динамические библиотеки

Библиотеками называют «сборники» подпрограмм или объектов, ориентированных на решение набора близких по тематике задач.

С точки зрения их организации и использования библиотеки бывают статическими и динамическими.

Статические (LIB) библиотеки

Статические (LIB) библиотеки — это набор исходных кодов, подключаемых программистом в свою программу, либо набор заранее скомпилированных объектных файлов, связываемых вместе на этапе компиляции.

В результате связывания со статической библиотекой программа включает в себя все используемые ей функции, что увеличивает ее размер, но делает более автономной.

Динамические (DLL) библиотеки

Динамические (DLL) библиотеки загружаются ОС по «требованию» запущенной программы уже в ходе ее выполнения.

При этом один и тот же набор функций или объектов библиотеки может быть использован одновременно несколькими работающими программами, что позволяет эффективно использовать ресурсы оперативной памяти.

1. **Использование динамически загружаемых библиотек.**

Динамические (DLL) библиотеки

Динамические (DLL) библиотеки загружаются ОС по «требованию» запущенной программы уже в ходе ее выполнения.

При этом один и тот же набор функций или объектов библиотеки может быть использован одновременно несколькими работающими программами, что позволяет эффективно использовать ресурсы оперативной памяти.

Для использования библиотеки ее необходимо подключить, после чего вызвать из нее требуемую функцию, при этом исходный текст функции не нужен.

Создание динамической библиотеки можно выполнить в среде Microsoft Visual Studio, выбрав в Мастере создания проекта пункт DLL

Использовать динамическую библиотеку в программе можно либо неявным, либо явным связыванием.

Неявное связывание подразумевает использование библиотеки импорта для определения адресов функций, предоставляемых библиотекой.

Операционная система загружает библиотеку DLL после загрузки исполняемого файла программы. Исполняемый файл вызывает экспортированные функции библиотеки DLL таким способом, как если бы функции содержались в самом исполняемом файле.

При явном связывании исполняемый файл, использующий библиотеку DLL, должен делать вызовы функции для явной загрузки и выгрузки библиотеки DLL и осуществления доступа к экспортированным функциям библиотеки DLL.

Клиентский исполняемый файл вызывает экспортированные функции с помощью указателя функции.

Приложения, работающие в ОС Windows, состоят из

исполняемых файлов и динамических библиотек (Dynamic Link

Library —DLL).

Динамические библиотеки представляют собой исполняемые

модули, содержащие функции и данные, которые могут быть

использованы другими модулями (приложениями или

динамическими библиотеками).

С использованием динамических библиотек создаются

приложения на основе модульного принципа, вследствие чего

эти приложения могут быстрее загружаться и проще

обновляться.

Кроме того, если некоторые приложения используют

одну и ту же динамическую библиотеку, то эта DLL

загружается в оперативную память только один раз и доступ

к ней разрешается всем заинтересованным в ее использовании

приложениям.

Набор API-функций, с помощью которых пользовательское приложение взаимодействует с ОС, реализован как набор динамических библиотек, поэтому все приложения Windows используют динамические библиотеки.

Динамическое связывание позволяет включать

в модули только информацию об экспортируемой в момент загрузки или во время выполнения программы функции DLL.

Применяют два метода вызова функции из DLL.

1.Динамическое связывание во время загрузки (Load-Time Dynamic Linking).

В этом случае модуль осуществляет явный вызов экспортируемой функции так, будто она является его локальной функцией.

Этот способ требует наличия библиотеки импорта, связанной с DLL, содержащей экспортируемые функции, которая обеспечивает ОС информацией, необходимой для загрузки этой DLL, и расположением экспортируемых функций внутри DLL.

2. Динамическое связывание во время выполнения программы (Run-Time Dynamic Linking).

В этом случае вызывающее приложение должно использовать специальные API-функции для загрузки DLL, выгрузки DLL и получения адреса экспортируемой функции.

Самостоятельное получение адреса экспортируемой функции приводит к тому, что нет необходимости создавать библиотеку импорта.

Динамические библиотеки не отличаются от обычных программ.

Разница заключается лишь в том, что функции, определенные в динамических библиотеках, могут вызываться другими программными модулями.

1. **Механизм решения проблем совместимости на основе «системных заплаток».**

Механизм решения проблем совместимости на основе системных заплаток.

Системные заплатки (patches) — это изменения в программном коде, которые вносятся разработчиком или поставщиком операционной системы для решения проблем совместимости. Эти заплатки могут быть представлены в виде исправлений, обновлений или обратной совместимости. Механизм решения проблем совместимости на основе системных заплаток состоит из следующих шагов:

1. Выявление и анализ проблемы совместимости

Шаг 1: Мониторинг и сбор данных

**Мониторинг систем**: Использование систем мониторинга (например, Nagios, Zabbix) для выявления отклонений в работе оборудования и ПО.

**Сбор отчетов об ошибках**: Анализ логов систем, отчетов об ошибках и жалоб пользователей.

**Обратная связь от пользователей**: Получение информации о проблемах от пользователей через системы тикетов и поддержки.

Шаг 2: Анализ проблемы

**Идентификация проблемы**: Определение конкретного характера проблемы совместимости и обстоятельств, при которых она возникает.

**Диагностика причин**: Выяснение корневой причины проблемы через анализ системных логов, трассировку выполнения программ и анализ конфигураций.

**Определение затронутых компонентов**: Определение, какие компоненты системы (оборудование, драйверы, ПО) взаимодействуют некорректно.

2. Разработка и тестирование системной заплатки для решения проблемы

Шаг 1: Разработка заплатки

**Проектирование решения**: Разработка концептуального решения для устранения проблемы совместимости.

**Написание кода**: Создание кода заплатки, который модифицирует проблемные компоненты системы для устранения несовместимости.

**Документирование**: Создание документации, описывающей изменения и методику их внедрения.

Шаг 2: Тестирование заплатки

**Первичное тестирование**: Тестирование заплатки в контролируемой среде, чтобы убедиться в ее работоспособности.

**Регрессионное тестирование**: Проверка, что внесенные изменения не нарушают существующую функциональность системы.

**Тестирование на разных конфигурациях**: Тестирование заплатки на различных аппаратных и программных конфигурациях, чтобы убедиться в ее универсальности и эффективности.

3. Распространение и установка заплатки на целевую платформу

Шаг 1: Подготовка к распространению

**Создание установочных пакетов**: Подготовка инсталляторов и пакетов для различных операционных систем и конфигураций.

**Планирование развертывания**: Определение этапов развертывания и приоритетных систем для установки заплатки.

Шаг 2: Развертывание заплатки

**Поэтапное развертывание**: Начало с небольшой группы тестовых систем или пользователей, чтобы минимизировать риск массовых проблем.

**Обратная связь**: Сбор отзывов и данных о работе системы после установки заплатки на тестовые системы.

4. Проверка корректности и эффективности заплатки

Шаг 1: Мониторинг после развертывания

**Непрерывный мониторинг**: Отслеживание систем для выявления любых новых или повторяющихся проблем.

**Анализ производительности**: Проверка, что производительность системы не ухудшилась после установки заплатки.

Шаг 2: Оценка и отчетность

**Оценка эффективности**: Анализ данных для оценки, насколько заплатка решила проблему и улучшила совместимость.

**Документирование результатов**: Создание отчетов, фиксирующих результаты развертывания и эффективность заплатки.

**Подготовка к широкому развертыванию**: Если заплатка показала свою эффективность на тестовых системах, ее можно развернуть на все затронутые системы.

1. **Разработка модулей обеспечения совместимости.**

Правила создания программ

1. к разработке программ необходимо привлекать команду программистов;
2. программу необходимо разбить на несколько независимых программных модулей, каждый из которых должен выполнять определенную функцию;
3. необходимо обеспечить совместимость программных модулей. Разбиение программы на модули — типичная задача обеспечения совместимости.

Технология реализации модульного принципа программирования.

1. Задача разбивается на относительно простые и  самостоятельные фрагменты.Критериями качества разбиения являются:

* решение своей задачи до конца каждым выделяемым фрагментом;
* возможность запрограммировать модуль в заданный срок;
* минимум связей с остальными модулями;
* проверяемость входных данных и результатов выполнения модуля. К правильности и полноте входных данных должны быть сформулированы четкие требования, выполнение которых должно быть проверено на входе модуля. Ситуации получения неверных данных должны быть учтены и обработаны. Решая определенную задачу до конца, модуль должен выдавать результаты, правильность которых также можно логически проверить

2. Формируются требования к модулям.

* Для каждой конкретной задачи выполняемой программы (варианта использования) должна существовать цепочка последовательно выполняемых модулей, полностью решающая данную задачу.
* В простейшем случае цепочка может состоять из одного модуля.

3. Разрабатывается межмодульный интерфейс — правила вызова модулей, передачи им параметров и данных.

Информация, необходимая для работы большинства модулей, организуется в виде общих областей, глобальных переменных, массивов или баз данных.

После завершения третьего этапа формируются задания на программирование.

4. Проводится комплексная отладка (после написания отдельных модулей). В ходе отладки проверяется возможность совместной работы модулей, правильность решения всех задач программного продукта.

Создание программных модулей.

Это процесс реорганизации программы в целях объединения ее взаимосвязанных частей в отдельном модуле.

После этого легче удалить избыточность в соответствующих компонентах, оптимизировать взаимосвязи и упростить интерфейс всей программы

1. **Создание в системе виртуальной машины для исполнения приложений.**

Виртуализация подразумевает запуск на одном физическом компьютере нескольких виртуальных компьютеров

Возможность запуска нескольких виртуальных машин на одной физической повышает гибкость ИТ-инфраструктуры, а также позволяет экономить деньги.

Microsoft и VMware, Inc. — обе компании производят средства управления виртуальными машинами и для рабочих станций, и для серверов.

Преимущества виртуализации

1. Эффективное использование вычислительных ресурсов.

Вместо 3х, а то 10 серверов, загруженных на 5-20% можно использовать один, используемый на 50-70%.

2.Сокращение расходов на инфраструктуру:

Виртуализация позволяет сократить количество серверов и связанного с ними ИТ-оборудования в информационном центре.

3. Снижение затрат на программное обеспечение.

4. Повышение гибкости и скорости реагирования системы:

Виртуализация предлагает новый метод управления ИТ-инфраструктурой и помогает ИТ-администраторам затрачивать меньше времени на выполнение повторяющихся заданий — например, на инициацию, настройку, отслеживание и техническое обслуживание.

5. Несовместимые приложения могут работать на одном компьютере.

При использовании виртуализации на одном сервере возможна установка linux и windows серверов, шлюзов, баз данных и прочих абсолютно несовместимых в рамках одной не виртуализированной системы приложений.

6. Повышение доступности приложений и обеспечение непрерывности работы предприятия:

Благодаря надежной системе резервного копирования и миграции виртуальных сред целиком без перерывов в обслуживании вы сможете сократить периоды планового простоя и обеспечить быстрое восстановление системы в критических ситуациях.

7.Возможности легкой архивации.

Поскольку жесткий диск виртуальной машины обычно представляется в виде файла определенного формата, расположенный на каком-либо физическом носителе, виртуализация дает возможность простого копирования этого файла на резервный носитель как средство архивирования и резервного копирования всей виртуальной машины целиком.

Возможность поднять из архива сервер полностью еще одна замечательная особенность.

8. Повышение управляемости инфраструктуры

Использование централизованного управления виртуальной инфраструктурой позволяет сократить время на администрирование серверов.

Виртуальная машина — это полностью изолированный программный контейнер, который работает с собственной ОС и приложениями, подобно физическому компьютеру.

Виртуальная машина действует так же, как физический компьютер, и содержит собственные виртуальные (т.е. программные) ОЗУ, жесткий диск и сетевой адаптер

Основные особенности виртуальных машин

* Совместимость. Виртуальные машины, как правило, совместимы со всеми стандартными компьютерами.
* Изолированность. Виртуальные машины полностью изолированы друг от друга, как если бы они были физическими компьютерами Виртуальные машины могут использовать общие физические ресурсы одного компьютера и при этом оставаться полностью изолированными друг от друга, как если бы они были отдельными физическими машинами.
* Инкапсуляция. Виртуальные машины полностью инкапсулируют вычислительную среду. Виртуальная машина представляет собой программный контейнер, связывающий, или "инкапсулирующий" полный комплект виртуальных аппаратных ресурсов, а также ОС и все её приложения в программном пакете. Благодаря инкапсуляции виртуальные машины становятся невероятно мобильными и удобными в управлении.
* Независимость от оборудования. Виртуальные машины полностью независимы от базового физического оборудования, на котором они работают.

Основные разновидности виртуализации

* виртуализация серверов (полная виртуализация и паравиртуализация)
* виртуализация на уровне операционных систем,
* виртуализация приложений,
* виртуализация представлений.

Многие трудности и проблемы разработки технологий виртуализации связаны с преодолением унаследованных особенностей программно-аппаратной архитектуры x86. Для этого существует несколько базовых методов:

Полная виртуализация (Full, Native Virtualization). Используются не модифицированные экземпляры гостевых операционных систем.

К достоинствам данного подхода можно причислить относительную простоту реализации, универсальность и надежность решения; все функции управления берет на себя хост-ОС.

Недостатки — высокие дополнительные накладные расходы на используемые аппаратные ресурсы, отсутствие учета особенностей гостевых ОС, меньшая, чем нужно, гибкость в использовании аппаратных средств

Паравиртуализация (paravirtualization). Модификация ядра гостевой ОС выполняется таким образом, что в нее включается новый набор API, через который она может напрямую работать с аппаратурой, не конфликтуя с другими виртуальными машинами.

Достоинства данной технологии заключаются в отсутствии потребности в хостовой ОС – ВМ, устанавливаются фактически на "голое железо", а аппаратные ресурсы используются эффективно.

Недостатки — в сложности реализации подхода и необходимости создания специализированной ОС-гипервизора.

Виртуализация на уровне ядра ОС (operating system-level virtualization).

Этот вариант подразумевает использование одного ядра хостовой ОС для создания независимых параллельно работающих операционных сред.

Достоинства — высокая эффективность использования аппаратных ресурсов, низкие накладные технические расходы, отличная управляемость, минимизация расходов на приобретение лицензий.

Недостатки — реализация только однородных вычислительных сред.

Виртуализация приложений подразумевает применение модели сильной изоляции прикладных программ с управляемым взаимодействием с ОС, при которой виртуализируется каждый экземпляр приложений, все его основные компоненты: файлы (включая системные), реестр, шрифты, INI-файлы, COM-объекты, службы.

Виртуализация представлений (рабочих мест)

Виртуализация представлений подразумевает эмуляцию интерфейса пользователя. Т.е. пользователь видит приложение и работает с ним на своём терминале, хотя на самом деле приложение выполняется на удалённом сервере, а пользователю передаётся лишь картинка удалённого приложения.

1. **Изменение настроек по умолчанию в образе. Подключение к сетевому ресурсу.**

Изменение настроек по умолчанию в образе.

-Система обслуживания образов развертывания и управления ими (DISM) может использоваться для изменения в образе Windows(R) программ по умолчанию.

-Создание XML-файла для сопоставлений приложений по умолчанию.

Для создания XML-файла необходимо развернуть образ Windows на тестовом компьютере и настроить программы, включенные в данный образ.

-Можно войти в Windows и выбрать сопоставления приложений по умолчанию с помощью панели управления.

-Настройки сопоставлений приложений по умолчанию можно экспортировать в XML-файл в сетевом ресурсе или на съемном носителе и импортировать в WIM- или VHD-файл перед его развертыванием на целевых компьютерах.

Настройка сопоставлений приложений по умолчанию.

В этом случае необходимо выполнить следующие действия:

1) установить образ Windows на тестовом компьютере;

2) запустить тестовый компьютер и выполнить установку Windows;

3) щелкнуть Поиск, выбрать Настройки, затем ввести Программы по умолчанию и выбрать Программы по умолчанию.

Утилита DISM — незаменимое консольное приложение, позволяющее производить множество различных манипуляций над образами ОС. Причем поддерживаются как online-, так и offline образы.

Поддержка online-образов означает, что можно производить установку необходимых компонентов непосредственно из запущенной ОС

DISM позволяет выполнять следующие действия:

* добавление, удаление и перечисление пакетов;
* добавление, удаление и перечисление драйверов;
* включение и отключение компонентов Windows;
* настройка региональных параметров;
* модернизация Windows до другого выпуска;
* улучшенное ведение журнала;
* обслуживание предыдущих версий Windows, таких как Windows 8 и Windows Server 2008;
* обслуживание 32-разрядного образа с 64-разрядного компьютера и обслуживание 64-разрядного образа с  32-разрядного компьютера;
* использование старых сценариев диспетчера пакетов.

К возможностям утилиты DISM относят:

* добавление, удаление и перечисление INF-файлов драйверов;
* добавление, удаление и перечисление пакетов, являющихся CAB-файлами или MSU-файлами;
* применение настроек в файле ответов автоматической установки;
* изменение WIM-файлов образов Windows;
* создание и изменение образов среды предустановки Windows (Windows PE);
* изменение параметров языка, языкового стандарта, шрифтов и ввода в образе Windows;
* обновление выпуска Windows до выпуска более высокого уровня;
* проверка применимости обновления приложения;
* системные требования;
* обновление подключенного образа Windows или в некоторых случаях работающей ОС. DISM входит в состав Windows 10

Подключение к сетевому ресурсу.

* Для использования сетевого ресурса необходимо получить дос­туп к нему.
* Метод доступа – набор правил, которые определяют, как компьютер должен отправлять и принимать данные по сетевому кабелю.
* Коллизия – наложение двух и более пакетов от компьютеров, пытающихся передать пакет в один и тот же момент времени.
* Существуют четыре метода доступа:

1. Множественный доступ с контролем несущей и обнаружением кол­лизий (Carrier-Sense Multiple Access with Collision Detection, CSMA/CD) – все компьютеры в сети прослушивают ка­бель, стремясь обнаружить передаваемые данные. Ком­пьютер может начать передачу только тогда, когда убедится, что ка­бель свободен. Если возникает коллизия, то компьютеры приостанавливают передачу на случайный интервал времени, а затем вновь стараются наладить связь.

Недостатки: при длине кабеля > 2,5 км механизм об­наружения коллизий становится неэффективным – некоторые ком­пьютеры могут не услышать сигнал и начнут передачу, что приведет к коллизии и разрушению данных. Чем больше компьютеров в сети, тем интен­сивнее сетевой трафик, и число коллизий возрастает, а это приво­дит к уменьшению пропускной способности сети.

2.Множественный доступ с контролем несущей и предотвращением коллизий (Carrier-Sense Multiple Access with Collision Avoidance, CSMA/СА) – каждый компьютер перед передачей данных в сеть сигнализирует о своем намерении, поэтому остальные компьютеры «узнают» о гото­вящейся передаче и могут избежать коллизий. Однако широковеща­тельное оповещение увеличивает общий трафик и уменьшает пропус­кную способность сети. Поэтому CSMA/CA работает медленнее, чем CSMA/CD.

3. Доступ с передачей маркера – пакет особого типа, маркер (token), циркулирует от компьюте­ра к компьютеру. Чтобы послать данные в сеть, любой компьютер должен сначала «дождаться» прихода свободного маркера и «захва­тить» его. Захватив маркер, компьютер может передавать данные. Когда какой-либо компьютер наполнит маркер своей информацией и пошлет его по сетевому кабелю, другие компьютеры уже не смогут передавать данные, так как в каждый момент времени только один компьютер использует маркер. В сети не возникает ни состязания, ни коллизий, ни временных задержек.

4. Доступ по приоритету запроса (demand priority) – концентраторы управляют доступом к кабелю, последовательно опрашивая каждый узел в сети и выявляя запросы на передачу. Концентратор должен знать все адреса связи и узлы и про­верять их работоспособность.

При доступе по приоритету запроса, как и при CSMA/CD, два компьютера могут конкурировать за право передать данные. Однако в этом методе реализуется принцип, по которому определенные типы данных, если возникло состязание, имеют соответствующий приори­тет. Получив одновременно два запроса, концентратор вначале от­дает предпочтение запросу с более высоким приоритетом.

1. **Настройка обновлений программ. Обновление драйверов.**

Настройка обновлений программ Windows

* Необходимость обновления стороннего ПО

В новых версиях программ производители добавляют дополнительные функции, исправляют найденные ошибки, исправляют конфликты с «железом».

Поэтому обновлять приложения необходимо. Особенно это касается антивирусов, менеджеров паролей, браузеров и других подобных приложений.

