

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

СЕВЕРО-КАВКАЗСКАЯ ГОСУДАРСТВЕННАЯ АКАДЕМИЯ

СРЕДНЕПРОФЕССИОНАЛЬНЫЙ КОЛЛЕДЖ

З. Н. Чагарова

ВНЕДРЕНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ

Практикум для обучающихся по специальности
09.02.07 Информационные системы и программирование

Черкесск
2024

УДК 004
ББК 16.33
Ч 12

Рассмотрено на заседании ЦК «Информационные дисциплины»
Протокол № 1 от «01» сентября 2023 г.
Рекомендовано к изданию редакционно-издательским советом СКГА
Протокол № 26 от «29» сентября 2023 г.

Рецензенты: Шовкарова З.С. – преподаватель СПК СКГА

Ч 12 **Чагарова, З.Н.** Внедрение информационных систем: практикум для обучающихся по специальности 09.02.07 «Информационные системы и программирование» / З.Н. Чагарова. – Черкесск: БИЦ СКГА, 2024. – 52 с.

Практикум по дисциплине Внедрение информационных систем определяют этапы выполнения работы на практическом занятии, содержат рекомендации по выполнению индивидуальных заданий и образцы решения задач.

УДК 004
ББК 16.33

© Чагарова З.Н., 2024
© ФГБОУ ВО СКГА», 2024

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
ПРАКТИЧЕСКАЯ РАБОТА № 1. «Разработка сценария внедрения информационной системы для рабочего места»	5
ПРАКТИЧЕСКАЯ РАБОТА № 2. «Разработка технического задания на внедрение информационной системы»	6
ПРАКТИЧЕСКАЯ РАБОТА № 3. «Разработка графика разработки и внедрения информационной системы»	10
ПРАКТИЧЕСКАЯ РАБОТА № 4. «Сравнительный анализ методологий проектирования»	22
ПРАКТИЧЕСКАЯ РАБОТА № 5. «Анализ бизнес-процессов подразделения»	28
ПРАКТИЧЕСКАЯ РАБОТА № 6. «Разработка и оформление предложений по расширению функциональности информационной системы»	31
ПРАКТИЧЕСКАЯ РАБОТА № 7. «Разработка перечня обучающей документации на информационную систему»	33
ПРАКТИЧЕСКАЯ РАБОТА № 8. «Разработка руководства оператора»	35
ПРАКТИЧЕСКАЯ РАБОТА № 9. «Разработка моделей интерфейсов пользователей»	38
ПРАКТИЧЕСКАЯ РАБОТА № 10. «Настройка доступа к сетевым устройствам»	40
ПРАКТИЧЕСКАЯ РАБОТА № 11. «Настройка политики безопасности»	45
Практическая работа № 12. «Выполнение задач тестирования в процессе внедрения»	50

ВВЕДЕНИЕ

Методические указания по учебной дисциплине **Внедрение информационных систем** разработаны с учетом ФГОС среднего профессионального образования специальности 09.02.07 Информационные системы и программирование, предназначены для студентов и преподавателей колледжа.

Методические указания определяют этапы выполнения работы на практическом занятии, содержат рекомендации по выполнению индивидуальных заданий и образцы решения задач.

В предлагаемом пособии на практическом примере рассмотрены особенности разработки внедрения информационных систем, разработки технического задания и руководства оператору, анализу бизнес-процессов, моделей интерфейсов и настройки устройств.

ПРАКТИЧЕСКАЯ РАБОТА № 1

Разработка сценария внедрения программного продукта для рабочего места

Цель: познакомиться с методом разработки программного обеспечения, научиться выполнять разработку программного продукта в команде.

Содержание работы

1. Рассмотрение теоретической части: групповая разработка программного обеспечения компьютерных систем

2. Выполнение заданий по организации команд разработки

Теоретическое обоснование

Существует две основные модели организации коллектива при разработке ПО:

1) Иерархическая модель определяет начальников и подчиненных.