* Ручное обновление

Многие программы предусматривают автообновление или, по крайней мере, проверку наличия новой версии. Данная опция обычно находится в разделе меню «Справка», откуда можно сразу скачать новую версию. Также обновиться можно с сайта разработчиков, адрес которого обычно указан в разделе меню «О приложении».

* Обновление с помощью специальных утилит

Процесс проверки наличия актуальных версий установленных программ и загрузку их обновлений станет гораздо проще с помощью специальных приложений.

Kaspersky Software Updater

Эта бесплатная утилита от Лаборатории Касперского имеет минимальный интерфейс и простое управление. После ее установки и запуска просто нажимаем кнопку «Искать обновления» и ждем завершения поиска.

Найденные обновления будут представлены в виде списка, где можно будет непосредственно запустить апдейт.

Утилита добавляется в автозагрузку и в дальнейшем следит за обновлениями самостоятельно.

Обновление драйвера

Одной из наиболее распространенных причин непо­ладок оборудования являются устаревшие или по­врежденные драйверы. Для каждой версии ОС требуются особые драйверы, поэтому некорректное обновление драйвера может повлечь за собой сбой устройства либо даже вывести его из строя.

(Напри­мер, после обновления Windows 98 до Windows XP может перестать работать драйвер принтера, кото­рый не был специально разработан для Windows XP).

Поиск драйверов.

Четыре основных источника в том порядке, в котором рекомендуется производить поиск требуемого драйвера.

1. Компакт-диск из комплекта устройства. Большинство периферийных устройств по­ставляется с диском, на котором есть все не­обходимые драйверы для различных версий ОС. Это наилучший источник драйверов для нового оборудования.

2. Сайт Windows Update. Компания Microsoft публикует последние версии драйверов раз­личных устройств, а также обновления самой ОС и ее приложений. Проанализировав ваш компьютер, программа Windows Update пред­ложит список подходящих для него драйве­ров и обновлений.

3. Веб-сайт изготовителя устройства. Боль­шинство производителей оборудования пуб­ликуют обновляемые ими драйверы для своей продукции на принадлежащих им веб-сайтах.

4. Установочный диск Windows содержит библиотеку драйверов для многих устройств входящих в список поддерживаемого оборудования «HCL»

Установка новой версии драйвера для уже имеюще­гося оборудования называется обновлением драйве­ра

Чтобы обновить драйвер, необходимо выполнить следующие действия:

1. Найдите требуемый драйвер на одном из ранее перечисленных источников.

2. Раскройте меню Пуск (Start), щелкните правой кнопкой мыши на пункте Мой компьютер (My Computer) и выберите в контекстном меню команду Управление (Manage).

3. В окне Управление компьютером выберите в списке слева пункт Диспетчер устройств.

4. Щелкните мышью на знаке «+» рядом с типом устройства, драйвер для которого нужно обно­вить.

5.Выполните двойной щелчок мышью на названии нужного устройства откроется окно его свойств. Перейдите в нем на вкладку Драйвер (Driver).

6. Щелкните мышью на кнопке Обновить (Update Driver) запустится Мастер обновление оборудования (Hardware Update Wizard).

7. Если новый драйвер имеется на компакт-диске, щелкните мышью на кнопке Далее (Next); иначе выберите опцию Установка из указанного места (Install From A List Or Specific Location (Advance­d)) и вновь щелкните мышью на кнопке Далее (Next).

8. Пометьте флажок Включить следующее место поиска (Include This Location In The Search), щелкните мышью на кнопке Обзор (Browse), найдите каталог с требуемым драйвером и щелк­ните мышью на кнопке Далее (Next).

После этого драйвер будет установлен, а устрой­ство должно начать нормально работать (возможно, для этого придется включить его вручную в окне свойств в Диспетчере устройств).

Отмена установки драйвера

Иногда после обновления драйвера устройство начи­нает давать сбои, а то и вовсе перестает работать. В этом случае необходимо отменить («откатить») уста­новку драйвера. После этого система возвращается к использованию прежнего драйвера. В Windows отменить установку драйвера достаточно просто, поскольку система сохраняет прежние драй­веры.

Чтобы отменить установку драйвера, необходимо выполнить следующие действия:

1. В окне свойств устройства (открытом при помо­щи Диспетчера устройств) перейдите на вкладку Драйвер.

2. Щелкните мышью на кнопке «Откатить» (Roll Back Driver).

1. **Решение проблем конфигурации с помощью групповых политик.**

Решение проблем конфигурации с помощью групповых политик.

* Групповые политики нужны для управления операционной системы Windows.
* Они применяются во время персонализации интерфейса, ограничения доступа к определенным ресурсам системы и многого другого.
* Используют данные функции преимущественно системные администраторы.
* Они создают однотипную рабочую среду на нескольких компьютерах, ограничивают доступ пользователям.

Запуск редактора групповой политики

1. Зажать клавиши *Win + R*, чтобы открыть *«Выполнить»*.
2. Напечатать в строке *gpedit.msc* и подтвердить действие, нажав *«ОК»*.
3. Разделяется главное окно управления на две части.
4. Слева располагается структурированные категории политик.
5. Они в свою очередь делятся еще на две различные группы – настройка компьютера и настройка пользователя.
6. В правой части отображается информация о выбранной политике из меню слева.
7. Из этого можно сделать вывод, что работа в редакторе осуществляется путем перемещения по категориям для поиска необходимой настройки.
8. Выберите, например, *«Административные шаблоны»* в *«Конфигурации пользователя»* и перейдите в папку *«Меню «Пуск» и диспетчер задач»*.
9. Теперь справа отобразятся параметры и их состояния. Нажмите на любую строку, чтобы открыть ее описание.

Настройки политики

* Каждая политика доступна для настройки.
* Открывается окно редактирования параметров по двойному щелчку на определенную строку.
* Стандартное простое окно имеет три различных состояния, которые настраиваются пользователем.
* Если точка стоит напротив *«Не задано»*, то политика не действует.
* *«Включить»* – она будет работать и активируются настройки.
* *«Отключить»* – находится в рабочем состоянии, однако параметры не применяются.
* Рекомендуем обратить внимание на строку *«Поддерживается»* в окне, она показывает, на какие версии Windows распространяется политика.

Фильтры политик

* Минусом редактора является отсутствие функции поиска.
* Существует множество различных настроек и параметров, их больше трех тысяч, все они разбросаны по отдельным папкам, а поиск приходится осуществлять вручную.
* Однако данный процесс упрощается благодаря структурированной группе из двух ветвей, в которых расположились тематические папки.

Экспорт списка политик

* Если все-таки появляется необходимость найти определенный параметр, то сделать это можно только путем экспорта списка в текстовый формат, а потом уже через, например Word, осуществлять поиск.
* В главном окне редактора есть специальная функция *«Экспорт списка»*, он переносит все политики в формат TXT и сохраняет в выбранном месте на компьютере.

Применение фильтрации

* Выберите, например, *«Конфигурация компьютера»*, откройте раздел *«Административные шаблоны»* и перейдите в *«Все параметры»*.
* Разверните всплывающее меню *«Действие»* и перейдите в *«Параметры фильтра»*.
* Поставьте галочку возле пункта *«Включить фильтры по ключевым словам»*. Здесь имеется несколько вариантов подбора соответствий.
* Откройте всплывающее меню напротив строки ввода текста и выберите
  + *«Любой»* – если нужно отображать все политики, которые соответствуют хотя бы одному указанному слову
  + *«Все»* – отобразит политики, содержащие текст из строки в любом порядке,
  + *«Точный»* – только параметры, точно соответствующие заданному фильтру по словам, в правильном порядке.
* Флажками снизу строки соответствий отмечаются места, где будет осуществляться выборка.
* Нажмите *«ОК»* и после этого в строке *«Состояние»* отобразятся только подходящие параметры.
* В том же всплывающем меню *«Действие»* ставится или убирается галочка напротив строки *«Фильтр»*, если нужно применить или отменить заранее заданные настройки подбора соответствий.

Изменение окна безопасности Windows

* Если зажать сочетание клавиш *Ctrl + Alt + Delete*, то будет запущено окно безопасности, где осуществляется переход к диспетчеру задач, блокировка ПК, завершение сеанса системы, смена профиля пользователя и пароля.
* Каждая команда за исключением *«Сменить пользователя»* доступна для редактирования путем изменения нескольких параметров. Выполняется это в среде с параметрами или путем изменения реестра.
* Откройте редактор.
* Перейдите в папку *«Конфигурация пользователя»*, *«Административные шаблоны»*, *«Система»* и *«Варианты действий после нажатия Ctrl + Alt + Delete»*
* Откройте любую необходимую политику в окне справа
* В простом окне управления состоянием параметра поставьте галочку напротив *«Включить»* и не забудьте применить изменения.

1. **Тестирование на совместимость в безопасном режиме. Восстановление системы.**

Тестирование на совместимость в безопасном режиме**.**

* Безопасный режим Windows 10 — Safe Mode представляет собой диагностический режим загрузки ОС, предназначенный для поиска и решения многих проблем в ОС с целью вернуть ее к работоспособному состоянию.
* Используется этот режим в случаях, когда компьютер отказывается запускаться в штатном режиме или невозможно выполнить какое-либо действие ввиду использования
* Во время загрузки персонального компьютера в Safe Mode в оперативную память *помещаются только необходимые для запуска и работы ОС компоненты*, такие как драйверы, системные службы, проводник, ядро ОС.
* Прикладное ПО и драйверы периферийных устройств, ненужных для работы компьютера, не загружаются.

Вариант загрузки компьютера в Safe Mode — использование утилиты для конфигурации системы при выполнении следующих действий.

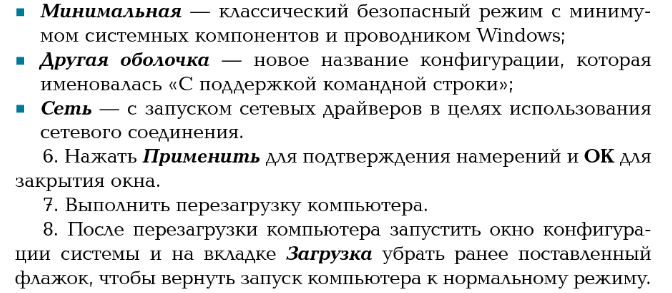
1. Запустить командный интерпретатор, представленный диалоговым окном с названием «Выполнить», которое запускается посредством комбинации клавиш [Win + R].

2. Ввести системную команду msconfig, запускающую утилиту для конфигурирования запуска Windows 10

3. В появившемся окне активировать вторую вкладку Загрузка и выбрать ОС, которая должна запускаться в диагностическом режиме.

4. Отметить флажком расположенную ниже формы со списком ОС опцию Безопасный режим.

5. Выбрать одну из предлагаемых конфигураций загрузки компьютера



Другой вариант запуска Windows 10 в безопасном режиме

1. Открыть окно Параметры (использовать поисковую строку, комбинацию клавиш [Win + R] или кнопку в Пуске).

2. Щелкнуть по названию раздела Обновление и безопасность, перейти в подраздел Восстановление

3. Найти пункт Особые варианты… и нажать кнопку Перезагрузить сейчас.

4. После самотестирования оборудования появится экран дополнительных вариантов запуска компьютера. Выбрать вариант Диагностика.

5. Выбрать необходимые параметры и выполнить перезагрузку

6. В меню параметров запуска выбрать нужный режим запуска компьютера с помощью клавиш [F4]…[F6]

Применение загрузочного устройства для вызова безопасного режима**.**

Для включения безопасного режима Windows 10, если система не запускается, понадобится загрузочный носитель с файлами установки ОС. Может использоваться диск для восстановления ОС, но он имеется в наличии у ограниченного числа пользователей. Необходимо выполнить определенную последовательность действий.

1. Запустить систему с загрузочного флеш-устройства, используя Boot Menu BIOS

2. Удерживать клавиши [Shift + F10] для запуска командной строки или нажать по вкладке Восстановление системы в окне с *кнопкой Установить* вызвать Диагностика, зайти в дополнительные параметры и вызвать командную строку.

3. С помощью командной строки выполнить команду bcdedit /set {default} safeboot minimal для запуска классического режима отладки и, заменив minimal на network, загрузиться в диагностическом режиме с запуском сетевых драйверов.

Выполнить фрагмент кода

bcdedit /set {default} safebootalternateshell yes

4. Закрыть окно командной строки, выполнить перезагрузку системы.

5. После решения проблем перезагрузить ОС, запустить командную строку, как ранее, и ввести для выполнения bcdedit/deletevalue {default} safeboot в целях выключения безопасного режима.

Восстановление системы

-Восстановление системы (англ. System restore) — компонент ОС Windows (процесс rstrui.exe), предназначенный для восстановления работоспособности ОС путем отката *(восстановления предыдущего состояния в персональном компьютере системных файлов, ключей реестра, установленных программ и т. д*.). Восстановление системы присутствует в системах Windows XP, Windows Vista, Windows 7, Windows 8 и Windows 10.

-Пользователь может создавать новую точку восстановления вручную, производить откат к уже существующей точке или изменять конфигурацию восстановления системы.

-Вплоть до Windows XP система может быть восстановлена, пока Windows загружается в нормальном, безопасном режиме или безопасном режиме с поддержкой командной строки.

-Если Windows не запускается, восстановление невозможно.

-Диск восстановления можно создать как своими силами, так и с помощью специального ПО от производителя компьютера или от стороннего разработчика.

-Многим пользователям персональных компьютеров с ОС Windows 10 известно, что в системе реализованы две панели управления компьютером.

Первая — это всем знакомая «Панель управления» и вторая — новая «Параметры»

На примере воспользуемся новой панелью «Параметры». Попасть в новую панель можно через меню Пуск, кликнув на пункт Параметры

При попадании в панель Параметры дальнейшим шагом будет переход во вкладки Обновление и безопасность / Восстановление

* В открывшемся окне появится кнопка Начать. По ее нажатию откроется окно с выбором действий
* В меню выбора пользователю предлагаются два варианта: сохранения или удаления личных данных.

-Первый вариант полностью сохраняет все личные файлы пользователя и полностью чистит систему.

-Второй вариант удаляет все данные на компьютере, включая файлы пользователя, настройки ОС и все приложения.

Данный способ удобен, если необходимо переустановить Windows 10 без инсталляционного DVD-диска, или когда нет загрузочного флеш-носителя.

По окончании сброса на персональном компьютере будет чистая Windows 10, включая только личные данные.

1. **Производительность ПК. Проблемы производительности. Анализ журналов событий.**

Производительность ПК

* Производительность компьютера — это скорость обрабатываемой информации за определенное время. И чем больше скорость, тем лучше показатель производительности.
* Детали, отвечающие за производительность. Самая главная деталь — это процессор компьютера, затем по значимости идет оперативная память и нужно знать, что не объем, а скорость, с которой данные считываются с жесткого диска и передаются процессору на обработку, является более важной характеристикой.

Причины падения производительности

* Главным фактором падения производительности является засорение компьютера различными данными. Это в основном игры и программы, а музыка, видео, картинки не столь важны.
* Также на производительности значительно сказываются антивирусы и подобные сканирующие программы.

Оценка производительности компьютера

Существуют специализированные испытательные программы (benchmark), которые измеряют производительность компьютера при выполнении на ней определенного набора задач.

В зависимости от ситуации они могут быть использованы в разных целях.

1. Сравнение систем. Испытательные программы часто используются *для сравнения компьютеров конкурирующих фирм* или выявления преимуществ новых моделей перед старыми.

2. Оценка степени улучшения системы после ее модернизации. Испытательные программы часто используют для оценки степени улучшения характеристик системы после ее модернизации.

*Запустив такую программу на компьютере до и после его модернизации, можно оценить, насколько изменилась его производительность после замены процессора, увеличения объема или замены модулей оперативной памяти, установки нового жесткого диска или системной платы.*

3. Диагностика неисправностей. Иногда испытательные программы оказываются полезными для диагностики системы. Проверяя и перенастраивая ключевые компоненты компьютера и оценивая результаты произведенных манипуляций с помощью испытательных программ, *специалист сможет быстрее определить и устранить причину замедления работы системы, чем просто наблюдая за ее работой*.

4. Проверка достоверности результатов работы испытательных программ. Одной из самых серьезных проблем, связанных с испытательными программами, является *достоверность получаемых с их помощью результатов.*

Выполнение теста производительности в Windows 10

Для выполнения теста из PowerShell необходимо исполнить следующие действия.

1. Открыть оболочку PowerShell от имени администратора. Для этого, используя поиск Windows, нужно набрать powershell и, кликнув правой клавишей мыши, запустить приложение от имени администратора.

2. В открывшемся окне ввести winsat formal и нажать [Enter].

3. Начнется тест производительности, который в зависимости от конфигурации компьютера может занять от одной до нескольких минут/

По завершении теста его итоги будут записаны в файл формата XML, который доступен по адресу  
C:\Windows\Performance\WinSAT\DataStore\…Formal.Assessment (Recent).WinSAT.xml.

Оценка производительности аппаратной конфигурации компьютера — это утилита Winaero WEI tool.

Приложение обладает полной совместимостью с Windows 10, не нуждается инсталляции и не вмещает никакого вредоносного кода.

Термин «узкое место» (bottleneck)

Термин «узкое место» (bottleneck) используется при рассмотрении вопросов производительности ПО и оборудования, относится к ограничивающему производительность состоянию, вызванному каким-либо компонентом или набором компонентов.

Выявление проблемы производительности.

* Чтобы определить наличие проблем в системе, нужно выполнить некоторые общие исследования по производительности системы.
* Многие из узких мест тесно связаны друг с другом, и одно узкое место может «маскироваться» под другое.
* Необходимо искать как аппаратные, так и программные узкие места, поскольку причиной многих проблем производительности являются комбинации узких мест.
* Причиной аппаратных узких мест могут становиться такие компоненты оборудования, как ЦП, память и подсистема ввода-вывода;
* причиной программных узких мест могут становиться приложения SQL Server и операторы SQL.

Центральный процессор

* Одной из наиболее распространенных проблем производительности является недостаток мощности.
* Процессорная мощность системы определяется количеством, типом и скоростью ЦП в этой системе. Если системе недостает мощности ЦП, то она не может достаточно быстро обрабатывать транзакции, как это требуется пользователям.

Память

* Количество памяти, доступной SQL Server, является одним из наиболее критичных факторов для производительности SQL Server. Важным фактором является также соотношение между памятью и мощностью подсистемы ввода-вывода.
* Чем больше данных, считываемых из памяти, тем выше производительность системы.
* Доступ к памяти осуществляется намного быстрее, чем доступ к диску.

Приложения

* Еще одним компонентом системы, который обычно вызывает проблемы производительности, являются приложения SQL Server.
* Причиной этих проблем может быть код самого приложения или операторы SQL, которые запускаются из этого приложения

Анализ журналов событий

* Журнал событий (event log) — в Microsoft Windows стандартный способ для приложений и ОС записи и централизованного хранения информации о важных программных и аппаратных событиях.
* Служба журналов событий сохраняет события от различных источников в едином журнале событий, программа просмотра событий позволяет пользователю наблюдать за журналом событий, программный интерфейс (API) дает возможность приложениям записывать в журнал информацию и просматривать существующие записи.
* Записи журнала событий хранятся в ключе реестра:

Данный ключ содержит подключи, называемые файлами журнала.

По умолчанию имеются:

* файл журнала приложений — для событий приложений и служб;
* файл журнала безопасности — для событий системы аудита;
* файл системного журнала — для событий драйверов устройств

Просмотр журнала событий на компьютере с Windows 10

1. Открыть меню Пуск. В открывшемся окне, в списке всех приложений, внизу списка, открыть вкладку Служебные Windows и в открывшемся списке нажать на вкладку Панель управления.

2. На странице Панель управления нажать на вкладку Система и безопасность

3. На странице Система и безопасность нажать на вкладку Просмотр журналов событий.

4. Откроется окно «Просмотр событий».

Здесь можно просматривать:

* сводку административных событий;
* недавно просмотренные узлы;
* сводку журнала.

В верхней левой части окна можно открыть и просмотреть:

* настраиваемые представления;
* журналы Windows;
* журналы приложений и служб;
* подписки.

1. **Настройка управления питанием.**

Настройка управления питанием

* Схема управления питанием представляет собой набор аппаратных и системных параметров, которые управляют тем, как компьютер использует энергию.
* Доступные схемы управления питанием зависят от типа персонального компьютера и настроек, которые производитель внес в планы электропитания, доступные в Windows по умолчанию.
* Некоторые планы питания предназначены для обеспечения высокой производительности с ущербом для автономности, в то время как другие настроены на обеспечение максимально возможного времени автономной работы, при этом производительность устройства ставится под угрозу.

Можно настроить параметры отключения компьютера и параметры перехода в спящий режим. Чтобы настроить питание компьютера в Windows 10, необходимо выполнить следующие действия.

1. Открыть меню Пуск. В открывшемся окне нажать на значок или вкладку Параметры

2. На странице Параметры нажать на вкладку Система

3. На странице Система, в левой боковой панели, перейти на вкладку Питание и спящий режим. Здесь можно настроить параметры отключения экрана и параметры перехода в спящий режим

* выбрать время отключения экрана при питании от сети;
* выбрать время перехода в спящий режим при питании от сети.
* В нижней части окна можно открыть дополнительные параметры питания. В дополнительных параметрах можно выбрать схему управления питанием.
* Рекомендуется установить сбалансированную схему питания.
* В следующем окне можно выбрать те же параметры, что и в предыдущем, а, чтобы открыть более детальные настройки, потребуется нажать на надпись ***«Изменить дополнительные параметры...»***
* Появится окно, где пользователю станет доступно множество опций.
* Здесь можно установить

параметры отключения жесткого диска, различные режимы сна, батареи, монитора и многое другое.

* Тут же, в случае нестабильной работы вашего ПК, можно восстановить настройки по умолчанию, нажав на соответствующую кнопку.

Тонкая настройка электропитания ПК

* Выбрать нужную схему, которую вы будете редактировать.
* Далее в центре вы увидите список параметров, которые можно изменить.
* Каждый из заголовков можно открыть, нажав на кнопку***«+»*** и выбрать требуемый параметр.

Жёсткий диск

Тут устанавливается время неактивности, после которого будет отключен диск. Это позволит не использовать его во время простоя, а как только вы возобновите работу, он автоматически включиться.

InternetExplorer

Здесь можно установить частоту обновления JavaScript. Эта настройка определяет, с какой частотой будут выполняться скрипты Java на интернет-ресурсах. Чем выше частота работы, тем плавнее воспроизводится. Повышенная частота будет интенсивнее использовать процессор и, соответственно, потреблять больше энергии.

Параметры фона рабочего стола

В этом разделе можно отключить смену обоев на рабочем столе.

Слайд-шоу из меняющихся изображений потребляет больше энергии, особенно если выставлен низкий интервал изменения фото.

Параметры адаптера беспроводной сети —

это ваш Wi-Fi приемник, можно установить различные параметры его работы — максимальное энергосбережение или производительность

Сон

Данная настройка отвечает за параметры перехода ПК в разнообразные режимы сниженного потребления энергии. Тут можно выставить время, когда компьютер автоматически заснет, а также разрешить гибернацию или гибридный вариант.

* Гибридный спящий режим — это настройка с пониженным потреблением энергии, комбинирующая в себе сон и гибернацию.

Разрешить таймеры пробуждения

В данном разделе можно разрешить тем или иным событиям выводить ПК из спящего режима.

*(Например, задание в планировщике системы должно сработать, когда ваш ПК спит, установив параметр* ***«Включить»****, компьютер проснется для выполнения запланированной задачи.)*

Параметры USB

Тут можно указать компьютеру отключать питание USB устройств при бездействии системы.

Кнопки питания и крышка

Данный раздел служит для установки действий, которые будут происходить в случае использования кнопок питания ПК или закрытии крышки ноутбука.  
Возможны следующие варианты: включать режим сна, гибернацию, отключить или не совершать никаких действий.

Действие кнопки спящего режима

В этом разделе предоставляется возможность изменить функцию кнопки системы «Спящий режим». При желании вместо сна можно установить гибернацию или отключение монитора.

Экран

Здесь настраиваются параметры регулировки яркости монитора и время его отключения. *Например, при хорошем освещении яркость экрана будет прибавляться и наоборот, в более затемненном помещении — убавится.*

1. **Оптимизация использования процессора.**

Оптимизация работы процессора

Все манипуляции по улучшению качества работы ЦП можно поделить на две группы:

* Оптимизация. Основной акцент делается на грамотное распределение уже доступных ресурсов ядер и системы, дабы добиться максимальной производительности. В ходе оптимизации трудно нанести серьёзный вред ЦП, но и прирост производительности, как правило, не очень высокий.
* Разгон. Манипуляции непосредственно с самим процессором через специальное ПО или BIOS для повышения его тактовой частоты. Прирост производительности в этом случае получается весьма ощутимым, но и возрастает риск повредить процессор и другие компоненты компьютера в ходе неудачного разгона.

Оптимизация работы ОС

* Это самый безопасный метод увеличения производительности ЦП путём очистки автозагрузки от ненужных приложений и дефрагментации дисков.
* Автозагрузка – это автоматическое включение той или иной программы/процесса при загрузке операционной системы. Когда в этом разделе скапливается слишком много процессов и программ, то при включении ОС и дальнейшей работе в ней, на центральный процессор может быть оказана слишком высокая нагрузка, что нарушит производительность.

Очистка Автозагрузки

* В автозагрузку приложения можно добавлять как самостоятельно, так и приложения/процессы могут добавляться сами. Чтобы второго случая не было, рекомендуется внимательно читать все пункты, которые отмечены галочкой во время установки того или иного софта.

Как убрать уже имеющиеся элементы из Автозагрузки:

* Для начала перейдите в **“Диспетчер задач”**. Чтобы перейти туда, используйте комбинацию клавиш **Ctrl+SHIFT+ESC** или в поиске по системе вбейте **“Диспетчер задач”** (последнее актуально для пользователей на Windows 10).
* Перейдите в окно **“Автозагрузка”**. Там будут представлены все приложения/процессы, которые запускаются вместе с системой, их состояние (включено/отключено) и общее влияние на производительность (Нет, низкое, среднее, высокое).
* В первую очередь, рекомендуется отключать все пункты, где в колонке **“Степень влияния на производительность”** стоят отметки

**“Высокое”**.

Чтобы отключить процесс, кликните по нему и в правой нижней части окна выберите **“Отключить”**

Чтобы изменения вошли в силу рекомендуется выполнить перезагрузку компьютера.

Проведение дефрагментации

* Дефрагментация диска увеличивает не только скорость работы программ на этом диске, но также немного оптимизирует работу процессора.
* Происходит это потому, что ЦП обрабатывает меньше данных, т.к. в ходе дефрагментации обновляется и оптимизируется логическая структура томов, ускоряется обработка файлов. Инструкция проведения дефрагментации:

Нажмите правой кнопкой мыши по системному диску (вероятнее всего, это **(C:)**) и перейдите в пункт **“Свойства”**.

В верхней части окна найдите и перейдите во вкладку **“Сервис”**. В разделе

**“Оптимизация и дефрагментация диска”** нажмите **“Оптимизировать”**.

* В открывшемся окне можно выбрать сразу несколько дисков. Перед дефрагментацией рекомендуется провести анализ дисков, нажав на соответствующую кнопку. Анализ может идти до нескольких часов, в это время не рекомендуется запускать программы, которые могут вносить какие-либо изменения на диске.
* После анализа система напишет, требуется ли дефрагментация. Если да, то выделите нужный диск (диски) и нажмите на кнопку **“Оптимизировать”**.

Рекомендуется также назначить автоматическую дефрагментацию дисков. Для этого перейдите по кнопке **“Изменить параметры”**, далее отметьте галочкой **“Выполнять по расписанию”** и задайте нужное расписание в поле **“Частота”**.

Узнаём, пригоден ли процессор для разгона

Перед разгоном обязательно просмотрите характеристики своего процессора при помощи специальной программы (например AIDA64).

 AIDA64 -условно-бесплатная, с её помощью можно узнать подробную информацию обо всех компонентах компьютера, а в платной версии даже проводить с ними некоторые манипуляции

Инструкция по использованию:

* Чтобы узнать температуру ядер процессора (это один из главных факторов при разгоне), в левой части выберите пункт **“Компьютер”**, затем перейдите в пункт **“Датчики”** из главного окна или меню пунктов.
* Здесь вы сможете просмотреть температуру каждого ядра процессора и общую температуру. На ноутбуке, при работе без особых нагрузок она не должна превышать 60 градусов, если она равна или даже немного превышает этот показатель, то от разгона лучше отказаться. На стационарных ПК оптимальная температура может колебаться в районе 65-70 градусов.
* Если всё нормально, то перейдите в пункт **“Разгон”**. В поле **“Частота ЦП”** будет указано оптимальное число МГц при разгоне, а также процент, на который рекомендуется увеличить мощность (обычно колеблется в районе 15-25%).

Способ 1: оптимизация при помощи CPU Control

Инструкция по использованию данной программы:

* После установки откроется главная страница. Изначально всё может быть на английском. Чтобы это исправить, перейдите в настройки (кнопка **“Options”** в правой нижней части окошка) и там в разделе **“Language”** отметьте русский язык.

На главной странице программы, в правой части, выберите режим **“Ручной”**

* В окне с процессорами выберите один или несколько процессов. Чтобы сделать выбор нескольких процессов, зажмите клавишу **Ctrl** и щёлкайте мышкой по нужным элементам.
* Затем нажмите правую кнопку мыши и в выпавшем меню выберите ядро, которое вы бы хотели назначить для поддержания той или иной задачи.
* Ядра носят названия по следующему типу CPU 1, CPU 2 и т.д.

Если вы не хотите назначать процессы вручную, то можно оставить режим **“Авто”**, который стоит по умолчанию.

После закрытия программа автоматически сохранит настройки, которые будут применятся при каждом запуске ОС.

Способ 2: разгон при помощи ClockGen

ClockGen — это бесплатная программа, подходящая для ускорения работы процессоров любой марки и серии (за исключением некоторых процессоров Intel, где разгон невозможен сам по себе).

Перед разгоном убедитесь, что все температурные показатели ЦП в норме.

Как пользоваться ClockGen:

* В главном окне перейдите во вкладку **«PLL Control»**, где при помощи ползунков можно изменить частоту процессора и работы оперативной памяти.
* Не рекомендуется за раз слишком сильно передвигать ползунки, лучше небольшими шагами, т.к. слишком резкие изменения могут сильно нарушить работу ЦП и ОЗУ.
* Когда получите необходимый результат, нажмите на **«Apply Selection»**.

Чтобы при перезапуске системы настройки не сбивались, в главном окне программы перейдите в пункт **«Options»**.

Там, в разделе **«Profiles Management»**, поставьте флажок напротив **«Apply current settings at startup»**.

1. **Оптимизация использования памяти. Оптимизация использования жесткого диска. Оптимизация использования сети.**

Оптимизация использования памяти

Способ 1: Очистка кэша ОЗУ

* Как известно, в оперативную память загружаются данные приложений, что позволяет ускорить их запуск и выполнение каких-либо операций. Информация, которая считается устаревшей, выгружается или перезаписывается автоматически, однако это происходит не всегда, что напрямую влияет на быстродействие и загруженность RAM.
* Время от времени необходимо очищать кэш самостоятельно и проверять, как это скажется на работе Windows 10.
* Самый простой вариант решения заключается в банальной перезагрузке компьютера, поскольку во время неё модули RAM кратковременно обесточиваются, что и удаляет кэшированное содержимое.

Очистку кэша ОЗУ можно произвести и системными средствами, с помощью одной из системных утилит.

-Вызовите окно **«Выполнить»** комбинацией клавиш **Win+R**. Когда оно появится, введите запрос: C:\\windows\system32\rundll32.exe

-Для 64-разрядных версий «десятки» запрос выглядит так: C:\\windows\SysWOW64\rundll32.exe

-Проверьте правильность ввода и нажмите **«ОК»**.

-Запущенное средство работает в фоновом режиме без каких-либо уведомлений, поэтому нужно подождать около 10 секунд, после чего кэш должен быть очищен.

Способ 2: Обновление драйверов

* Следующая стандартная рекомендация заключается в ручной проверке обновлений драйверов для всех комплектующих, установленных в ПК. Это требуется для того, чтобы исключить вероятность появления конфликтов из-за недостающих файлов или несовместимости. Вы можете сами с помощью стандартных или сторонних средств запустить эту проверку и установить все найденные драйверы

Способ 3: Установка системных обновлений

* Исправления и нововведения от Майкрософт тоже оказывают прямое влияние на быстродействие и загрузку оперативной памяти разными службами и процессами. Лучше всегда поддерживать ПК в актуальном состоянии, чтобы избегать различных сбоев и конфликтов.
* Проверить системные обновления можно всего в несколько кликов.
* Откройте **«Пуск»** и перейдите в **«Параметры»**.
* Отыщите **«Обновление и безопасность»**.
* В первом же разделе **«Центр обновления Windows»**

запустите проверку апдейтов и инсталлируйте их, если такие будут найдены.

Способ 4: Проверка системы на наличие вирусов

* Заражение вирусами — одна из самых частых проблем, влияющих на понижение производительности операционной системы. Многие вредоносные файлы функционируют в фоновом режиме под видом различных процессов, потребляя ресурсы оперативной памяти и других комплектующих. От пользователя требуется лишь предотвратить действие подобных угроз, регулярно проверяя компьютер на их наличие.

Способ 5: Отключение программ автозагрузки

Программы, которые запускаются сразу же при входе в Windows, используют оперативную память и другие системные ресурсы даже в фоновом режиме, поэтому рекомендуется следить за тем, какие из инструментов добавлены в автозагрузку.

Способ 6: Отключение открытия приложений после перезапуска

По умолчанию в Windows 10 активирована функция, автоматически запускающая незакрытые программы при перезагрузке или обновлении системы. Далеко не всем эта опция требуется, поэтому ее можно отключить, чтобы разгрузить оперативную память, ведь теперь кэш сохраняться не будет.

Способ 7: Отключение фоновых приложений

* В некоторых случаях стандартные приложения Windows или те, которые были скачаны пользователем вручную из Microsoft Store, могут функционировать в фоновом режиме, что тоже оказывает влияние на оперативную память. Такие программы не получится отключить через **«Автозагрузку»**, поэтому придется произвести немного другие действия.

Оптимизация использования жесткого диска

Поддержание порядка на диске

* Для того, чтобы на диске всегда был порядок, достаточно свободного места и не снижалась его производительность, нужно выполнить несколько простых настроек и придерживаться определенных правил хранения файлов.

Отключение лишних обновлений

* Операционная система Windows по умолчанию настроена на получение важных и рекомендуемых обновлений. К важным относятся обновления безопасности и исправление ошибок.
* А вот с рекомендуемыми приходят гигабайты всякого хлама. Это дополнительные системные компоненты и службы, которые почти никому не нужны. Они занимают много места на диске, оперативную память и использует ресурсы процессора. В результате компьютер начинает тормозить.
* Кроме этого, система в обязательном порядке создает резервные копии системных файлов, которые нельзя удалить. В результате папка «Windows» на диске «C» серьезно разрастается в размерах и на нем начинает не хватать места.

Систематизация файлов

* Чтобы на диске всегда был порядок, вы знали где и какие файлы у вас находятся и сколько они занимают места, приучите себя к их правильному размещению.
* Не храните файлы и папки на рабочем столе, он предназначен в основном для ярлыков.
* Сохраняйте все файлы сразу же в нужные папки с интуитивными именами.
* Группируйте файлы одной и той же тематики в одну папку.
* При скачивании новой версии файла или программы удаляйте старые версии, чтобы они не дублировались и не занимали двойной объем.

Ускорение работы диска

Обновление драйвера контроллера дисков

От драйвера контроллера дисков зависит не только скорость его работы, но и стабильность всей системы.

Обновить драйвер контроллера дисков до последней версии. Это особенно важно если у вас SSD диск.

Включение функции TRIM на SSD

* Если нет SSD диска и ваш компьютер работает не так быстро, как хотелось, то установите его и вы получите значительный прирост быстродействия и отзывчивости системы!
* Все современные диски SSD поддерживают функцию TRIM, которая предназначена для оптимизации их скорости.
* Эта функция обязательно должна быть включена в операционной системе.
* Для проверки состояния TRIM для вашего SSD накопителя можно использовать командную строку, запущенную от имени администратора.
* Введите команду *fsutil behavior query disabledeletenotify* и нажмите Enter.
* Если функция TRIM включена, то вы должны увидеть строку «DisableDeleteNotify=0».

Отключение лишних служб

* Во всех версиях Windows есть множество не совсем нужных системных служб. Некоторые из них замедляют работу диска и всей системы, а также приводят к повышенному износу HDD и SSD дисков.
* Нажмите сочетание клавиш «Win+R», введите «services.msc» и нажмите «Enter». Кликните правой кнопкой мыши на требуемой службе и выберите «Свойства».

Отключение автозагрузки программ

Лишние программы в автозагрузке не только замедляют загрузку компьютера, но и используют его память и процессор.

Поэтому их отключение положительно влияет не только на скорость работы диска, но и на производительность системы в целом.

Дефрагментация диска

В процессе работы диска происходит его фрагментация, т.е. файлы разбиваются на множество мелких фрагментов. При обращении к тому или иному файлу, диску приходится собирать все его фрагменты в единое целое, что значительно снижает производительность самого диска и всей системы.

* Для ускорения работы диска, после его тщательной очистки, необходимо выполнить его дефрагментацию. В этом процессе все файлы, разбитые на фрагменты, соединяются. Это значительно облегчает и ускоряет работу диска.

Производить дефрагментацию дисков SSD не требуется, так как они работают по другому принципу и не подвержены проблемам фрагментации

Оптимизация использования сети.

Скорость интернета на устройствах с Windows 10 может быть низкой из-за:

* механических повреждений кабеля при проводном подключении;
* устаревших драйверов Wi-Fi адаптера;
* включённых ограничений скорости;
* неправильных настроек файлообменника или браузера и т. д.

Оптимизация скорости файлообмена на устройстве

* Если владелец ПК активно пользуется файлообменниками типа µTorrent, постоянно скачивает и раздаёт большие объёмы мультимедиа, скорость интернета будет снижена.
* Настройте программу:

Кликом по иконке файлообменника откройте контекстное меню.

Из перечня значений для пунктов «Ограничение приёма» и «Ограничение отдачи» выберите более низкую скорость. Из перечня значений для пунктов «Ограничение приёма» и «Ограничение отдачи» нужно выбрать 0 или «Не ограничено»

Отключение ограничения скорости QoS

QoS — планировщик пакетов, обеспечивающий нормальную работу интернет-соединения, когда сеть перегружена. Чтобы убедиться в том, что он используется на ПК:

* Щёлкните правой кнопкой по меню «Пуск» и выберите «Параметры».
* Зайдите в раздел «Сеть и интернет». Зайдите в раздел «Сеть и интернет»
* Кликните по вкладке Ethernet и перейдите по ссылке «Настройка параметров адаптера».
* Выберите задействованный адаптер и откройте его свойства правой кнопкой мыши.
* Напротив пункта «Планировщик пакетов QoS» должен стоять флажок.
* Ускорить интернет на «десятке» можно путём изменения ограничений скорости доступа в интернет. По умолчанию система снижает эту скорость на 20%, резервируя тем самым место для канала связи. Чтобы изменить дефолтные настройки:
  + Вызовите диалоговое окно сочетанием Win + R.
  + Введите команду gpedit.msc.
  + Найдите папку «Конфигурация компьютера», в которой поочерёдно откройте «Административные шаблоны», «Сеть» и «Планировщик пакетов QoS».
  + Выберите параметр «Ограничить резервируемую пропускную способность» и откройте его. Выберите параметр «Ограничить резервируемую пропускную способность» и откройте его
  + В следующем окне поставьте флажок на пункте «Включено» и установите 0 в качестве значения ограничения.
  + Нажмите «ОК», чтобы настройки сохранились.

Изменение лимита пропускной способности сети

Причиной низкой скорости сетевого соединения может быть активное использование исходящего канала каким-либо приложением, работающим в фоновом режиме

Программы и утилиты для увеличения скорости интернета

Скорость интернет-соединения можно увеличить с помощью специальных программ:

* Advanced System Care;
* Ashampoo Internet Accelerator;
* NetScream;
* СFosSpeed.

1. **Инструменты повышения производительности программного обеспечения.**

Инструменты повышения производительности программного обеспечения.

На скорость выполнения вашей программы влияет 2 непосредственных фактора:

* Скорость исполнения кода
* Объём выделяемой памяти

Минимизировать одновременно влияние обоих этих факторов невозможно: заигрывания с памятью неизбежно приводят к раздуванию кода, а «малобуквенный» текст к повышенному потреблению ресурсов исполняемой машиной.

Поэтому главным правилом оптимизации является исключение излишеств в вашей программе.

Это означает никакого лишнего кода и строго ограниченное использование памяти.