2) Модель проектной группы, согласно которой в проектной группе предусматривается распределение обязанностей руководителя между членами коллектива. При этом за проект отвечает не один человек, а все члены группы — каждый за свой участок.

При разработке программного обеспечения выполняемые задачи распределяются:

по семи ролевым группам:

- 1) управление программой (program management);
- 2) архитектура продукта (architecture);
- 3) разработка (development);
- 4) тестирование (test);
- 5) управление выпуском (release operations);
- 6) удовлетворение потребителя (user experience);
- 7) управление продуктом (product management)

и шести ролям:

- 1) менеджер проекта (project manager) – ролевая группа Управление программой;
- 2) архитектор (architect) – ролевая группа Архитектура;
- 3) разработчик (developer) – ролевая группа Разработка;
- 4) тестер (tester) – ролевая группа Тестирование;
- 5) релиз-менеджер (release manager) – ролевая группа Управление выпуском;
- 6) бизнес-аналитик (business analyst) – ролевые группы Управление продуктом и Удовлетворение потребителя.

Методические указания

Для выполнения заданий вам потребуется вспомнить типологию программного обеспечения и основные понятия о групповой разработке программного обеспечения.

Третье и четвертое задания зафиксируйте в тетради.

Задание

1. Найдите в интернете информацию по «Принципы и методы коллективной работы»

2. Разбейтесь на группы по 2-4 человека и выберите один из вариантов для вашей группы

3. Обсудите в командах, какие работы предстоит выполнить, и распределите функции между членами группы в соответствии с выбранной моделью групповой разработки.

4. Обсудите в командах перечень работ каждого участника группы и составьте их список

Задание

1) Разработать автоматизированную информационную систему для банка

2) Разработать обучающую компьютерную игру

3) Разработать образовательный портал учебного заведения

Контрольные вопросы

1. Назовите типологию программного обеспечения

2. Дайте описательную характеристику групповой разработки программного обеспечения

3. Укажите положительные и отрицательные стороны групповой разработки

Форма отчёта

Конспект с распределенными функциями, устные ответы на контрольные вопросы.

ПРАКТИЧЕСКАЯ РАБОТА № 2

«Разработка технического задания на внедрение информационной системы»

Цель работы: получение практических навыков составления технического задания на внедрение информационной системы

Методика выполнения работы

Основные теоретические знания

Определение. *Техническое задание (ТЗ)* – это завершающий предпроектную стадию документ, который содержит цели и обоснование проектирования, а также определяет основные требования к ИС и исходные данные необходимые при разработке.

Техническое задание составляется по результатам предпроектного обследования объекта автоматизации. В настоящее время при составлении

технического задания обычно руководствуются требованиями следующих ГОСТов:

- 34.602-89 «Техническое задание на создание автоматизированной системы» – описывает состав и содержание ТЗ, которые распространяются на автоматизированную (информационную) систему в целом, в том числе:

- общесистемные требования к ИС;
- требования к компонентному составу ИС;
- требования к интеграции компонентов ИС между собой и с другими системами;

- требования к составу и содержанию работ по внедрению ИС;

- 19.201-78 «Техническое задание. Требования к содержанию и оформлению» – входит в единую систему программной документации и устанавливает порядок построения и оформления технического задания на программное изделие для ЭВМ.

При проектировании сложных систем обычно разрабатывают общее техническое задание на ИС в целом (в соответствии с требованиями ГОСТа 34.602-89), а также дополнительные технические задания на части системы:

- на создание информационно-вычислительной сети,
- на отдельные подсистемы ИС,
- на элементы программного обеспечения ИС – программные компоненты и/или комплексы (в соответствии с требованиями ГОСТа 19.201-78).

Для сравнительно небольших информационных систем допускается разработка единого технического задания, в котором устанавливаются как общие требования к ИС, так и требования к соответствующим программным средствам.