Операции по оптимизации

Если задача оптимизации встала перед вами уже после написания кода, то наиболее разумным решением будет предварительная очистка и разбиение на части с последующим изучением времени выполнения отдельных блоков.

Очистка предполагает удаление неиспользуемых участков, переменных, избыточных заходов циклов.

Анализ скорости выполнения после разбиения чистого кода даст вам объективную оценку того, в какую сторону направить свои усилия по оптимизации.

Логично, что если одна операция выполняется 40% времени, а другая – 2%, то и разработчик должен куда больше усилий приложить к уменьшению производственных потерь в первом случае.

Так как мы предполагаем, что код уже чистый, то необходимо в первую очередь рассмотреть выполнение следующих правил:

* Минимизировано количество используемых переменных.

Например, ваша программа изобилует различными циклами. По правилам красивого кода, вы можете создать для каждого цикла свой уникальный счётчик, но с точки зрения оптимизации – это трата драгоценных ресурсов. Если в одной части программы переменная “отработала”, то её вполне можно применить в другой.

* Правильно выбраны типы данных.

Как известно, каждый тип данных имеет свой используемый диапазон, то есть собственно тот размер памяти, который резервируется под его использование. Например, создавая код в Java и имея переменную, способную принимать только два значения (например, «on» и «off»), лучше использовать boolean с созданием последующего соответствия, но никак не char.

* Минимизировано количество присваиваний.

Опять-таки, руководствуясь принципами красоты кода, вы можете разбить длинное арифметическое выражение на несколько более мелких. Это чревато появлением избыточных переменных и лишними операциями присваивания, что позитивно на быстродействии точно не скажется.

* Переменные инициируются при объявлении.

Это правило хорошего тона в программировании не только повысит производительность, исключая лишние операции, но и избавит от невынужденных ошибок.

* Однотипные повторяющиеся операции объединены в процедуру или функцию.

Еще один приём из основ программирования, который, тем не менее, часто игнорируется новичками.

* Однотипные циклы объединены.

Допустим, у вас есть несколько массивов одинаковой размерности, которые надо заполнить в цикле. Вы можете создать несколько циклов и повысить читабельность или запихнуть все операции в общий цикл и повысить быстродействие. Решать вам.

* Использованная память немедленно очищается.

Безусловно, не стоит удалять каждую переменную сразу после окончания её использования, но когда речь идет о работе с существенным объемом памяти (например, с большими массивами), контролировать потребление ресурсов просто необходимо.

1. **Средства диагностики оборудования. Разрешение проблем аппаратного сбоя.**

Выявление и устранение конфликтов оборудования

Конфликты почти всегда оказываются результатом неудачно проведенной модернизации компьютера.

Конфликты отличаются тремя характерными признаками:

* в систему было установлено новое устройство или программа;
* проблема возникла после установки нового устройства или программы;
* до установки нового устройства или программы система работала нормально.

Если присутствуют все эти три признака, то с большой долей вероятности можно утверждать, что возник аппаратный или программный конфликт (а не неисправность какого-либо устройства).

В отличие от других неполадок, проявления которых связаны с выходом из строя подсистем, признаки конфликтов гораздо более многообразны и неоднозначны. Из них можно выделить следующие:

* компьютер зависает в процессе выполнения POST или инициализации ОС;
* система зависает при выполнении прикладной программы;
* система зависает при использовании конкретного устройства (например, сканер);
* компьютер зависает произвольным образом, без предупреждения и независимо от исполняемой программы;
* компьютер может и не зависать, но новое устройство не работает. При этом ранее установленные в систему устройства могут продолжать работать нормально;
* компьютер может не зависать, но устройства или программы, ранее работавшие нормально, теперь не функционируют. Вновь установленное устройство (и соответствующее ПО) может работать, а может и не работать.
* Общим для всех этих проблем является то, что серьезность и частота проявления неисправностей, зависят от нескольких факторов.
* К ним относятся устройства, вовлеченные в конфликт, ресурсы из-за которых он возникает и операции, при которых конфликт проявляется.
* Поскольку аппаратные и программные конфигурации компьютеров чрезвычайно разнообразны, более точно выявить симптомы конфликтов практически невозможно.

Распознание и разрешение конфликтов.

* Существуют некоторые общие подходы и методы, позволяющие упростить ее решение.
* Основное правило разрешения конфликта можно кратко сформулировать следующим образом: то, что было подключено последним, отключается первым.
* Самым быстрым способом устранения конфликта является удаление из системы устройства или программы, которая его вызвала.
* Программные конфликты. В компьютере конфликтовать могут программы двух типов: резидентные программы и драйверы устройств.

Резидентные программы

* Загружаются в память обычно в период инициализации компьютера и ожидают некоторого системного события (например, нажатия на клавиатуре комбинации «горячих клавиш»).
* Для написания таких служебных программ не существует единых правил. Поэтому некоторые из них могут вступать в конфликты с прикладными программами и даже с самой ОС.

Драйверы устройств

являются еще одним потенциальным источником конфликтов.

* При проведении большинства аппаратных модернизаций приходится устанавливать в систему один или несколько драйверов.
* В командных строках загрузки драйверов обычно присутствуют дополнительные ключи, с помощью которых задаются используемые этими драйверами системные ресурсы, что, в свою очередь, необходимо для обеспечения нормальной работы обслуживаемых драйверами устройств.
* Если ключи в командной строке загрузки драйвера не соответствуют аппаратной настройке соответствующего устройства (или частично или полностью совпадают с параметрами другого драйвера), то может возникнуть конфликт.
* Не исключено, что вызывающая конфликт программа написана с ошибкой.
* Свяжитесь с ее разработчиками. Выясните, не выпущена ли корректирующая программа или новая версия вашего ПО, в которой устранены недостатки предыдущей.

Аппаратные конфликты

* Во многих случаях аппаратные конфликты не приводят к полной потере работоспособности системы.
* Теоретически устранить конфликт очень просто — для этого нужно перенастроить параметры одного из конфликтующих устройств или программы.
* Трудность заключается в том, что сначала необходимо выяснить, какие ресурсы уже используются, а какие свободны.

Средства диагностики

Диагностические программы

Диагностические программы, проверяющие наиболее важные параметры оборудования, позволяют предотвратить подавляющее большинство проблем, возникающих по вине аппаратного обеспечения.

Диагностические программы подразделяются по нескольким основаниям.

* Диагностическое ПО включает в себя стандартное ПО, встроенное в состав ОС, и сторонние утилиты.
* По широте использования выделяют комплексные и специализированные утилиты.

Комплексные утилиты

* Комплексные утилиты предназначены для получения подробной информации об установленных компонентах, позволяют провести тестирование их на быстродействие, работоспособность, производительность, наличие ошибок и конфликтов и т. д.
* AIDA, ASTRA (Advanced SysInfo Tool and Reporting Assistant).

Специализированные утилиты

* Специализированные утилиты — более узкого назначения. Позволяют получить более детальную информацию о конкретном устройстве, произвести его настройку, включить/отключить различные режимы работы, которые не доступны с помощью стандартных средств.
* Специализированные утилиты в свою очередь классифицируются по предназначению, например, для тестирования памяти, проверки видеосистемы, проверки жесткого диска.
* MemTech (для тестирования оперативной памяти)
* HD Tach, SIGuardian (для получения информации о жестких дисках)
* CPUInfo (для тестирования процессора).

К аппаратным средствам диагностики неисправностей средств вычислительной техники (СВТ) относятся:

* универсальные измерительные приборы (тестеры, осциллографы, мультиметры и др.);
* специальная сервисная аппаратура для СВТ (диагностическая плата POST Card, диагностические заглушки для проверки портов Loop Back).

1. **Аппаратно-программные платформы серверов и рабочих станций. Установка серверной части.**

Аппаратно-программные платформы серверов и рабочих станций

* В качестве серверов в средних и крупных информационных системах используют специализированные многопользовательские мощные компьютеры — серверы.
* В информационных системах с небольшим количеством пользователей и малыми объемами информации в качестве сервера вполне может использоваться и персональный компьютер, обладающий приемлемыми техническими характеристиками.
* Первый признак, по которому подразделяются компьютеры, —платформа.
* Как правило, различные платформы компьютеров несовместимы между собой, — хотя в отдельных случаях программы, написанные для компьютеров одного типа, можно запустить на другом с использованием специальных программ-эмуляторов.

Основными достоинствами персональных компьютеров являются:

* небольшие физические габаритные размеры;
* мощные вычислительные возможности;
* простота эксплуатации пользователем-непрофессионалом;
* невысокая стоимость;
* отсутствие серьезных требований и ограничений по условиям эксплуатации.

Платформа IBM.

Платформа IBM-совместимых компьютеров включает в себя громадный спектр самых различных компьютеров, от простеньких домашних до сложных серверов.

Платформа Apple.

В отличие от IBM компания Apple всегда делала ставку на закрытую архитектуру — комплектующие и программы для этих компьютеров выпускались лишь небольшим числом авторизованных производителей.

Именно на компьютерах Apple впервые появились многие новинки, со временем ставшие неотъемлемой частью персонального компьютера: графический интерфейс и мышь, звуковая подсистема и компьютерное видео и  т. д.

В конструкцию серверов включены возможности для непрерывного наращивания, избыточные вентиляторы и источники питания, жесткие диски с возможностью «горячей» замены и встроенный на системной плате сервисный процессор.

Важнейшее направление компьютерных систем — их объединение в высоконадежные и информационно-безопасные структуры — кластеры

Кластерная структура сервера. Кластер представляет собой многомашинный компьютерный комплекс, который с точки зрения пользователя:

* является единой системой;
* обеспечивает высокую надежность (отказоустойчивость);
* имеет общую файловую структуру;
* обладает свойством эффективной масштабируемости — производительности при добавлении ресурсов;
* управляется (администрируется) как единая система.

Иногда *кластером называют комплекс* из двух компьютеров, один из которых делает полезную работу, а другой включен и находится в  «горячем» резерве.

Главные же качества кластеров — высокая надежность и масштабируемость. В отличие от систем с «горячим резервированием» все компьютеры в кластере не простаивают, выполняют полезную работу.

Установка серверной части

* Для того чтобы компьютер мог выступать в роли сетевого сервера необходимо установить серверную часть сетевой операционной системы, которая позволяет поддерживать ресурсы и распространять их среди сетевых клиентов.
* Важным вопросом для сетевых серверов является возможность ограничить доступ к сетевым ресурсам. Это называется сетевой защитой (network security).

Установку системы Windows Server можно производить в  одном из трех режимов:

1) ручная установка, в процессе которой администратор отвечает на все вопросы Мастера установки системы;

2) полуавтоматическая установка (с минимальным участием администратора в процессе установки);

3) автоматическая установка.

Процедура настройки Server 2012.

При выполнении процедуры настройки необходимо проверить, чтобы на компьютере корректно была настроена сетевая карта.

Затем нужно выполнить следующие действия.

1. После загрузки Microsoft Windows Server 2012 открыть Диспетчер серверов

2. Выбрать пункт Добавить роли и компоненты

3. В разделе «выбора» типа установки оставить галочку Установка ролей или компонентов, так как пока что служба удаленных рабочих столов не нужна, нажать Далее

4. Выбрать тот сервер, к которому применяются изменения, здесь ничего пока не нужно менять; нажать Далее

5. В появившимся Мастере найти строку Доменные службы Active Directory, выделить ее, после открывшегося меню нажать Добавить компоненты. Для продолжения работы Мастера нажать Далее Продолжить установку, нажав Далее

6. Вниманию будет представлена краткая информация о том, что такое доменные службы, для чего они нужны и на что следует обратить внимание.

7. В поле подтверждения установки компонентов активизировать Установить

Установка занимает определенное время, и после того, как она завершится, появится сообщение о том, что требуется дополнительная настройка, но так как инсталляция завершилась успешно, следует просто закрыть окно.

8. Для продолжения настройки нужно добавить роль DNS, войдя для этого в Диспетчер серверов / Добавить роли и компоненты. Дойдя до вкладки Роли сервера, следует найти и добавить роль DNS сервера путем Добавления компонентов

9. Для подтверждения установки компонентов нажать кнопку Установить

1. **Виды серверного программного обеспечения. Особенности эксплуатации различных видов серверного программного обеспечения.**

Виды серверного программного обеспечения

Серверное программное обеспечение — в информационных технологиях — программный компонент вычислительной системы, выполняющий сервисные (обслуживающие) функции по запросу клиента, предоставляя ему доступ к определенным ресурсам или услугам.

Для взаимодействия с клиентом сервер выделяет необходимые ресурсы межпроцессного взаимодействия (разделяемая память, пайп, сокет и др.) и ожидает запросы на открытие соединения.

Универсальные серверы — особый вид серверной программы, не предоставляющий никаких услуг самостоятельно. Вместо этого универсальные серверы предоставляют серверам услуг упрощенный интерфейс к ресурсам межпроцессного взаимодействия и (или) унифицированный доступ клиентов к различным услугам.

Универсальные серверы часто используются для написания всевозможных информационных серверов, которым не нужна какая-то специфическая работа с сетью, серверов, не имеющих никаких задач, кроме обслуживания клиентов.

Сервер маршрутизациине является сервером в классическом смысле, а является базовой функцией поддержки сети ОС.

Маршрутизацию своих пакетов к месту назначения выполняет любая система в сети, маршрутизацию же чужих пакетов осуществляют только маршрутизаторы (также известные как роутеры или шлюзы)

Решения динамической маршрутизации призваны собирать информацию о текущем состоянии сложной сети и поддерживать таблицу маршрутов через эту сеть, чтобы обеспечить доставку пакета по кратчайшему и самому эффективному маршруту.

Файл-серверыпредставляют собой серверы для обеспечения доступа к файлам на диске сервера. Прежде всего это серверы передачи файлов по заказу, по протоколам FTP, TFTP, SFTP и HTTP.

Недостатками файл-серверной системы являются**:**

* очень большая нагрузка на сеть, повышенные требования к пропускной способности. На практике это делает практически невозможной одновременную работу большого числа пользователей с большими объемами данных;
* осуществление обработки данных на компьютере пользователей. Это влечет повышенные требования к аппаратному обеспечению каждого пользователя. Чем больше пользователей, тем больше денег придется потратить на оснащение их компьютеров;

Серверы доступа к даннымобслуживают базу данных и отдают данные по запросам. Один из самых простых серверов подобного типа — LDAP (Lightweight Directory Access Protocol — облегченный протокол доступа к спискам).

* Для доступа к серверам баз данных единого протокола не существует, однако все серверы баз данных объединяет использование единых правил формирования запросов — язык SQL (Structured Query Language — язык структурированных запросов

Службы обмена сообщениями позволяют пользователю передавать и получать сообщения (обычно — текстовые). В первую очередь это серверы электронной почты, работающие по протоколу SMTP. SMTP-сервер принимает сообщение и доставляет его в локальный почтовый ящик пользователя или на другой SMTP-сервер

Серверы удаленного доступа через соответствующую клиентскую программу обеспечивают пользователя консольным доступом к удаленной системе. Для обеспечения доступа к командной строке служат серверы telnet, RSH, SSH.

Стандартный сервер удаленного доступа к графическому интерфейсу Microsoft Windows называется терминальным сервером. Некоторую разновидность управления (точнее мониторинга и конфигурирования) также предоставляет протокол SNMP. Компьютер или аппаратное устройство для этого должно иметь SNMP-сервер.

Игровой сервер служит для одновременной игры нескольких пользователей в единой игровой ситуации.

Серверные решения — операционные системы и (или) пакеты программ, оптимизированные под выполнение компьютером функций сервера и (или) содержащие в своем составе комплект программ для реализации типичного набора сервисов.

1. **Виды клиентского программного обеспечения. Установка, адаптация и сопровождение клиентского программного обеспечения.**

Виды клиентского программного обеспечения

Программное обеспечение – это совокупность программных и документальных средств для создания и эксплуатации систем обработки данных средствами вычислительной техники.

Клиентское ПО – это ПО, осуществляющее доступ к ресурсам, которые предоставляет сервер.

Для работы с сетью на клиентских рабочих станциях должно быть установлено клиентское ПО.

Компоненты клиентского ПО:

* + Редиректоры
  + Распределители
  + Имена UNC

Редиректоры

* Редиректор – сетевое программное обеспечение, которое принимает запросы ввода-вывода для удаленных файлов, именованных каналов или почтовых слотов и затем переназначает их сетевым сервисам другого компьютера.
* Редиректор перехватывает все запросы, поступающие от приложений, и анализирует их.
* По существу, редиректоры скрывают от пользователя сложность доступа к сети. После того как сетевой ресурс определен, пользователи могут получить к нему доступ без знания его точного расположения.

Распределители

* Распределитель представляет собой часть ПО, управляющую присвоением букв накопителя как локальным, так и удаленным сетевым ресурсам или разделяемым дисководам, что помогает во взаимодействии с сетевыми ресурсами.
* Когда между сетевым ресурсом и буквой локального накопителя создана ассоциация, известная также как отображение дисковода, распределитель отслеживает присвоение такой буквы дисковода сетевому ресурсу. Затем, когда пользователь или приложение получат доступ к диску, распределитель заменит букву дисковода на сетевой адрес ресурса, прежде чем запрос будет послан редиректору.

Имена UNC

* Большинство современных сетевых ОС распознают имена UNC (Universal Naming Convention — Универсальное соглашение по наименованию).
* UNC представляют собой стандартный способ именования сетевых ресурсов. Эти имена имеют форму \\Имя\_сервера\имя\_ресурса
* Способные работать с UNC приложения и утилиты командной строки используют имена UNC вместо отображения сетевых дисков.

Если компьютер предоставляет свои ресурсы другим пользователям сети, то он играет роль сервера.

При этом компьютер, обращающийся к ресурсам другой машины, является клиентом.

Компьютер, работающий в сети, может выполнять функции либо клиента, либо сервера, либо совмещать обе эти функции.

Если выполнение каких-либо серверных функций является основным назначением компьютера, то такой компьютер называется выделенным сервером.

В зависимости от того какой ресурс сервера является разделяемым, он называется файл-сервером, факс-сервером, принт-сервером, сервером приложений, сервером базы данных, веб-сервером и др.

Выделенный сервер не принято использовать в качестве компьютера для выполнения текущих задач, не связанных с его основным назначением, так как это может уменьшить производительность его работы как сервера.

В одноранговых сетях все компьютеры равны в правах доступа к ресурсам друг друга. Каждый пользователь может по своему желанию объявить какой-либо ресурс своего компьютера разделяемым, после чего другие пользователи могут его эксплуатировать.

В таких сетях на всех компьютерах устанавливается одна и та же ОС, которая предоставляет всем компьютерам в сети потенциально равные возможности.

Установка, адаптация и сопровождение клиентского программного обеспечения

Подготовка компьютера перед установкой

* Убедится, что конфигурация компьютера отвечает минимальным требованиям ПО.
* Проверить наличия свободного места на жёстком диске, если его не хватает, освободить.
* Отключить антивирусные программы.
* Остановить выполнение посторонних программ.
* Установка различного ПО может существенно различаться, но чаще всего установка приложения сводится к стандартному набору действий, в которых вам помогает мастер установки.
* В случае если программа уже была когда-то проинсталлирована на ПК мастер установки спросит, что делать со старой версией удалить, заменить или восстановить.
* После этого можно будет указать путь установки программы и завершить установку выполнив все действия, указанные мастером.
* Завершающим этапом установки, чаще всего, становится регистрация программного продукта.

Поддержка программного обеспечения

* Пакеты исправлений и заплатки
* Время от времени производители программного обеспечения выпускают исправления к своим программным продуктам. Как правило, исправления корректируют выявленную проблему, а затраты на их тестирование могут варьироваться в зависимости от программного продукта.
* Протестировав исправление и приняв решение о его развертывании, администратор копирует необходимые файлы в точку распространения ПО, производя замену старых файлов.
* Производители программного обеспечения распространяют обновления либо в виде нового пакета установщика, либо в качестве исправления установщика.

Пакеты обновления

* Между пакетами обновления и исправлениями нет больших отличий. Как правило, в состав пакета обновления входят несколько уже протестированных исправлений. Пакеты обновления распространяются реже, чем исправления, но чаще, чем полные обновления программного продукта.
* Рекомендуется распространять пакет обновления в качестве исправления, если он содержит небольшое количество файлов. Распространение пакета с большим количеством файлов и управление им рекомендуется рассматривать в качестве полного обновления программного продукта.
* В любом случае, следуйте инструкциям производителя заплатки и тестируйте ее в лаборатории или на небольшой группе пользователей, прежде чем осуществлять распространение заплатки для всех пользователей и компьютеров, управляемых заданным объектом групповой политики.

Обновления

* Обновление подразумевает замену значительного числа файлов. Поскольку производятся большие изменения в программном продукте, обновление получает другой номер версии.
* Производитель предоставляет новую версию в пакете установщика ПО, с заложенной в него информацией о том, какие более ранние версии он может обновлять.
* В пакете также должен содержаться сценарий процесса обновления, в котором описано, какие файлы должны быть заменены, удалены и добавлены.

1. **Многоуровневая модель качества программного обеспечения.**

Модели качества ПО

* Модели качества являются иерархическими моделями, созданные на основе критериев качества и связанных с ними показателей.
* Все модели качества могут быть подразделены на три категории в соответствии с методами, на основе которых они были созданы:
  1. Теоретические модели – основанные на гипотезе отношений между переменными качества.
  2. Модели управления данными – основаны на статическом анализе.
  3. Комбинированная модель – в которой интуиция исследователя используется для определения нужного вида модели, а анализ данных применяется для определения констант модели качества.

Модель МакКола

* Первая модель качества была предложена в 1977 г. Дж.МакКолом.
* Эта модель предназначена для определения полной характеристики качества ПО через его различные характеристики. Имеет три главных направления:
  1. Использование(корректность, надежность, эффективность, целостность, практичность)
  2. Модификация (тестируемость, гибкость, сопровождаемость).
  3. Переносимость (мобильность, возможность многократного использования, функциональная совместимость) — факторы качества, важные для переносимости программного продукта на другие аппаратные и программные платформы

Модель Боэма.

* Второй из основополагающих моделей качества является модель, предложенная в 1978 г. Б. Боэмом
* Модель качества Боэма представляет характеристики ПО в более крупном масштабе, чем модель МакКола.

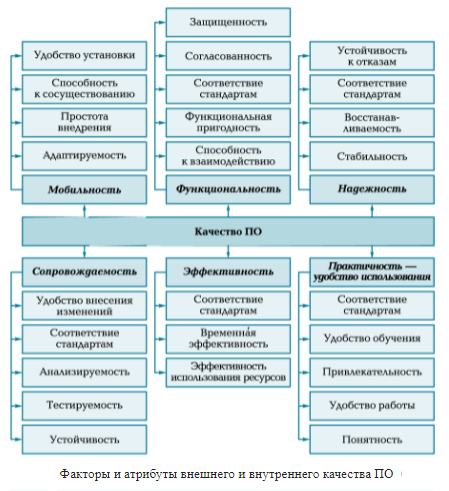
В этой модели:

1. Практичность описывает то, как легко, надежно и эффективно может быть использован пакет программ;
2. сопровождаемость характеризует, насколько легко изменить и повторно протестировать пакет программ;
3. мобильность описывает, как пакет программ может использоваться даже при изменении программных и аппаратных средств.

Модель качества ISO 9126.

В настоящее время наиболее распространенной и широко используемой является многоуровневая модель качества программного обеспечения, представленная в наборе стандартов ISO 9126.

На верхнем уровне выделяют 6 основных характеристик качества ПО, каждая из которых определяется набором атрибутов с соответствующими метриками для дальнейшей оценки



Качество ПО определяется в стандарте ISO 9126-1 как всякая совокупность его характеристик, относящихся к возможности удовлетворять высказанные или подразумеваемые потребности всех заинтересованных лиц.

Модель качества ISO 9126-1 различает понятия

1. внутреннего качества, связанного с характеристиками ПО самого по себе, без учета его поведения;
2. внешнего качества, характеризующего ПО с точки зрения его поведения;
3. качества ПО при использовании в различных контекстах — того качества, которое ощущается пользователями при конкретных сценариях работы ПО.

Для всех этих аспектов качества введены метрики, позволяющие оценить их.

Метрика качества программ — система измерений качества программ. Эти измерения могут проводиться на уровне критериев качества программ или на уровне отдельных характеристик качества*.*

Многоуровневый подход к моделям качества.

В основе моделей качества лежит многоуровневый подход, причем число уровней может быть 2 (модели МакКола и Боэма) или 3 включая метрики).

1. **Объекты уязвимости.**

Проблема безопасности программ

Проблема безопасности программ является одной из первостепенных в области информационной безопасности (ИБ), так как наличие уязвимостей в программных ресурсах информационных систем обуславливает возможность реализации промышленных компьютерных атак и вирусных эпидемий, а также причину разного рода непреднамеренных отказов и потери ресурсов.

С учетом динамичности и сложности программ и развития технологий информационного противоборства решение указанной проблемы требует непрерывного совершенствования методов контроля, тестирования и испытаний, а именно:

* статического и динамического анализа;
* функционального тестирования;
* экспертиз бюллетеней безопасности;
* сканирования уязвимостей;
* антивирусного контроля.

Понятие уязвимости и дефекта безопасности программ

Дефект безопасности - недостаток создания ПО, потенциально влияющий на степень безопасности информации.

В таком случае эксплуатируемый дефект безопасности представляет собой уязвимость, реализация которой составляет угрозу информационной безопасности.

Классификация уязвимостей программных систем

В настоящее время можно встретить ряд классификаций в области программной безопасности, которые можно разделить на следующие:

-классификации угроз и атак;

-классификации вредоносных программ;

-классификации и реестры уязвимостей;

-классификации дефектов

Классификации угроз и атак

Данные классификации являются самыми методически проработанными и систематизируют различные виды искусственных и естественных, случайных и злонамеренных, внутренних и внешних угроз по множеству всевозможных параметров.

Классификации выделяют класс угроз, связанный с возможностью реализации нарушителем программных уязвимостей.

Данные классификации являются основой для построения моделей угроз безопасности информации

Классификации вредоносных программ

Разработчики средств антивирусной защиты придерживаются классификаций «вредоносного» ПО по модели распространения, по способу активации, по действию и другим параметрам, что позволяет разрабатывать эффективные тесты и базы сигнатур антивирусов.

Данные классификации полезны при описании подкласса уязвимостей эксплуатационного типа.

Классификации и реестры уязвимостей

Исторически реестры уязвимостей были обусловлены потребностью в регулярном распространении бюллетеней и сводок о найденных уязвимостях для каждого типа и версии программных продуктов и сред.

Такие реестры поддерживаются крупными разработчиками ПО (например Adobe, Microsoft, RedHat).

Классификации дефектов

Большая часть дефектов возникает в процессе создания ПО. Это могут быть ошибки проектирования, ошибки кодирования программистов, ошибки, допущенные при сборке дистрибутива и интеграции различных версий компонентов ПО.

Развитие программных и информационных технологий обусловливает потребность в национальном стандарте, определяющем принципы систематики, таксономии и классификации уязвимостей и дефектов безопасности программных ресурсов.

Лучшие инструменты для тестирования на проникновение безопасности на рынке

1. Netsparker
2. Core Impact
3. Metasploit
4. **Дестабилизирующие факторы и угрозы надежности.**

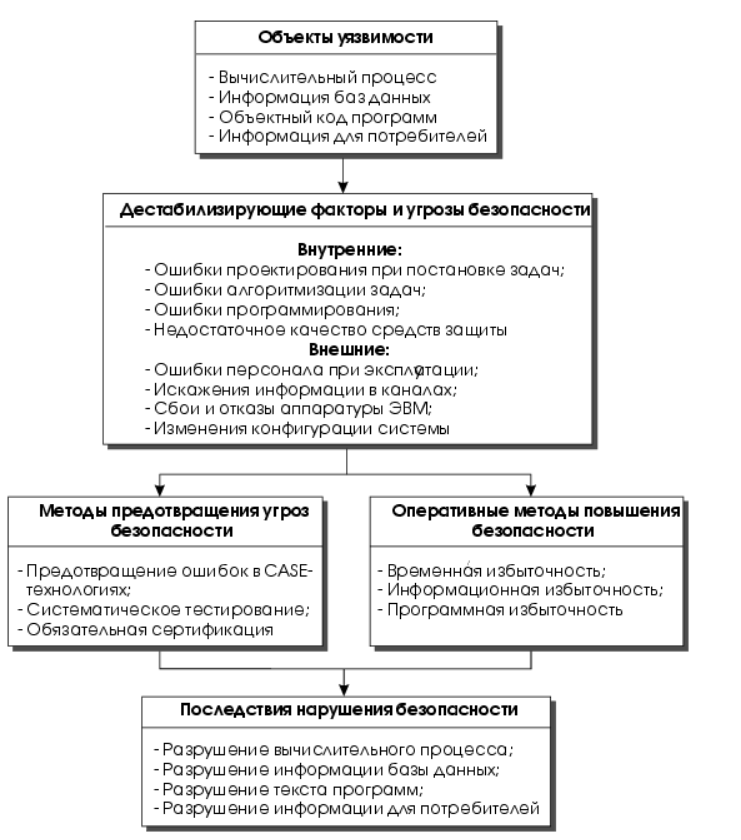
Объекты уязвимости

В качестве объектов уязвимости рассматриваются:

* динамический вычислительный процесс обработки данных, автоматизированной подготовки решений и выработки управляющих воздействий;
* информация, накопленная в базах данных;
* объектный код программ, исполняемых вычислительными средствами в процессе функционирования ИС;
* информация, выдаваемая потребителям и на исполнительные механизмы.



Модель анализа безопасности информационных систем при отсутствии злоумышленных угроз.



Внутренние дестабилизирующие факторы

* Различия между ожидаемыми и полученными результатами функционирования программ и баз данных могут быть следствием ошибок не только в созданных программах и данных, но и системных ошибок в первичных требованиях спецификаций, явившихся основой при создании ИС.
* В результате важной особенностью процесса выявления ошибок в программах и данных сложных, критических ИС является отсутствие полностью определенного эталона, которому должны соответствовать текст и результаты функционирования разработанной программы.
* При отладке и тестировании обычно сначала обнаруживаются вторичные ошибки, то есть последствия и результаты проявления некоторых дефектов, которые следует квалифицировать как первичные ошибки.
* Проявления ошибок в разной степени влияют на работоспособность программ и их нельзя целиком квалифицировать как отказы. В худшем случае вторичная ошибка проявляется как полный отказ — потеря работоспособности ПС и БД на длительное время, угрожающая безопасности.
* Изучение характеристик первичных и вторичных ошибок, а также их взаимосвязи важно для выработки стратегий проектирования безопасных ИС.

Характеристики в процессе проектирования программ и баз данных помогают:

* оценивать реальное состояние проекта и планировать трудоемкость и длительность его завершения;
* выбирать методы и средства автоматизации тестирования программ, адекватные текущему состоянию разработки ПС и наиболее эффективные для устранения определенных видов ошибок;
* прогнозировать эффективность средств оперативной защиты от не выявленных первичных ошибок;
* оценивать требующиеся дополнительные ресурсы ЭВМ с учетом затрат на устранение ошибок.

Внешние дестабилизирующие факторы

* ошибки оперативного и обслуживающего персонала в процессе эксплуатации ИС;
* искажения в каналах телекоммуникации информации, поступающей от внешних источников и передаваемой потребителям, а также недопустимые изменения характеристик потоков информации;
* сбои и отказы аппаратуры;
* изменения состава и конфигурации ИС за пределы, проверенные при испытаниях или сертификации.

Полное устранение перечисленных угроз принципиально невозможно.

* Задача состоит в выявлении факторов, от которых они зависят;
* В создании методов и средств уменьшения их влияния на безопасность ИС;
* В эффективном распределении ресурсов для обеспечения защиты, равнопрочной по отношению ко всем негативным воздействиям.

Внутренние дестабилизирующие факторы, при отсутствии злоумышленных воздействий оказывают наибольшее влияние на безопасность функционирования ИС.

1. **Методы предотвращения угроз надежности.**

Методы снижения угроз безопасности ИС, вызванных дефектами программных средств и баз данных

* Уровень и влияние внутренних дестабилизирующих факторов, а также некоторых внешних угроз на безопасность применения ИС определяется в наибольшей степени качеством технологий проектирования, разработки, сопровождения и документирования ИС и их основных компонент — программных средств и баз данных.

В современных автоматизированных технологиях создания и развития ПС и БД с позиции обеспечения их потенциальной технологической безопасности можно выделить методы и средства, позволяющие:

* предотвращать дефекты проектирования за счет эффективных технологий обеспечения всего жизненного цикла комплексов программ и данных;
* обнаруживать и устранять ошибки проектирования путем систематического тестирования на всех этапах жизненного цикла ПС и БД;
* удостоверять достигнутые качество и безопасность применения ПС и БД в процессе их сертификации перед передачей в эксплуатацию.
* Комплексное, скоординированное применение таких методов и средств в процессе создания и развития ПС и БД позволяет исключить некоторые виды угроз или значительно ослабить их влияние.

При создании критических ИС высокой сложностиважная проблема состоит в правильном системотехническом и информационно-технологическом проекте, обеспечивающем высокие потребительские свойства и безопасность ИС.

Одновременно в **силу высокого качества** проработки и документирования проекта *создается основа для снижения трудоемкости отладки, испытаний и особенно сопровождения и развития прикладной ИС.*

Совместное применение современных CASE-технологий и языков четвертого поколения способно снизить трудоемкость разработки сложных программных средств до 10 раз и сократить длительность их проектирования с 2-3 лет до нескольких месяцев.

Систематическое тестирование

* Тестирование является основным методом измерения качества и определения реальной безопасностиприменения программ и информации баз данных на любых этапах разработки.
* Результаты тестирования и измерения показателей качества сравниваются с требованиями технического задания или спецификаций для определения степени соответствия предъявлявшимся требованиям, полученным разработчиком от заказчика.
* Наличие таких достаточно полных эталонов, как совокупность требований технического задания и поэтапная их декомпозиция в спецификациях, является необходимой базой тестированияпри промежуточных и завершающих испытаниях.
* Непосредственной целью тестирования является обнаружение, локализация и устранение ошибок в программах и данных.
* Важной особенностью тестирования сложных критических ИС является необходимость достаточно полной их проверки при ограниченной длительности испытаний.
* Это определяет целесообразность тщательного планирования тестирования с учетом всех результатов, полученных на предыдущих этапах разработки.
* При планировании *основная задача состоит в достижении максимальной достоверности испытаний*, *определении качества и безопасности ИС* при ограниченных затратах ресурсов всех видов на проведение тестирования

Для обеспечения высокого качества целесообразно проводить испытания не только завершенной ИС, но на ряде промежуточных этапов разработки проверять состояние и характеристики компонент проекта.

Обязательная сертификация

* Для удостоверения качества и безопасности применения сложных критических ИС используемые в них ПС и БД следует подвергать обязательной сертификации.
* Однако сертификация удостоверяет качество и безопасность применения ИС только в условиях, ограниченных конкретными стандартами и нормативно-техническими документами.
* В реальных условиях эксплуатации принципиально возможны отклонения характеристик внешней среды функционирования ИС за пределы, ограниченные сертификатом, и ситуации, не проверенные при сертификационных испытаниях.
* Эти обстоятельства способны вызывать последствия, угрожающие безопасности функционирования ИС.

Наличие сертификата у ПС и БД для критических систем является необходимым условием их допуска к эксплуатации, однако любой сертификат на сложные системы не может гарантировать абсолютную безопасность их применения и всегда остается некоторый риск возникновения катастрофических ситуаций**.**

1. **Оперативные методы повышения надежности: временная, информационная, программная избыточность.**

Оперативные методы повышения надежности

* Невозможность обеспечить в процессе создания ИС ее абсолютную защищенность даже при отсутствии злоумышленных воздействий заставляет искать дополнительные методы и средства повышения безопасности функционирования ПС и БД на этапе эксплуатации.
* Для этого разрабатываются и применяются методы оперативного обнаружения дефектов при исполнении программ и искажений данных путем введения в них временной, информационной и программной избыточности.
* Эти же виды избыточности используются для оперативного восстановления искаженных программ и данных и предотвращения возможности развития угроз до уровня, нарушающего безопасность функционирования ИС.
* Для обеспечения высокой надежности и безопасности функционирования ПС и БД необходимы вычислительные ресурсы для максимально быстрого обнаружения проявления дефектов, возможно точной классификации типа уже имеющихся и вероятных последствий искажений, а также для автоматизированных мероприятий, обеспечивающих быстрое восстановление нормального функционирования ИС.
* Неизбежность ошибок в сложных ИС, искажений исходных данных и других аномалий приводит к необходимости регулярной проверки состояния и процесса исполнения программ, а также сохранности данных.

Временная избыточность

* Временная избыточность состоит в использовании некоторой части производительности ЭВМ для контроля исполнения программ и восстановления (рестарта) вычислительного процесса.
* Для этого при проектировании ИС должен предусматриваться запас производительности, который будет затем использоваться на контроль и повышение надежности и безопасности функционирования.
* Величина временной избыточности может лежать в  интервале от единиц процентов производительности  компьютера до  3…4-кратного дублирования производительности отдельной машины в многопроцессорных вычислительных системах.

Информационная избыточность

* Информационная избыточность состоит в дублировании накопленных исходных и промежуточных данных, обрабатываемых программами.
* Избыточность используется для сохранения достоверности данных, которые в наибольшей степени влияют на нормальное функционирование ИС и требуют значительного времени для восстановления.

Программная избыточность

* Программная избыточность используется для контроля и обеспечения достоверности наиболее важных решений по управлению и обработке информации.
* Она заключается в сопоставлении результатов обработки одинаковых исходных данных программами, различающимися используемыми алгоритмами, и в исключении искажений при несовпадении результатов.
* Программная избыточность необходима также для реализации средств автоматического контроля и восстановления данных с использованием информационной избыточности и для функционирования всех средств защиты, использующих временную избыточность.

Средства, использующие временную избыточность

Средства, использующие временную избыточность:

* авторизация доступа пользователей к системе,
* анализ доступных пользователю ресурсов, выделение ресурсов согласно ролям и уровням подготовки пользователей,
* разграничение прав доступа пользователей к отдельным задачам,
* функциям управления, записям и полям баз данных

Средства обеспечения надежности, использующие информационную избыточность

Средства обеспечения надежности, использующие информационную избыточность:

* ссылочная целостность баз данных, которая обеспечивается за счет системы внутренних уникальных ключей для всех информационных записей системы;
* открытая система кодирования, позволяющая пользователю в  любой момент изменять коды любых объектов классификации, которая обеспечивает стыковку системы классификации АИС делопроизводства с  ПО других разработчиков;
* механизмы проверки значений контрольных сумм записей системы, обеспечивающие выявление всех несанкционированных модификаций (ошибок, сбоев) информации.

Способы обеспечения и повышения надежности ПО

К способам обеспечения и повышения надежности ПО относятся:

* усовершенствование технологии программирования (например, формальное описание этапов программирования с  помощью языка UML);
* выбор алгоритмов, не чувствительных к различного рода нарушениям вычислительного процесса (использование алгоритмической избыточности);
* резервирование программ — N-версионное программирование;
* верификация и валидация программ с последующей коррекцией

Валидация – гарантированная уверенность производителя в том, что он создал продукт по всем необходимым стандартам.

Верификация – помогает увериться в том, что изделие соответствует всем изначально заданным требованиям к нему.

В зависимости от степени проявления и причин обнаружения искажений возможны следующие оперативные меры для ликвидации их последствий, восстановления информации и сохранения безопасности процессов обработки данных и управления:

* игнорирование обнаруженного искажения вследствие слабого влияния на весь процесс функционирования и выходные результаты;
* исключение сообщения из обработки вследствие его искажения или трудности предстоящего восстановления вычислительного процесса;
* повторение функциональной группы программ при тех же исходных данных или восстановление данных в процессе последующей обработки;
* кратковременное прекращение решения задач данной группы прикладных программ до обновления исходных данных;
* перестройка режима работы или структуры ИС для снижения влияния перегрузки или в связи с потерей информации о ходе процесса обработки данных и управления;
* переход на резервную ЭВМ с накопленной информацией о ходе процесса управления или восстановление информации за счет ее дублирования;
* восстановление процесса управления или обработки информации с режима начального пуска всей ИС с оперативным вмешательством обслуживающего персонала.

Введение избыточности в программы и данные способствует повышению качества функционирования ИС.

Средства оперативной защиты вычислительного процесса, программ и данных, в свою очередь, являются сложными системами и не застрахованы от ошибок, способных привести к нарушению безопасности функционирования ИС.

Поэтому необходим комплексный анализ, распределение ресурсов и видов избыточности для максимальной безопасности применения критических ИС.

1. **Первичные ошибки, вторичные ошибки и их проявления.**

Ошибка

Под ошибкой в широком смысле слова понимается непра­вильность, погрешность или неумышленное, невольное иска­жение объекта или процесса.