В соответствии с ГОСТом 19.201-78 техническое задание должно включать в себя следующие разделы:

- Введение;
- Основание для разработки;
- Назначение и цели разработки;
- Требования к программе и программной документации;
- Техничко-экономические показатели;
- Стадии и этапы разработки;
- Порядок контроля и приёмки;
- Приложение.

При составлении технического задания целесообразно учитывать следующие практические рекомендации:

1. Все изменения в структуре ТЗ (по сравнению с ГОСТ) должны быть обязательно согласованы с заказчиком.

2. При составлении ТЗ целесообразно использовать методику «дробления и детализации». Это значит, что структура документа (разбиение на разделы и подразделы) должна быть тщательно проработана, так чтобы

заинтересованное лицо могло быстро найти необходимые ему сведения относительно ИС по содержанию ТЗ.

3. Сведения о функциональном назначении должны давать ответ на вопрос «для чего?», а сведения об эксплуатационном назначении – на вопрос «для кого?». Описание функционального назначения разработки должно содержать общие сведения без указания отдельных деталей.

4. Требования к программе желательно составлять на основе ГОСТов и нормативно-технической документации заказчика.

5. При составлении требований к программе целесообразно использовать метод «шаблонного построения фраз», например

- При изложении требований к функциональным и иным характеристикам: «Программа должна обеспечивать возможность ...» или «Требования к ... не предъявляются».

- При изложении требований к квалификации персонала: «Каждый пользователь должен обладать практическими навыками работы с графическим пользовательским интерфейсом ОС»;

- и. т.п.

1. Требования к программным изделиям должны носить *императивный* характер. Если какие-либо требования (из перечисленных в ГОСТе) не предъявляются, об этом следует указывать специально.

2. Требования к пользовательскому интерфейсу рекомендуется оформлять в разделе «Специальные требования».

3. Минимальный состав программной документации, который должен упоминаться в соответствующем разделе ТЗ, включает в себя:

- Спецификация программной документации;
- Техническое задание;
- Программа и методики испытаний;
- Руководства администратора и оператора

1. В раздел «Технико-экономические показатели» можно включать оценку потребности в программном изделии и приблизительную оценку стоимости и трудоёмкости разработки.

2. Стадии и этапы разработки обычно излагаются в форме таблицы:

- Какие функции программного изделия подлежат испытанию;
- В какие сроки и чьими силами разрабатываются программные испытания;
- Срок проведения испытания;
- Оформление испытания;
- Иные условия (например, на какой технике проводятся испытания)

Задания к лабораторной работе

В соответствии с полученным вариантом задания разработать проект технического задания на программное изделие для ИС. Руководствоваться требованиями Единой системы программной документации (ЕСПД), в частности, ГОСТом 19.201-78.

Контрольные вопросы и задания

1. Понятие ТЭО и ТЗ
2. ГОСТ 34.602-89.
3. ГОСТ 19.201-78.
4. Основные разделы ТЗ на программу.
5. Методика «дробления и детализации».
6. Метод «шаблонного построения фраз».
7. Требования к программе.
8. Требования к программной документации.

Задание для выполнения.

Для работы выбрать информационную систему из списка.

1. Информационная система «Цветочный магазин»
2. Информационная система «Продуктовый магазин»
3. Информационная система «Аэропорт»
4. Информационная система «ЖД вокзал»
5. Информационная система «Грузоперевозки»
6. Информационная система «Свадебный салон»
7. Информационная система «Такси»
8. Информационная система «Поликлиника»
9. Информационная система «Отдел кадров»
10. Информационная система «Ресторан»
11. Информационная система «Библиотека»
12. Информационная система «Автосалон»
13. Информационная система «Мебельный салон»
14. Информационная система «Туристическое агентство»
15. Информационная система «Школа»
16. Информационная система «Тренажерный зал»
17. Информационная система «Аптека»
18. Информационная система «Строительная фирма»
19. Информационная система «Компьютерный салон»
20. Информационная система «Рекламное агентство»
21. Информационная система «Театр»
22. Информационная система «Зоомагазин»
23. Информационная система «Автосервис»
24. Информационная система «Дом Творчества»
25. Информационная система «Агрокомбинат»
26. Информационная система «Ювелирный магазин»
27. Информационная система «Салон красоты»
28. Информационная система «Прокат автомобилей»
29. Информационная система «Интернет-магазин техники»
30. Информационная система «Магазин по пошиву и продаже обуви»

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №3

разработка графика разработки и внедрения информационной системы

Цель работы: получение практических навыков разработки графика разработки и внедрения информационной системы.

Методика выполнения работы

Выполняем график разработки и внедрения информационной системы на примере Диаграммы Ганта с помощью MS Excel.

Основные теоретические знания

Что такое диаграмма Ганта?

Диаграмма Ганта названа в честь Генри Гантта, американского инженера-механика и консультанта по вопросам управления, который изобрел этот график еще в 1910-х годах. **Диаграмма Ганта в Excel** представляет собой проекты или задачи в виде каскадных горизонтальных гистограмм. **Диаграмма Ганта** иллюстрирует структуру разбивки проекта, отображая даты начала и окончания, а также различные отношения между действиями проекта и таким образом помогает отслеживать задачи по сравнению с запланированным временем или предопределенными вехами.

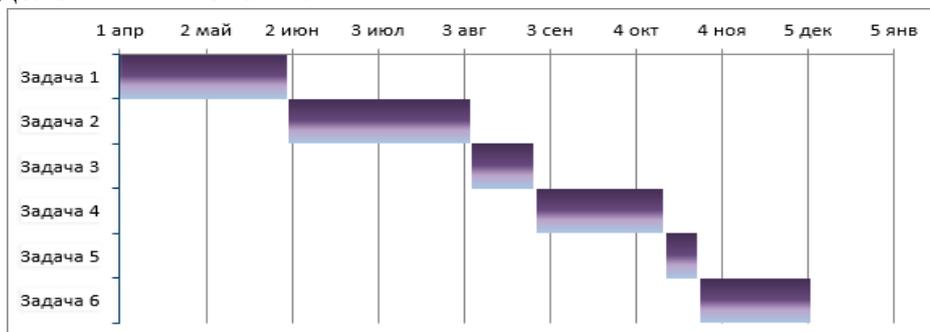


Диаграмма Ганта в Excel – Пример диаграммы Ганта.

Как сделать диаграмму Ганта в Excel 2016, 2013 и 2010

К сожалению, Microsoft Excel не имеет встроенного **шаблона диаграммы Ганта** в качестве опции. Тем не менее, вы можете быстро сделать **диаграмму Ганта в Excel**, используя функциональность гистограммы и добавив немного форматирования.

Далее мы по шагам рассмотрим, **как сделать диаграмму Ганта** за 3 минуты.

Мы будем использовать **Excel 2013** для создания **шаблона диаграммы Ганта**, но вы можете точно так же создать **диаграмму Ганта в Excel 2016** и **Excel 2010**.

Шаг 1. Создание таблицы проекта

Для начала необходимо ввести данные своего проекта в электронную таблицу Excel. Список каждой задачи представляет собой отдельную строку и структуру вашего плана проекта, включая *дату начала*, *дату окончания* и *длительность*, т. е. количество дней, необходимых для выполнения задачи.

	A	B	C	D
1	Название задачи	Дата начала	Дата окончания	Длительность, дни
2	Сбор требований	01.04.17	06.04.17	5
3	Дизайн интерфейса	07.04.17	18.04.17	11
4	Разработка	19.04.17	11.05.17	22
5	Модульное тестирование	12.05.17	18.05.17	6
6	Системное тестирование	19.05.17	30.05.17	11
7	Бета-тестирование	31.05.17	02.06.17	2
8	Исправление ошибок	03.06.17	17.06.17	14
9	Улучшения	18.06.17	02.07.17	14
10	Итоговое тестирование	03.07.17	09.07.17	6
11	Выпуск окончательной версии	10.07.17	12.07.17	2

Примечание. Для создания диаграммы Ганта в Excel нужны только столбцы «Дата начала» и «Длительность». Однако, если вы указываете даты окончания, вы можете использовать простую формулу для вычисления «Длительности»:

Диаграмма Ганта в Excel – Таблица проекта

Шаг 2. Создание стандартной диаграммы Excel на основе даты начала

Приступим к созданию диаграммы Ганта в Excel, настраивая обычную линейчатую диаграмму с накоплением.