Источники ошибок:

* *Плохая спецификация.* (Плохо представили назначение программы → невозможно предусмотреть обработку всех ошибок)
* *Неполные программы.* В ходе разработки неизбежно возникают варианты, которые мы не предусмотрели.
* *Непредусмотренные аргументы.* Если функция принимает аргумент, который не был предусмотрен, то возникнет проблема.
* *Непредусмотренные входные данные.*
* *Неожиданное состояние.* Большинство программ хранит большое количество данных ("состояний"): списки адресов, каталоги телефонов и данные о температуре, записанные в объекты типа vector.
* *Логические ошибки.* Эти ошибки приводят к тому, что программа просто делает не то, что от нее ожидается.

Уровни ошибок

* По уровням ошибки подразделяют на первичные, вторичные, небольшие, умеренные и критические.
* Критические ошибки с  высоким влиянием, останавливают выпуск версии программного продукта.
* При отладке и тестировании ПО обычно сначала обнаруживаются вторичные ошибки, т. е. последствия внутренних его дефектов, которые следует квалифицировать как первичные причины обнаруженных аномалий результатов.
* Вторичные ошибки по-разному влияют на общую эффективность СПК(сложных программных комплексов) и являются определяющими при определении эффективности его функционирования.
* Не каждая первичная ошибка вносит заметный вклад в выходные результаты, вследствие чего ряд из них может оставаться не обнаруженным.
* Наибольшее число наиболее сложных для обнаружения и  устранения первичных ошибок вносится на этапах системного анализа и разработки модификаций программ.
* На последующих этапах разработки изменений ПО ошибки вносятся и устраняются в программах в процессе их корректировки по результатам тестирования.

Классы ошибок

* системные ошибки;
* ошибки в выборе алгоритма;
* алгоритмические ошибки;
* технологические ошибки;
* программные ошибки.

Системные ошибки

* Системные ошибкив большом (сложном) программном обес­печении определяются, прежде всего неполной информацией о реальных процессах, происходящих в источниках и потребителях информации.
* Некачественное определение требований к программе при­водит к созданию программы, которая будет правильно решать неверно сформулированную задачу. В таких случаях, как правило, требуется полное перепрограммирование.

Ошибки в выборе алгоритма*.*

К *алгоритмическим ошибкам* следует отнести, прежде всего, ошибки, обусловленные некорректной постановкой функцио­нальных задач, когда в спецификациях не полностью оговорены все условия, необходимые для получения правильного результата. Эти условия формируются и уточняются в значительной части в процессе тестирования и выявления ошибок в результатах функ­ционирования программ.

Технологические ошибки

* Технологические ошибки— это ошибки документации и фик­сирования программ в памяти ЭВМ.
* Боль­шинство технологических ошибок выявляются автоматически формализованными методами (например, транслятором).

Программные ошибки

* Синтаксис — это набор правил построения из символов алфавита специальных конструкций, с помощью которых можно составлять различные алгоритмы (программы).
* Семантика языка — это система правил истолкования пост­роений конструкций.

Как избежать ошибок?

* Чтобы решить данную проблему, необходимо проводить тестирование.
* Один из инструментов – отладчик. Он позволяет в автоматическом режиме контролировать работоспособность программы и сразу предупреждает о проблемах. Еще одно решение - unit-тесты. Специалист описывает ситуации для каждого компонента, и указывает, какой результат ожидается. Потом запускается проверка, и если результат не совпадает с ожидаемым, появляется предупреждение. Далее программисты находят и устраняют проблему.
* В ходе альфа-тестирования команда разработчиков имитирует деятельность пользователей и пытается найти проблемные участки. В ходе бета-тестирования программа предоставляется пользователям для реального использования. Все эти тесты помогают запустить продукт, минимизируя риски ошибок.

1. **Математические модели описания статистических характеристик ошибок в программах.**

Математические модели ошибок в программах

Математические модели позволяют оценивать характеристики ошибок в программах и прогнозировать их надёжность при проектировании и эксплуатации.

Для построения математических моделей надежности следует помнить, что модели имеют вероятностный характер и дают удовлетворительные результаты при относительно высоких уровнях проявления ошибок (т. е. при невысокой надежности ПО).

Математические модели надежности программных средств подразделяются на аналитические и эмпирические.

* Аналитические модели дают возможность рассчитать количественные показатели надежности, основываясь на данных о поведении программы в процессе тестирования.
* Эмпирические модели базируются на анализе структурных особенностей программ

Аналитические модели

Аналитические модели, в свою очередь, делятся на две группы: динамические и статические.

* В динамических моделях поведение ПО, т. е. появление отказов, анализируется во времени. От способа фиксации момента отказа модели могут быть модели с непрерывным временем или модели с дискретным временем.
* В статических моделях учитывают зависимость количества ошибок от числа тестовых прогонов.

Эти математические модели предназначены для оценки:

- показателей надёжности комплексов программ в процессе отладки;

- количества ошибок, оставшихся не выявленными;

- времени, необходимого для обнаружения следующей ошибки в функционирующей программе;

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

Аналитические - динамические модели

Использование моделей позволяет эффективно и целеустремлённо проводить отладку и испытания комплексов программ, помогает принять рациональное решение о времени прекращения отладочных работ.

В настоящее время предложен ряд математических моделей, основными из которых являются:

- экспоненциальная модель изменения ошибок в зависимости от времени отладки;

- модель, учитывающая дискретно - понижающуюся частоту появления ошибок как линейную функцию времени тестирования и испытаний;

- модель, базирующаяся на распределении Вейбула;

Аналитические - статические модели

К аналитическим статическим моделям относятся, например, модели Миллса, Липова, Нельсона.

Для использования модели Миллса характерно внесение некоторого количества искусственных ошибок перед началом тестирования.

Ошибки вносятся случайным образом и фиксируются в протоколе искусственных ошибок. Специалист, проводящий тестирование, не знает ни количества, ни характера внесенных ошибок до момента оценки показателей надежности по модели Миллса.

Предполагается, что все ошибки (как естественные, так и искусственно внесенные) имеют равную вероятность быть найденными в процессе тестирования.

Эмпирических моделей

Что касается эмпирических моделей, то они, как правило, основываются на сложности ПО и характеризуются размером программного обеспечения:

* количеством программных модулей,
* количеством и сложностью межмодульных интерфейсов.

Под программным модулем в данном случае понимают программную единицу, выполняющую определенную функцию и взаимосвязанную с другими модулями ПО

1. **Анализ рисков и характеристик качества программного обеспечения при внедрении.**

Оценка риска проектов программного обеспечения

Риск (по определению SEI (Software Engineering Institute)) - это возможность понести потери.

Риск проекта ПО — это возможность:

1) снижения качества конечного продукта,

2) повышения стоимости его разработки,

3) задержки окончания разработки или срыва проекта (то есть, отказа от проекта).

Управление риском

Эффективное управление риском состоит в принятии (по каждому риску) компромиссных решений по:

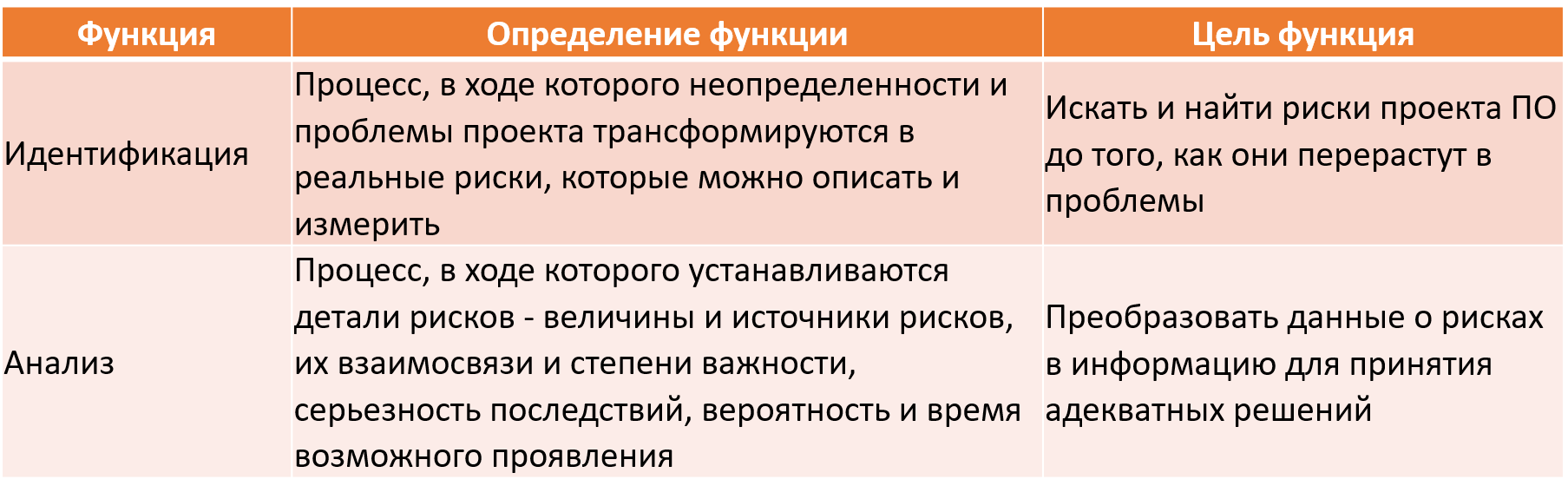
* учету рисков и анализу старых проектов;
* оценке трудоемкости устранения определенного риска,
* величине потенциального отрицательного воздействия этого риска на проект,
* резервировании в проекте времени на борьбу с рисками.

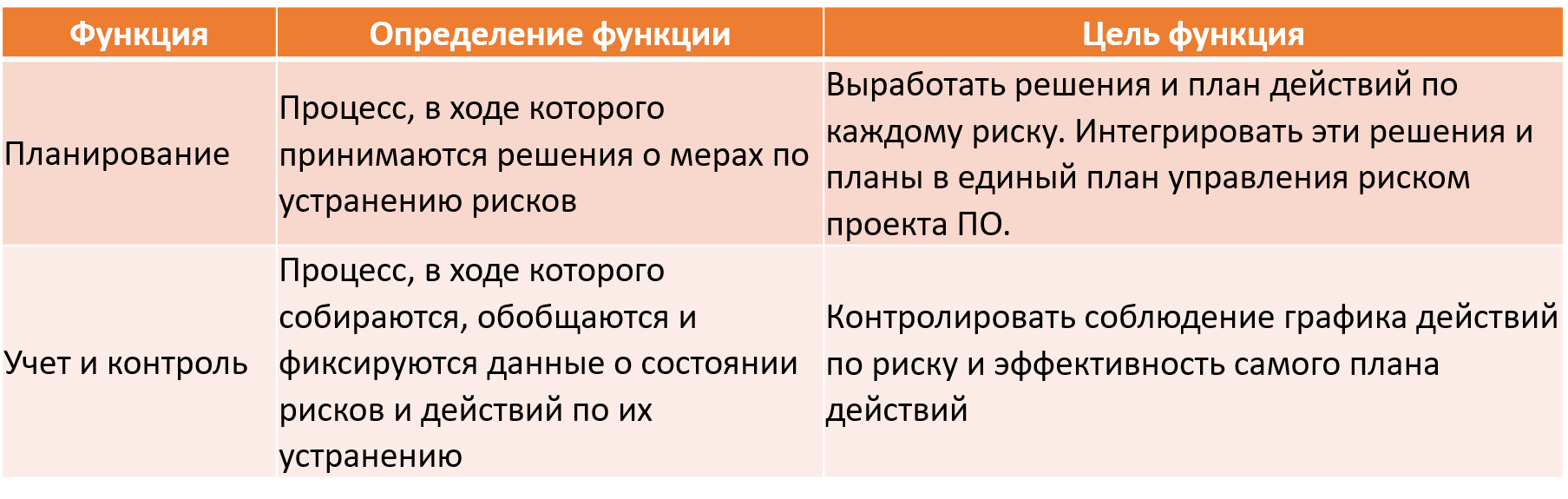
Концепция управления риском проекта ПО

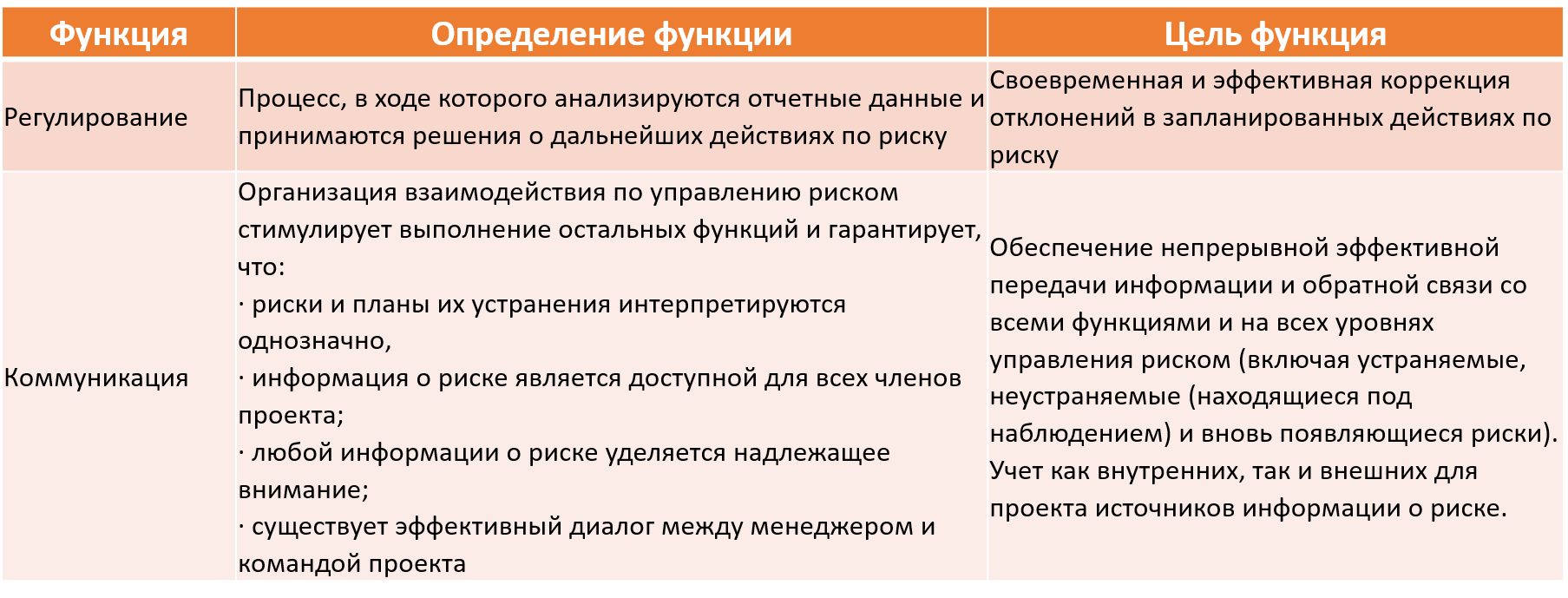
Базовыми конструкциями концепции управления риском являются:

* функции управления риском,
* таксономия (классификация) риска;
* методология оценки и управления риском

Функции управления риском







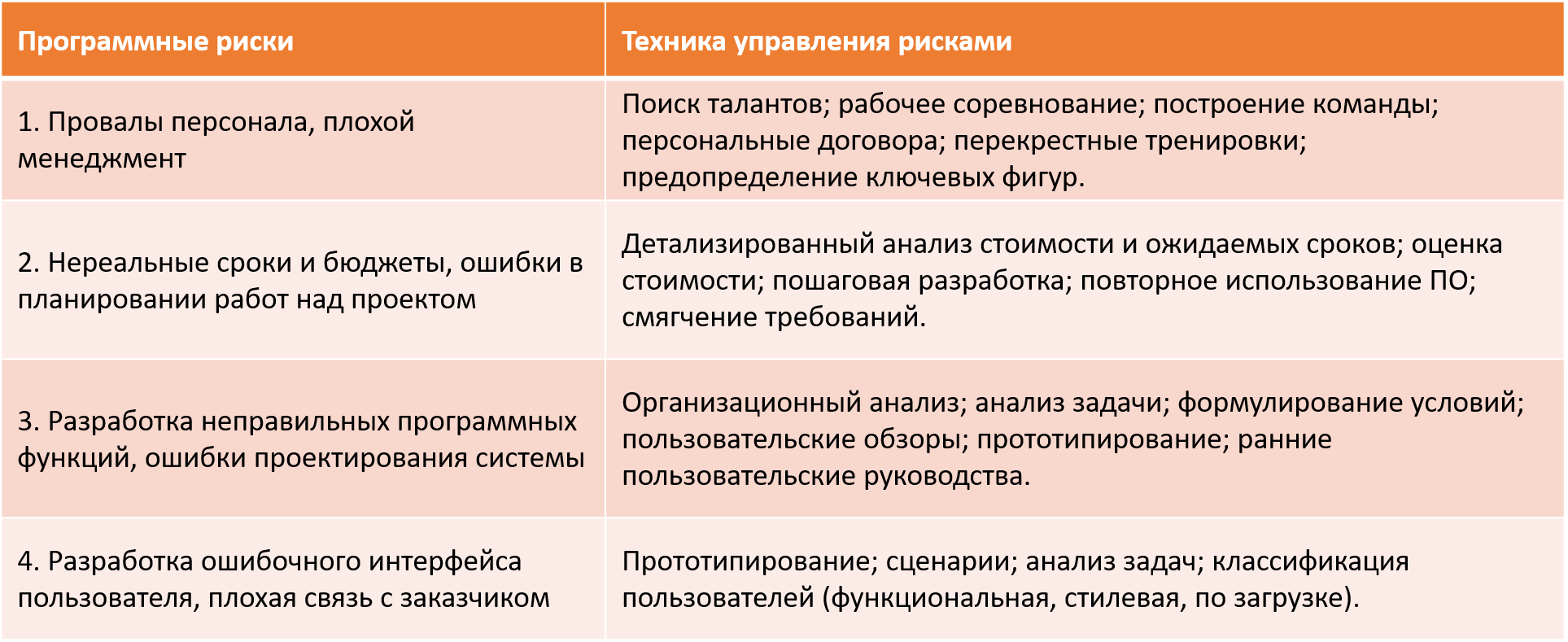
Таксономия (классификация) риска

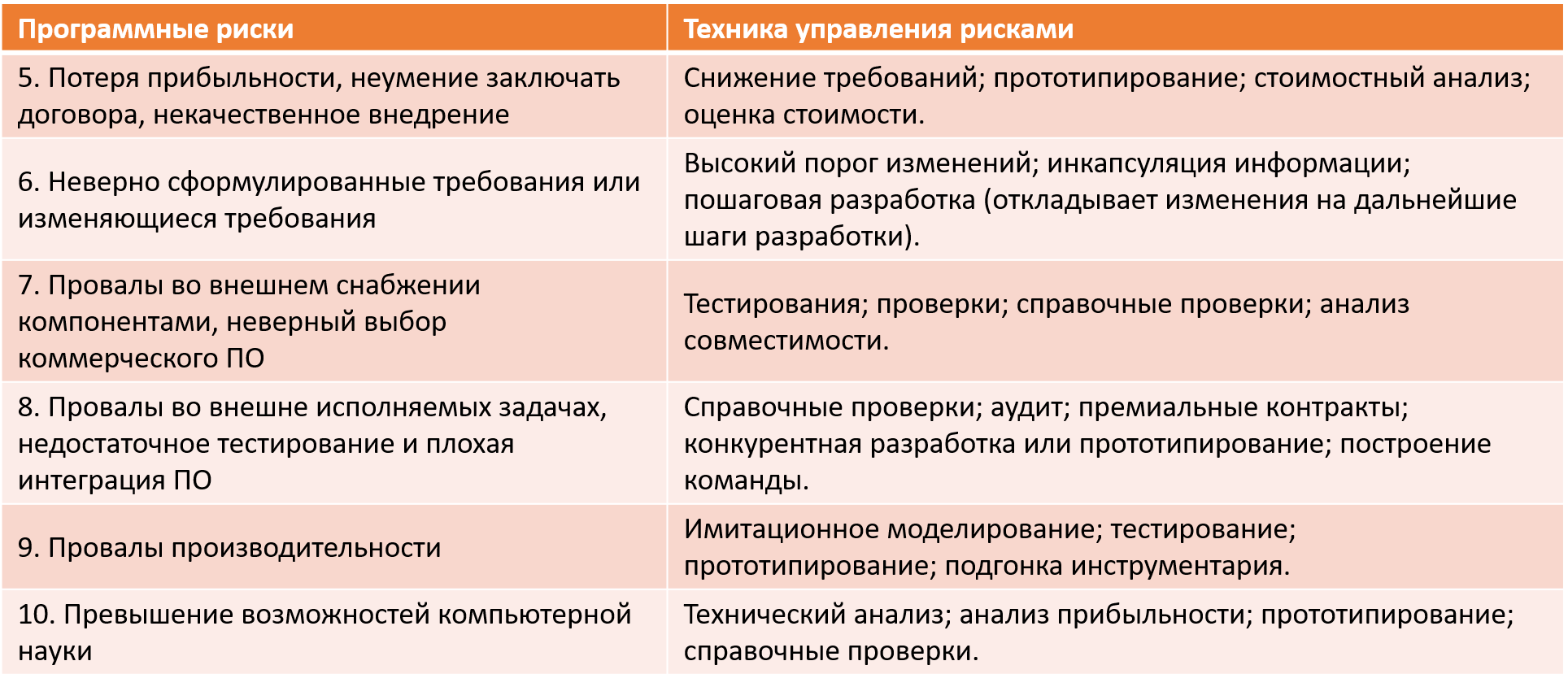
* Таксономия риска обеспечивает базис для организации данных и изучения различных аспектов риска проекта ПО
* Таксономия риска разрабатывалась SEI в течение трех лет и была проверена на более чем 30 проектах ПО. Она составлена с учетом типовых процессов жизненного цикла (ЖЦ) ПО и охватывает наиболее общие области риска проекта, касающиеся характеристик ПО, среды и процессов разработки и ограничений проекта.