- Выберите диапазон начальных дат с заголовком столбца, в нашем случае это B1:B11. Обязательно выберите только ячейки с данными, а не весь столбец.

- Перейдите на вкладку «ВСТАВКА» --> группа «Диаграммы» и нажмите «Вставить линейчатую диаграмму».

- В разделе «Линейчатая» щелкните «Линейчатая с накоплением».

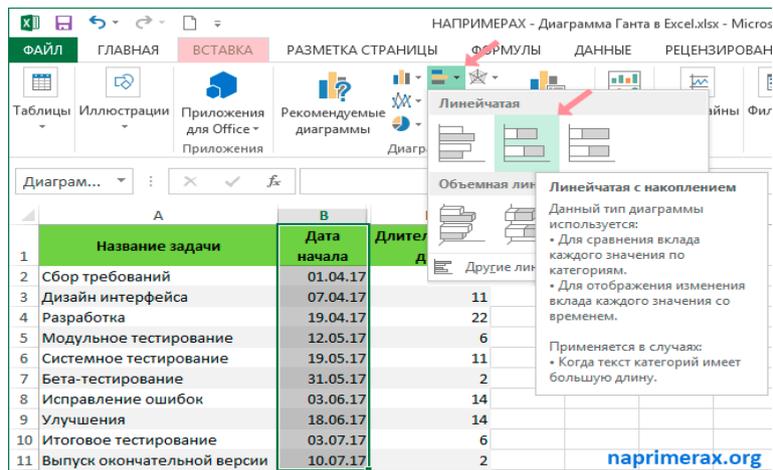


Диаграмма Ганта в Excel – Создание линейчатой диаграммы с накоплением.

В результате на вашем листе будет добавлена следующая диаграмма:

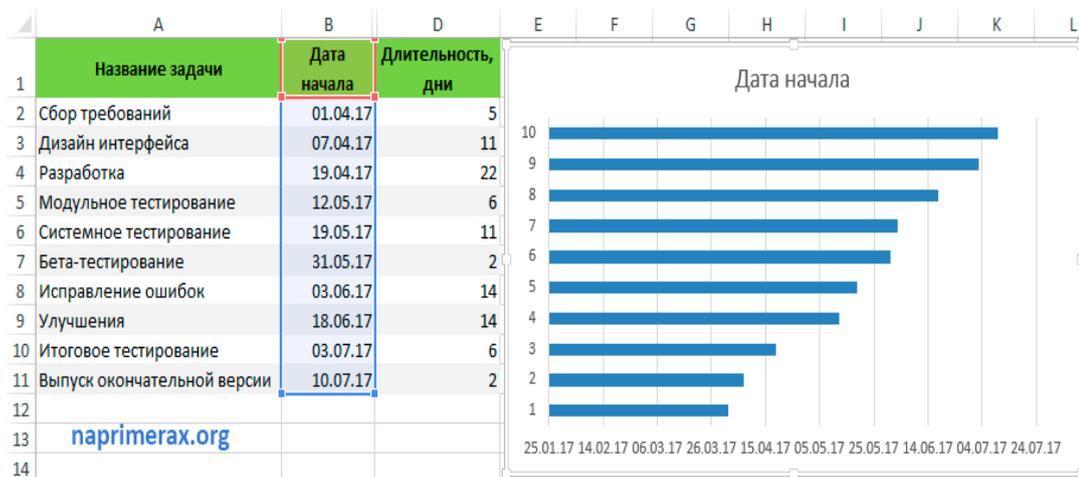


Диаграмма Ганта в Excel – Создание линейчатой диаграммы с накоплением.

Шаг 3. Добавление данных Длительность к диаграмме

Теперь нам нужно добавить еще один ряд в будущую диаграмму Ганта в Excel.

1. Щелкните правой кнопкой мыши в любом месте области диаграммы и выберите «**Выбрать данные**» в контекстном меню.

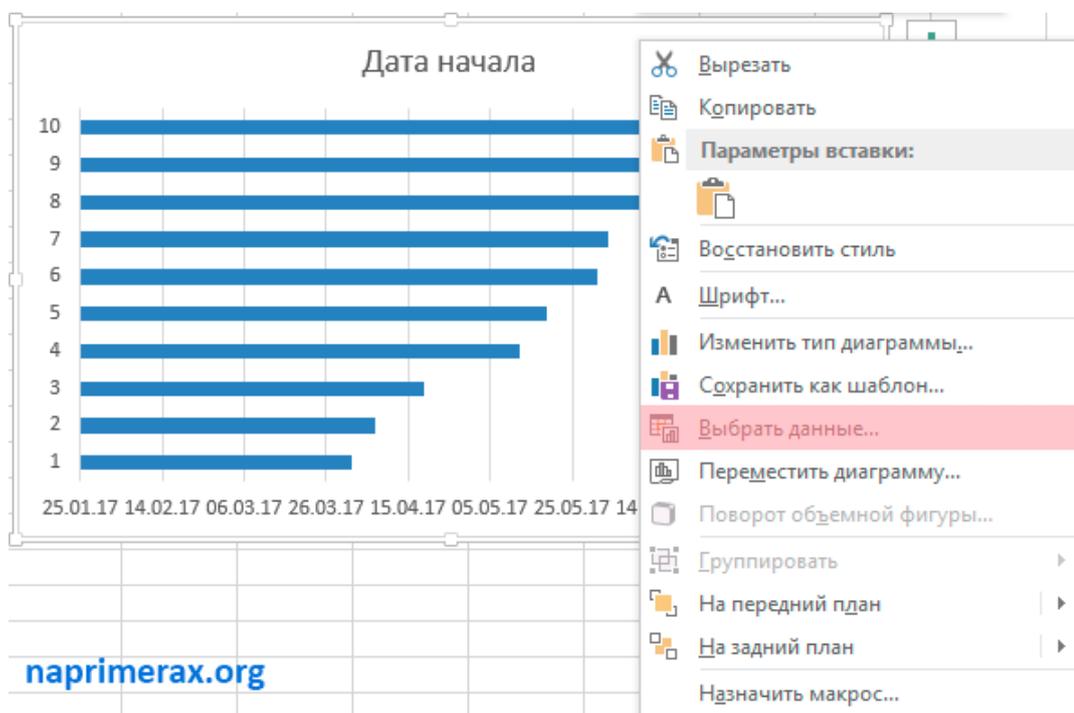


Диаграмма Ганта в Excel – Выбрать данные.

Откроется окно «**Выбор источника данных**».

2. В разделе **Элементы легенды (ряды)** уже добавлена **Дата начала**. Теперь нам нужно добавить **Длительность**.

Нажмите кнопку «**Добавить**», чтобы выбрать дополнительные данные (**Длительность**), которые вы хотите **построить в диаграмме Ганта**.