Таксономия риска SEI имеет иерархическую структуру и систематизирует источники (области) риска по трем уровням:

* класс
* элемент класса
* атрибут элемента

Список 10 главных программных рисков





Методология оценки и управления риском

Исследования риска — это длительный (несколько месяцев) процесс, в ходе которого предпринимаются совместные усилия ведущей организацией по вопросам управления риском (в рамках страны, определенной отрасли или государственной структуры) и организацией-клиентом:

* по изучению существующей практики разработки конкретного проекта,
* оценке альтернативных приемов управления риском,
* выработке концепции управления риском клиента,
* обучению методам управления риском
* созданию необходимой инфраструктуры и плана управления риском ПО в организации-клиенте.

Процесс управления риском проекта ПО

Процесс управления риском проекта ПО включает следующие четыре стадии:

* согласование целей - определение нужд и целей проекта, достижение соглашений по управлению риском,
* подготовка работ - планирование и координация предстоящих работ по оцениванию риска проекта ПО,
* оценка риска - выполнение функций управления риском и получение рекомендаций по управлению риском проекта ПО,
* подготовка к устранению риска - разработка рекомендаций по устранению рисков по всем областям устранения риска, разработка плана управления риском и приведение его в действие.

Разработано 3 методологии:

1. Оценивание риска ПО - SRE (от Software Risk Evaluation),
2. Непрерывное управление риском - CRM (от Continuous Risk Мanagement),
3. Коллективное управление риском - TRM (от Team Risk Management).

Оценивание риска ПО

Методология SRE предлагает формальный метод идентификации, анализа, контроля и устранения риска ПО, который применяется первоначально на самой ранней стадии разработки проекта ПО (еще до заключения договора с разработчиком) и затем периодически в ходе всего ЖЦ проекта.

К основным функциям управления риском относятся:

* обнаружение,
* спецификация,
* оценивание,
* структурирование (консолидация)
* устранение рисков.

Непрерывное управление риском

Методология основана на определенных принципах управления риском в ходе всего ЖЦ проекта и не зависит от конкретных применяемых методов и инструментов оценки и устранения риска.

Коллективное управление риском

* Эта методология определяет дополнительные действия в деятельности по управлению риском, которые связаны с осуществлением совместного управления риском со стороны заказчика проекта ПО и его исполнителя.
* Ее применение обеспечивает формирование среды, содержащей множество процессов, методов и инструментов, необходимых для совместной работы заказчика и исполнителя над непрерывным управлением риском в ходе ЖЦ проекта.

Основные положения трех методологий, представленных выше, послужили базисом для формирования подхода к управлению риском проектов ПО в терминах процесса управления риском и стадий выполнения, а также необходимой инструментальной поддержки модели принятия решений при оценивании риска.

1. **Целесообразность разработки модулей адаптации.**

Адаптация программного обеспечения

Программное обеспечение, способное к адаптации, обладает значительными конкурентными преимуществами по следующим двум причинам:

* *Во-первых,* способное к адаптации ПО решает более широкий круг задач в интересах конечных пользователей.
* *Во-вторых*, вокруг адаптируемого приложения возникает сообщество людей (занимающихся адаптацией), которое заинтересовано в увеличении числа конечных пользователей и повсеместном распространении приложения.
* Одной из причин, длительное время затруднявших создание адаптируемых приложений, было отсутствие у разработчиков согла-шения о том, в каком виде должны предоставляться программные интерфейсы, требуемые для адаптации.
* В настоящее время эта проблема для ОС Windows снята в связи с распространением технологии COM.

Технология COM

Технология COM (Component Object Model —модель компонентного объекта) представляет собой технологический стандарт от компании Microsoft, предназначенный для создания ПО на основе взаимодействующих компонентов объекта, каждый из которых может использоваться во многих программах одновременно.

Состав COM-объекта

В COM-технологии различаются следующие строительные блоки, используемые для создания объектов:

* Interface (COM-интерфейс) - множество прототипов функций (методов), чисто определенных. Термин «чисто определенный метод» или «абстрактный метод» исходит теории объектно-ориентированного анализа, и означает, что в определении класса отсутствует реализация метода, а присутствует только его определение. От такого класса нельзя создавать объекты. Его предназначение – описать фундаментальные общности для всех производных классов;
* COM object (COM-объект) – объект класса CoClass, который содержит реализацию COM интерфейса;

Интерфейсы

Интерфейсы являются основными строительными единицами COM.

Они объединяются на семантически связанные группы подпрограмм, через которые COM-объекты осуществляют взаимодействие

Свойства COM-объектов

COM-объект – это объект CoClass, который является классом, реализующим один или более интерфейсов.

COM-объект предоставляет функции, которые доступны через указатель на один из его интерфейсов.

В связи с этим, COM-объект обладает следующими особенностями:

- COM-объект защищен от прямого изменения внешними программами в своих данных, т.к. доступ к COM-объекту возможен только через указатель на интерфейс;

- COM-объект может быть использован приложениями, реализованными практически на любых современных средствах разработки приложений (алгоритмических языках), с любой внутренней организацией приложения, главное, чтобы средство разработки позволяло работать с указателями на структуры, на массивы и на функции.

COM-серверы

* Объект COM-класса должен иметь в своем составе фабрику классов, и идентификатор класса CLSID (Class Identifier), так чтобы COM-объект мог быть создан на основе существующего модуля.
* COM-сервер – это приложение, или библиотека, предоставляющее определенный набор сервисных функций для клиентских приложений или библиотек.

Local server (локальный сервер) – программный EXE модуль, работающий в отдельном адресном пространстве;

Remote server (удаленный сервер) – программный EXE модуль, работающий на удаленной машине:

Диспетчерский интерфейс

Диспетчерский интерфейс (dispatch interface) – стандартная специальная реализация интерфейса IDispatch, которую предлагает COM.

Диспетчерский интерфейс хранит внутри себя таблицу диспетчерских идентификаторов (dispID), каждый из которых является уникальным идентификатором члена интерфейса, и таблица, по сути, реализует отображение соответствующего dispID на имя каждого члена интерфейса.

Пользовательские интерфейсы

Vtable-интерфейсы (vtable interfaces) или пользовательские интерфейсы - определяются пользователем, и допускают вызывать методы интерфейса, пользуясь ссылками из vtable, при условии, если известен порядок записи ссылок на методы, число и тип передаваемых аргументов.

Двойные интерфейсы

Несмотря на то, что COM предоставляет возможность обращения к ресурсам серверов используя vtable-интерфейсы, что повышает скорость взаимодействия клиента и сервера, некоторые клиенты могут быть разработаны таким образом, что обращаются к объектам только через интерфейс диспетчеризации.

Таким образом, COM предлагает то, что называется двойственным, или двойным интерфейсом (dual interface). Двойные интерфейсы предлагают два пути для доступа к ресурсам сервера: через диспетчерский интерфейс и через vtable-интерфейс.

Преимущества использования двойных интерфейсов

* Двойные интерфейсы предлагают возможность получения указателей на ресурсы сервера по их именам при компиляции объекта, таким образом, позволяя создавать клиентов, с привязкой к vtable на этапе компиляции;
* Двойные интерфейсы позволяют клиентам осуществить прямой доступ к ресурсам сервера через vtable-интерфейсы, что увеличивает скорость взаимодействия объектов;
* Двойные интерфейсы имею преимущества проверки соответствия типов на этапе компиляции (преимущества раннего связывания);
* Клиенты, не работающие напрямую с vtable-интерфейсами имеют возможность взаимодействовать с объектами через диспетчерские интерфейсы;

Расширения COM

Одним из расширением технологии COM является OLE, представляющая собой библиотеку собственных интерфейсов, типов данных и подпрограмм, предназначенных для обеспечения функциональности OLE.

Automation

Технология автоматизации (automation) предлагает возможность программного управления одного приложения другим. В данной технологии различаются две составные компоненты:

* Клиентская часть, называемая контроллером автоматизации (automation controller);
* Серверная часть, которая носит название объекта автоматизации (automation object) – объект, которым управляет клиент.

Средства разработки COM-приложений

Основным инструментом разработки COM-приложений, что закономерно, являются продукты Microsoft, относящиеся к семейству визуальных средств программирования Visual Studio. Все компоненты этого семейства предлагают средства работы по технологии COM, и направлены в основном именно на разработку продуктов в рамках этой технологии.

1. **Вредоносные программы: классификация, методы обнаружения.**

Компьютерный вирус — это программа, способная создавать свои копии (не обязательно совпадающие с оригиналом) и внедрять их в файлы и системные области компьютера.

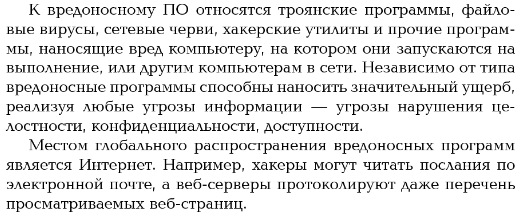
Основная черта – способность распространяться при запуске!

Вредоносные программы — это программы, предназначенные для незаконного доступа к информации, для скрытого использования компьютера или для нарушения работы компьютера и компьютерных сетей.

Последствия внедрения вирусов

* Снижение производительности ПК;
* Беспорядочная смена пользовательских настроек;
* Появление новых сомнительных панелей инструментов(аддонов);
* Вирусы могут глубоко внедряться в  сложные механизмы работы ОС так, чтобы в значительной степени осложнить их обнаружение и уничтожение;
* Вирус способен уничтожить всю ОС.
* Развитие получили и вирусные технологии для мобильных устройств. В качестве пути проникновения используются не только Bluetooth-устройства, но и обычные MMS-сообщения (червь ComWar)

Виды вредоносных программ



Троян, или троянская программа (троян, троянец, троянский конь, трой), — это вирус, проникающий на компьютер под видом безвредной программы.

Он не имеет собственного механизма распространения, и этим отличается от вирусов, которые распространяются, прикрепляя себя к обычной программе, и от «червей», которые копируют себя по сети.

Если же троян несет вирусное тело, то он становится очагом «заразы»

Трояны крайне просты в написании: простейшие из них состоят из нескольких десятков строк кода языка C++.

Троян, запущенный на ПЭВМ, может мешать работе пользователя, шпионить за ним, использовать ресурсы компьютера для целей запустившего его злоумышленника (хакера).

Присваивая трояну служебное имя, хакер маскирует его под другую программу, например программу установки другой программы.

Чтобы успешно выполнять эти функции, троян может имитировать или полноценно заменить приложение и (или) файл, под который он маскируется.

Если антивирус сообщает, что при обнаружении трояна он не может удалить его, то можно попробовать выполнить загрузку ОС с альтернативного источника и повторить проверку антивирусом.

Если троян обнаружен в системе, то его можно также удалить вручную, но уже в «безопасном режиме»

Шпион — вирус, скрытно устанавливающийся на ПЭВМ в целях полного или частичного контроля за работой компьютера и пользователя без согласия последнего.

Шпионы способны:

* собирать информацию о наиболее часто посещаемых сайтах;
* запоминать нажатия клавиш на клавиатуре,
* записывать скриншоты экрана и отправлять информацию хакерам;
* несанкционированно и удаленно управлять компьютером;
* инсталлировать на компьютер пользователя дополнительные программы;
* сканировать порты, пароли и др.;
* изменять параметры ОС (руткиты, перехватчики управления);
* перенаправлять активность браузеров, что влечет за собой посещение веб-сайтов вслепую с риском заражения вирусами.
* Шпионы обычно не саморазмножаются, а внедряются в ПЭВМ преимущественно с коммерческими целями. Типичные проявления шпионов — демонстрация рекламных всплывающих окон, кража персональной информации и др.

До настоящего времени борьба со шпионами остается дорогостоящей проблемой: если значительное их число проникло в ОС, единственным средством борьбы с ними остается полная переустановка ОС

Сетевой червь — разновидность самовоспроизводящихся вирусов — программ, распространяющихся в локальных и глобальных компьютерных сетях.

Черви являются самостоятельными программами, которые могут использовать различные механизмы распространения.

Одни — требуют определенного действия пользователя для распространения, например, открытия инфицированного сообщения в клиенте электронной почты.

Другие — могут распространяться автономно, выбирая и атакуя компьютеры в полностью автоматическом режиме. Третьи — это эксплойты, маскируются под различные ОС.

Выделяют черви, которые могут инфицировать работающую программу и находиться в оперативном запоминающем устройстве (ОЗУ), не затрагивая жесткие диски. От них можно избавиться перезапуском компьютера.

Существуют черви, которые после «успешного» инфицирования памяти сохраняют код в  реестре Windows на жестком диске.

От таких червей можно избавиться только с помощью антивируса

Руткит — вирусная программа или набор программ, использующих технологии сокрытия системных объектов (файлов, процессов, драйверов и др.) посредством обхода механизмов системы.

Руткит позволяет хакеру закрепиться во взломанной системе и скрыть следы своей деятельности.

В системе Windows под термином «руткит» принято понимать программу, которая внедряется в систему и перехватывает системные функции или производит замену системных библиотек.

Различают руткиты, работающие в режиме пользователя (user-mode), и работающие в режиме ядра (kernel-mode).

Первая категория основана на перехвате функций библиотек пользовательского режима, вторая — на установке в систему драйвера, осуществляющего перехват функций уровня ядра.

Признаки заражения

Можно отметить следующие основные признаки заражения ПЭВМ вирусами:

* вывод на экран непредусмотренных сообщений или изображений;
* подача непредусмотренных звуковых сигналов;
* неожиданное открытие и закрытие лотка CD-ROM-устройства;
* самопроизвольный запуск на компьютере каких-либо программ;
* при наличии на ПЭВМ межсетевого экрана, появление предупреждений о попытке программы выйти в Интернет, хотя вы это никак не инициировали;
* друзьям или знакомым идут от вас сообщения, которые вы не отправляли;
* наличие в почте массы сообщений без обратного адреса и заголовка.

Действия при обнаружении вируса

* отключить компьютер от локальной сети;
* установить (если не установлен) антивирус;
* получить последние обновления антивирусных баз;
* запустить полную проверку компьютера

Методы защиты от вредоносных программ

Чтобы снизить риск потерь от воздействия вредоносных программ, рекомендуется:

* использовать современные ОС;
* включить режим автоматического обновления ОС;
* постоянно работать на ПЭВМ исключительно под правами пользователя;
* использовать антивирусы известных производителей с автоматическим обновлением сигнатурных баз;
* использовать персональный Firewall, контролирующий выход в Интернет с персонального компьютера на основании политик, которые устанавливает сам пользователь
* ограничить физический доступ к компьютеру посторонних лиц;
* использовать внешние носители информации от проверенных источников;
* не открывать компьютерные файлы, полученные от ненадежных источников;

1. **Антивирусные программы: классификация, сравнительный анализ.**

Антивирусные сканеры

Самым популярными и эффективными антивирусными программами являются антивирусные сканеры, принцип работы которых основан на проверке файлов, секторов и системной памяти в целях поиска в них известных и новых вирусов.

Для поиска известных вирусов используются так называемые маски.

Маской вирусаявляется некоторая последовательность кода, специфичная для этого конкретного вируса.

Сканеры подразделяются на резидентные (мониторы), производящие сканирование «на лету», и нерезидентные, обеспечивающие проверку системы только по запросу.

Резидентные сканеры обеспечивают более надежную защиту системы, поскольку они немедленно реагируют на появление вируса, в то время как нерезидентный сканер способен опознать вирус только во время своего очередного запуска.

К достоинствамсканеров относиться их универсальность, к недостаткам – размеры антивирусных баз, которые сканерам приходится хранить и пополнять, и относительно небольшая скорость поиска вирусов

Принцип работы CRC-сканеров

Принцип работы CRC-сканеров основан на подсчете CRC-сумм (контрольных сумм) для присутствующих на диске файлов/системных секторов.

Эти CRC-суммы затем сохраняются в базе данных антивируса, и некоторая другая информация: длины файлов, даты их последней модификации и т. д.

При последующем запуске CRC-сканеры сверяют данные, содержащиеся в базе данных, с реально подсчитанными значениями.

Если информация о файле, записанная в базе данных, не совпадает с реальными значениями, то CRC-сканеры сигнализируют о том, что файл был изменен или заражен вирусом.

Характерный недостатокэтих антивирусов заключается в невозможности обнаружения вируса с момента его появления и до тех пор, пока не будут произведены изменения на компьютере.

CRC-сканеры не могут определить вирус в новых файлах (в электронной почте, на дискетах, в восстанавливаемых файлах или при распаковке файлов из архива), поскольку в их базах данных отсутствует информация об этих файлах.

Программы-детекторы

Программы-детекторы осуществляют поиск характерной для конкретного вируса сигнатуры в оперативной памяти и в файлах, и при обнаружении выдают соответствующее сообщение.

Недостатком таких антивирусных программ является то, что они могут находить только те вирусы, которые известны разработчикам таких программ.

Антивирусные блокировщики

Антивирусные блокировщики **—** это резидентные программы, перехватывающие вирусоопасные ситуации и сообщающие об этом пользователю.

К вирусоопасным относятся вызовы на открытие для записи в выполняемые файлы, запись в загрузочный сектор диска, которые характерны для вирусов в моменты их размножения.

К достоинствам блокировщиковотносится их способность обнаруживать и блокировать вирус на самой ранней стадии его размножения, что, кстати, бывает очень полезно в случаях, когда давно известный вирус постоянно активизируется.

Программы-доктора

Программы-доктора, или фаги, не только находят зараженные вирусами файлы, но и «лечат» их, т. е. удаляют из файла тело программы-вируса, возвращая файлы в исходное состояние.

В начале своей работы фаги ищут вирусы в оперативной памяти, уничтожая их, и только затем переходят к «лечению» файлов.

Среди фагов выделяют полифаги, т. е. программы-доктора, предназначенные для поиска и уничтожения большого числа вирусов.

Наиболее известные из них Norton AntiVirus и Doctor Web.

Учитывая, что постоянно появляются новые вирусы, программы-детекторы и программы-доктора быстро устаревают, и требуется регулярное обновление версий.

Программы-ревизоры

Программы-ревизоры относятся к самым надежным средствам защиты от вирусов.

Ревизоры запоминают исходное состояние программ, каталогов и системных областей диска тогда, когда компьютер не заражен вирусом, а затем периодически или по желанию пользователя сравнивают текущее состояние с исходным.

Обнаруженные изменения выводятся на экран монитора. Как правило, сравнение состояний производят сразу после загрузки ОС. К числу программ - ревизоров относится широко распространенная в России программа Adinf

Программы-фильтры

Программы-фильтры, или сторожа, представляют собой небольшие резидентные программы, предназначенные для обнаружения cktle.ob —подозрительных действийпри работе компьютера, характерных для вирусов:

* попытки коррекции файлов с расширениями COM, EXE;
* изменение атрибутов файла;
* прямая запись на диск по абсолютному адресу;
* запись в загрузочные секторы диска;
* загрузка резидентной программы

Антивирусные программы

* ESET NOD32 — отличается высокой скоростью сканирования ресурсов компьютера в специальном режиме, не мешая остальным запущенным процессам;
* Kaspersky Total Security — является кроссплатформенным антивирусом и применяется для защиты сразу нескольких синхронизированных устройств.
* DR.Web **—** мощный антивирус, обладающий тем недостатком, что запущенное им сканирование локальных носителей заметно снижает быстродействие персонального компьютера;

1. **Файрвол: задачи, сравнительный анализ, настройка.**

Что такое Firewall

Межсетевой экран или сетевой экран — комплекс аппаратных или программных средств, осуществляющий контроль и фильтрацию проходящих через него сетевых пакетов в соответствии с заданными правилами.

Фаерволом может быть как программное средство, так и комплекс ПО и оборудования.

Основная задача Firewall

* Основной задачей сетевого экрана является защита компьютерных сетей или отдельных узлов от несанкционированного доступа. Также сетевые экраны часто называют фильтрами, так как их основная задача — не пропускать (фильтровать) пакеты, не подходящие под критерии, определённые в конфигурации.
* При выборе межсетевого экрана, необходимо руководствоваться главным критерием – его надежностью.

Как работает Firewall

Firewall контролирует порты и проходящие через них пакеты. Важной функцией Firewall является наблюдение за всеми установленными и запущенными приложениями. У брандмауэра есть набор правил, как встроенных по умолчанию, так и задаваемых самим пользователем.

Фаервол, или брандмауэр, — термин для обозначения технических и программных средств, которые обрабатывают входящий и исходящий сетевой трафик.

Разновидности сетевых экранов

Сетевые экраны подразделяются на различные типы в зависимости от следующих характеристик:

* обеспечивает ли экран соединение между одним узлом и сетью или между двумя или более различными сетями;
* на уровне каких сетевых протоколов происходит контроль потока данных;
* отслеживаются ли состояния активных соединений или нет.

Виды Firewall в зависимости от охвата контролируемых потоков

* традиционный сетевой (или межсетевой) экран - программа (или неотъемлемая часть операционной системы) на шлюзе (сервере передающем трафик между сетями) или аппаратное решение, контролирующие входящие и исходящие потоки данных между подключенными сетями.
* персональный сетевой экран — программа, установленная на пользовательском компьютере и предназначенная для защиты от несанкционированного доступа только этого компьютера.
* вырожденный случай — использование традиционного сетевого экрана сервером, для ограничения доступа к собственным ресурсам

Виды Firewall в зависимости от уровня, на котором происходит контроль доступа

* сетевой уровень - когда фильтрация происходит на основе адресов отправителя и получателя пакетов, номеров портов транспортного уровня модели OSI и статических правил, заданных администратором;
* сеансовый уровень (также известные как stateful) отслеживающие сеансы между приложениями, не пропускающие пакеты нарушающих спецификации TCP/IP, часто используемых в злонамеренных операциях — сканировании ресурсов, взломах через неправильные реализации TCP/IP, обрыв/замедление соединений, инъекция данных.
* уровень приложений - фильтрация на основании анализа данных приложения, передаваемых внутри пакета. Такие типы экранов позволяют блокировать передачу нежелательной и потенциально опасной информации, на основании политик и настроек.

Виды Firewall в зависимости от отслеживания активных соединений

* stateless (простая фильтрация) которые не отслеживают текущие соединения (например, TCP), а фильтруют поток данных исключительно на основе статических правил;
* stateful, stateful packet inspection (SPI) (фильтрация с учётом контекста) с отслеживанием текущих соединений и пропуском только таких пакетов, которые удовлетворяют логике и алгоритмам работы соответствующих протоколов и приложений. Такие типы сетевых экранов позволяют эффективнее бороться с различными видами DoS-атак и уязвимостями некоторых сетевых протоколов.

Типичные возможности

* фильтрация доступа к заведомо незащищенным службам;
* препятствование получению закрытой информации из защищенной подсети, а также внедрению в защищенную подсеть ложных данных с помощью уязвимых служб;
* контроль доступа к узлам сети;
* может регистрировать все попытки доступа как извне, так и из внутренней сети, что позволяет вести учёт использования доступа в Интернет отдельными узлами сети;
* регламентирование порядка доступа к сети;
* уведомление о подозрительной деятельности, попытках зондирования или атаки на узлы сети или сам экран;

Ashampoo FireWall FREE

* Ashampoo FireWall FREE – бесплатный брандмауэр, перед установкой которого необходимо убедится, что в системе не установлена антивирусная программ или какой-то другой межсетевой экран.
* После запуска приложение переходит в режим обучения, при котором отслеживает любую активность. Обнаружив, что постороннее приложение пытается установить сетевое соединение Ashampoo FireWall запрашивает пользователя разрешение на блокировку приложения, после чего запоминает ответ и в дальнейшем следует выбранному алгоритму действий.
* Утилита Ashampoo FireWall служит для блокировки IP-спама - разновидности назойливых всплывающих административных со общений от внутренних служб Windows.
* Для их блокировки модуль отключает службу сообщений.

1. **Групповые политики. Аутентификация. Учетные записи.**

Групповая политика

Групповая политика– это один из главных инструментов для настройки Windows, с помощью которого можно как задействовать множество функций системы, так и отключить их.

С помощью групповых политик можно настроить систему для пользователя, под которым произведен вход в систему, для других пользователей данного компьютера или других пользователей на других компьютерах в доменной сети.

Групповые политики доступны только в профессиональных, максимальных и корпоративных версиях Windows, у пользователей домашней версии Windows данный набор инструментов отключен.

Централизованные групповые политики

Во множестве организаций используют централизованное управление групповыми политиками.

В закрытой сети на контроллере домена в Active Directory можно задавать различные доступы, закрывать или открывать различные ресурсы. Также возможно задать групповые политики для всех персональных компьютеров в сети:

* установить открытие определенных сайтов в качестве домашней страницы
* запретить доступ к различным настройкам панели управления
* закрыть доступ к определенным папкам

Системный администратор может менять настройки компьютера пользователя с помощью групповых политик на сервере.

Локальные групповые политики

Групповые политики используются не только системными администраторами, их можно использовать для настройки своего же компьютера.

Например, с помощью групповых политик можно изменить экран входа в Windows, также можно ограничить доступ к некоторым папкам другим пользователям данного компьютера, можно управлять паролями, отображением папок, отключить автоматическое обновление, разрешать запуск только определенных приложений, отключить доступ к  панели управления и др.

Работа с командной строкой

Можно добавить через консоль групповые политики и настроить систему для других пользователей данного компьютера.

1. В строке поиска окна Выполнить, открываемого клавишами [Win  + R], ввести команду gpedit.msc и  нажать клавишу [Enter]

2. В открывшемся редакторе локальной групповой политики найти нужные параметры, отключить или включить их.

*Например, можно отключить автозапуск съемных носителей, чтобы при подключении съемного носителя не появлялось окно с вопросом, что делать с этим устройством*.

3. В левой колонке открыть

Конфигурация компьютера / Административные шаблоны/ Компоненты Windows / Политики автозапуска.

С правой стороны выбрать политику Отключить автозапуск

4. Поставить точку возле Включить / ОК

5. После каждого изменения групповых политик перезагрузить компьютер, чтобы изменения вступили в силу.

Аутентификация

Аутентификация **—** проверка подлинности пользователя путем сравнения введенного им пароля с паролем, сохраненным в базе данных пользователей.

Аутентификация пользователя включает сервер безопасности HTTP FireWall-1, который предоставляет механизм реализации указанной проверки.

Сервер FireWall обеспечивает три вида аутентификации:

1) аутентификация пользователя, которая позволяет администратору давать каждому пользователю свои привилегии доступа;

2) аутентификация клиентов, которая дает механизм для аутентификации пользователя любого приложения — стандартного — или собственной разработки;

3) аутентификация сессий, дающая прозрачную аутентификацию каждой сессии, что может быть интегрировано с любым приложением

Учетная запись

Учетная запись — это данные о пользователе, хранящиеся в компьютерной системе. Она нужна для верификации человека, предоставления ему доступа к личным настройкам и данным.

Для входа в систему с помощью учетной записи необходимо ввести логин и пароль. В отдельных случаях может понадобиться дополнительная информация.

Существует три вида учетных записей: обычные, администраторские и гостевые**.**

Обычные учетные записипредназначены для ежедневной работы пользователей.

Администраторские учетныезаписи дают полный контроль над компьютером. Это означает, что такая учетная запись позволяет владельцу изменять любые данные, принадлежащие системе, менять параметры безопасности и устанавливать программы на других компьютерах. Администраторские учетные записи обычно применяются в офисах.

Гостевые учетные записи, как правило, защищены стандартным логином и паролем и служат для обеспечения временного доступа (например, для стажеров или случайных пользователей)

1. **Тестирование защиты программного обеспечения.**

Тестирование – процесс проверки соответствия заявленных к продукту требований и реально реализованной функциональности, осуществляемый путем наблюдения за его работой в искусственно созданных ситуациях и на ограниченном наборе тестов, выбранных определенным образом.

Тестирование по требованиям безопасности - процесс выявления наличия или отсутствия уязвимостей в продукте в искусственно созданных ситуациях и на ограниченном наборе тестов, выбранных определенным образом.

Тестирование защищенности

Тестирование защищенности: Тестирование с целью оценить защищенность программного продукта

Объекты тестирования:

* пароли
* шифрование
* аппаратные устройства доступа
* уровни доступа к информации
* авторизация
* скрытые каналы
* безопасность на физическом уровне

Приемы выявления уязвимостей

* + Ручной (экспертный анализ)
  + Статически анализ безопасности (по шаблону)
  + Динамический анализ безопасности

Ручной (экспертный анализ)

* При ручном подходе выявления уязвимостей применяется экспертный анализ, т.е. специалист, который проводит данное исследование, полагается на свои знания и опыт.
* Данный подход не подразумевает использования, каких либо автоматизированных средств.
* Данный прием имеет большие затраты по времени и предполагает наличие специалистов высокой квалификации.
* Данный подход считается самым эффективным с точки зрения точности и полноты покрытия проверок.

Статически анализ безопасности (по шаблону)

* Прием выявления уязвимостей «по шаблону» подразумевает использование материалов и наработок полученных исходя из опыта работы в данной области.
* При подходе выявления уязвимостей «по шаблону» часто применяется автоматизированный подход поиска уязвимостей по заданным шаблонам (спискам потенциально опасных сигнатур).
* Часто при данном подходе используется совмещение методов автоматизированного и ручного поиска уязвимостей.

Динамический анализ безопасности

* Динамический анализ является обязательным подходом при выявлении уязвимостей.
* Он позволяет проводить тестирование при непосредственном выполнении программного изделия.
* В процессе динамического тестирования программного комплекса реализуется составленный список тестов, направленный на достижение, либо провал нарушения функций безопасности продукта.
* Т.е. определяется возможность или невозможность эксплуатировать найденную потенциально опасную сигнатуру в рамках работающего программного продукта, при заданном тестовом окружении.

Традиционные методы и подходы тестирования

Уровни тестирования:

* Модульное тестирование (Unit testing)
  + Этот уровень тестирования позволяет проверить функционирование отдельно взятого элемента системы. Что считать элементом – модулем системы определяется контекстом.
* Интеграционное тестирование (Integration testing)
  + Данный уровень тестирования является процессом проверки взаимодействия между программными компонентами/модулями.
  + Классические стратегии интеграционного тестирования – “сверху-вниз” и “снизу-вверх” – используются для традиционных, иерархически структурированных систем и их сложно применять, например, к тестированию слабосвязанных систем, построенных в сервисно-ориентированных архитектурах (SOA).
* Системное тестирование (System testing)
  + Системное тестирование охватывает целиком всю систему.
  + Большинство функциональных сбоев должно быть идентифицировано еще на уровне модульных и интеграционных тестов.
  + В свою очередь, системное тестирование, обычно фокусируется на нефункциональных требованиях – безопасности, производительности, точности, надежности т.п.
  + На этом уровне также тестируются интерфейсы к внешним приложениям, аппаратному обеспечению, операционной среде и т.д.

Виды тестирования:

* Приёмочное тестирование (Acceptance/qualification testing)
  + Проверяет поведение системы на предмет удовлетворения требований заказчика.
* Установочное тестирование (Installation testing)
  + Из названия следует, что данные тесты проводятся с целью проверки процедуры инсталляции системы в целевом окружении.
* Альфа- и бета-тестирование (Alpha and beta testing)
  + Перед тем, как выпускается программное обеспечение, как минимум, оно должно проходить стадии альфа (внутреннее пробное использование) и бета (пробное использование с привлечением отобранных внешних пользователей) версий.
  + Отчеты об ошибках, поступающие от пользователей этих версий продукта, обрабатываются в соответствии с определенными процедурами, включающими подтверждающие тесты (любого уровня), проводимые специалистами группы разработки.
* Функциональные тесты/тесты соответствия (Conformance testing/Functional testing/Correctness testing)
  + Эти тесты могут называться по разному, однако, их суть проста – проверка соответствия системы, предъявляемым к ней требованиям, описанным на уровне спецификации поведенческих характеристик.
* Достижение и оценка надежности (Reliability achievement and evaluation)
  + Помогая идентифицировать причины сбоев, тестирование подразумевает и повышение надежности программных систем.
  + Случайно генерируемые сценарии тестирования могут применяться для статистической оценки надежности.
* Регрессионное тестирование (Regression testing)
  + Определение успешности регрессионных тестов (IEEE 610–90 “Standard Glossary of Software Engineering Terminology”) гласит: “повторное выборочное тестирование системы или компонент для проверки сделанных модификаций не должно приводить к непредусмотренным эффектам”.
  + На практике это означает, что если система успешно проходила тесты до внесения модификаций, она должна их проходит и после внесения таковых.
* Тестирование производительности (Performance testing)
  + Специализированные тесты проверки удовлетворения специфических требований, предъявляемых к параметрам производительности.
  + Существует особый подвид таких тестов, когда делается попытка достижения количественных пределов, обусловленных характеристиками самой системы и ее операционного окружения.
* Нагрузочное тестирование (Stress testing)
  + Необходимо понимать отличия между рассмотренным выше тестированием производительности с целью достижения ее реальных (достижимых) возможностей производительности и выполнением программной системы c повышением нагрузки, вплоть до достижения запланированных характеристик и далее, с отслеживанием поведения на всем протяжении повышения загрузки системы.
* Сравнительное тестирование (Back-to-back testing)
  + Единичный набор тестов, позволяющих сравнить две версии системы.
* Восстановительные тесты (Recovery testing)
  + Цель – проверка возможностей рестарта системы в случае непредусмотренной катастрофы (disaster), влияющей на функционирование операционной среды, в которой выполняется система.
* Конфигурационное тестирование (Configuration testing)
  + В случаях, если программное обеспечение создается для использования различными пользователями (в терминах “ролей”), данный вид тестирования направлен на проверку поведения и работоспособности системы в различных конфигурациях.
* Тестирование удобства и простоты использования (Usability testing)
  + Цель – проверить, насколько легко конечный пользователь системы может ее освоить, включая не только функциональную составляющую – саму систему, но и ее документацию; насколько эффективно пользователь может выполнять задачи, автоматизация которых осуществляется с использованием данной системы; наконец, насколько хорошо система застрахована (с точки зрения потенциальных сбоев) от ошибок пользователя.

1. **Средства и протоколы шифрования сообщений.**

Шифрование

* Шифрование – обратимое преобразование информации в целях скрытия от неавторизованных лиц, с предоставлением в это же время авторизованным пользователям доступа к ней.
* Главным образом, шифрование служит задачей соблюдения конфиденциальности передаваемой информации.
* Важной особенностью алгоритма шифрования является использование ключа, который утверждает выбор конкретного преобразования из совокупности возможных для данного алгоритма.

Криптография

* Криптография– наука о защите информации от прочтения её посторонними.
* Защита достигается шифрованием, т.е. преобразованием, которое делает защищенные данные трудно раскрываемыми по входным данным без знания специальной ключевой информации – ключа.
* Под ключом понимается легко изменяемая часть криптосистемы, хранящаяся в тайне и определяющая какое шифрующее преобразование из возможных выполняется в данном случае.

Криптосистема

* Криптосистема – семейство выбираемых с помощью ключа обратимых преобразований, которые преобразуют защищаемый открытый текст в шифрограмму, и обратно.
* К средствам криптографической защиты информации относятся аппаратные, программно-аппаратные средства и ПО, реализующие криптографические алгоритмы преобразования информации.

Криптографические методы защиты информации

* Криптографические методы защиты информации — это специальные методы шифрования, кодирования или иного преобразования информации, в результате которого ее содержание становится недоступным без предъявления ключа криптограммы и обратного преобразования.
* Криптографический метод защиты, безусловно, самый надежный, так как охраняется непосредственно сама информация, а не доступ к ней (например, зашифрованный файл нельзя прочесть даже в случае кражи носителя).
* Данный метод защиты реализуется в виде программ или пакетов программ.

Все многообразие существующих криптографических методов в симметричных криптосистемах можно свести к следующим  
классам преобразований:

*-*подстановка *—* символы шифруемого текста заменяются символами того же или другого алфавита в соответствии с заранее определенным правилом;

*-*перестановка *—* символы шифруемого текста переставляются по некоторому правилу в пределах заданного блока передаваемого текста;

*-*аналитическое преобразование — шифруемый текст преобразуется по некоторому аналитическому правилу, например, гаммирование заключается в наложении на исходный текст некоторой псевдослучайной последовательности, генерируемой на основе ключа;

*-*комбинированное преобразование — представляет собой последовательность (с возможным повторением и чередованием) основных методов преобразования, применяемую к блоку (части) шифруемого текста

Шифрование с закрытым ключом

* *Шифрование с закрытым ключом* (симметричное шифрование) осуществляет преобразование данных для предотвращения их просмотра третьей стороной.
* В данном способе для шифрования и расшифровки данных используется общий закрытый ключ.
* При шифровании с закрытым ключом для дешифровки данных используется один закрытый ключ. *Необходимо обезопасить этот ключ от несанкционированного доступа*, потому что любое обладающее им лицо может использовать его для расшифровки данных или шифрования собственных данных с подменой источника.

Криптографический протокол

* *Криптографический протокол* — это такая процедура взаимодействия двух или более абонентов с использованием криптографических средств, в результате которой абоненты достигают своей цели, а их противники — не достигают.
* В основе протокола лежит набор правил, регламентирующих использование криптографических преобразований и алгоритмов в информационных процессах.

Свойства криптографического протокола

* при выполнении протокола важен порядок действий; каждое действие должно выполняться в свою очередь и только по окончании предыдущего;
* протокол должен быть непротиворечивым;
* протокол должен быть полным, т. е. для каждой возможной ситуации должно быть предусмотрено соответствующее действие.

Типы криптографического протокола

* Типы протоколов можно условно подразделить на две группы: *прикладные протоколы и примитивные*.
* *Прикладной протокол* решает конкретную задачу, которая возникает (или может возникнуть) на практике.
* *Примитивные протоколы* используются как своеобразные «строительные блоки» при разработке прикладных протоколов.

Назначение видов протоколов

1. Протоколы конфиденциальной передачи сообщений.

Задача конфиденциальной передачи сообщений состоит в следующем. Имеются два участника протокола, которые являются абонентами сети связи. Участники соединены некоторой линией связи, по которой можно пересылать сообщения в обе стороны.

Линию связи может контролировать противник. У одного из абонентов имеется конфиденциальное сообщение m, и задача состоит в том, чтобы это сообщение конфиденциальным же образом передать второму абоненту.

2. Протоколы аутентификации и идентификации.

Они предназначены для предотвращения доступа к  некоторой информации лиц, не являющихся ее пользователями, а также предотвращения доступа пользователей к тем ресурсам, на которые у них нет полномочий.

Сфера применения — организация доступа пользователей к ресурсам некоторой большой информационной системы.

3. Протоколы распределения ключей. Необходимы для обеспечения секретными ключами участников обмена зашифрованными сообщениями.

4. Протоколы электронной цифровой подписи.

Позволяют ставить под электронными документами подпись, аналогичную обыкновенной подписи на бумажных документах. В результате выполнения протокола электронной цифровой подписи к передаваемой информации добавляется уникальное числовое дополнение, позволяющее проверить ее авторство.

5. Протоколы обеспечения неотслеживаемости(электронные деньги).

Под электронными деньгами в криптографии понимают электронные платежные средства, обеспечивающие неотслеживаемость, т. е. невозможность проследить источник пересылки информации

Этот протокол достаточно прост, однако он может действительно использоваться на практике.

В современной криптографии большое внимание уделяется разработке криптографических протоколов, т. е. процедур или алгоритмов взаимодействия абонентов с использованием криптографических средств.

В основе протокола лежит набор правил, регламентирующих использование криптографических преобразований.