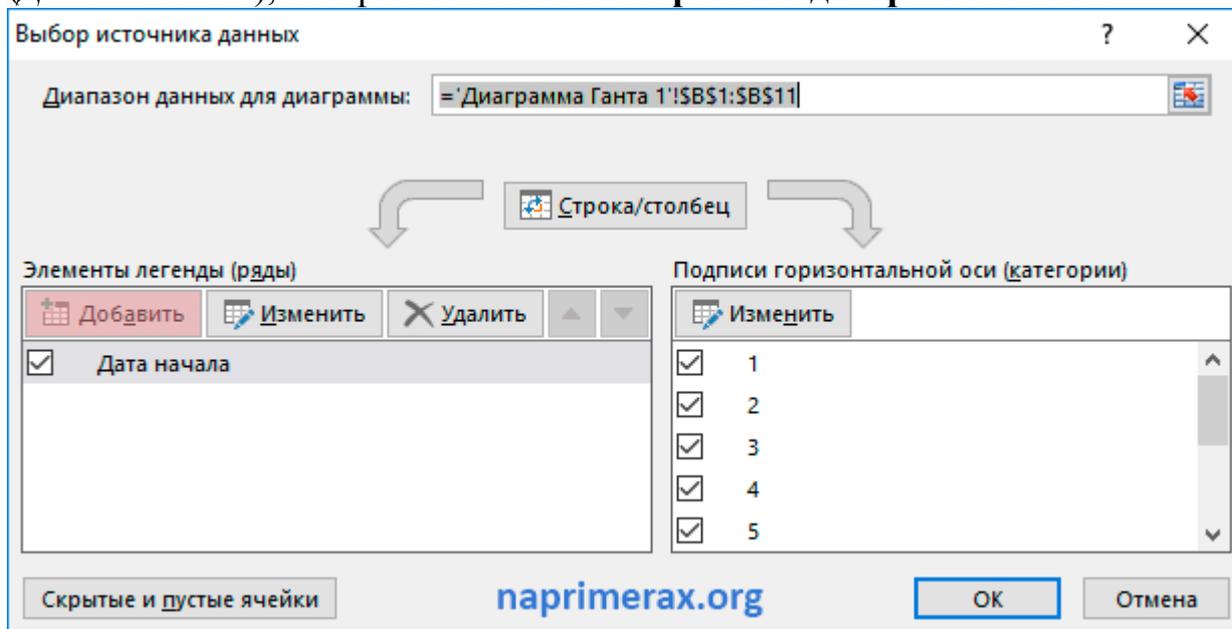


Диаграмма Ганта в Excel – Выбор источника данных.

3. Откроется окно **Изменение ряда**. Выполните следующие пункты:

- В поле «**Имя ряда**» введите «**Длительность**» или любое другое имя по вашему выбору. Кроме того, вы можете поместить курсор мыши в это

поле и щелкнуть заголовок столбца (ячейка D1) в своей таблице, выбранный заголовок будет добавлен как имя ряда для диаграммы Ганта.

• В поле «**Значения**» выберите Длительность, щелкнув по первой ячейке D2 и перетащив мышь до последней D11. Убедитесь, что вы ошибочно не включили заголовок или любую пустую ячейку.

	A	B	D	E	F	G	H	I	J
1	Название задачи	Дата начала	Длительность, дни						
2	Сбор требований	01.04.17	5						
3	Дизайн интерфейса	07.04.17	11						
4	Разработка	19.04.17	22						
5	Модульное тестирование	12.05.17	6						
6	Системное тестирование	19.05.17	11						
7	Бета-тестирование	31.05.17	2						
8	Исправление ошибок	03.06.17	14						
9	Улучшения	18.06.17	14						
10	Итоговое тестирование	03.07.17	6						
11	Выпуск окончательной версии	10.07.17	2						

Изменение ряда

Имя ряда:
Длительность = Длительность

Значения:
= 'Диаграмма Ганта 1'!\$D\$2:\$D\$11 = 5; 11; 22; 6; ...

OK Отмена

naprimerax.org

Диаграмма Ганта в Excel – Добавление ряда данных.

Нажмите кнопку **OK**.

4. Теперь вы вернулись в окно «**Выбор источника данных**» с указанием *начальной даты* и *длительности* в разделе «**Элементы легенды (ряды)**» (*ряды*). Просто нажмите «**OK**» и «**Длительность**» будет добавлена в диаграмму.

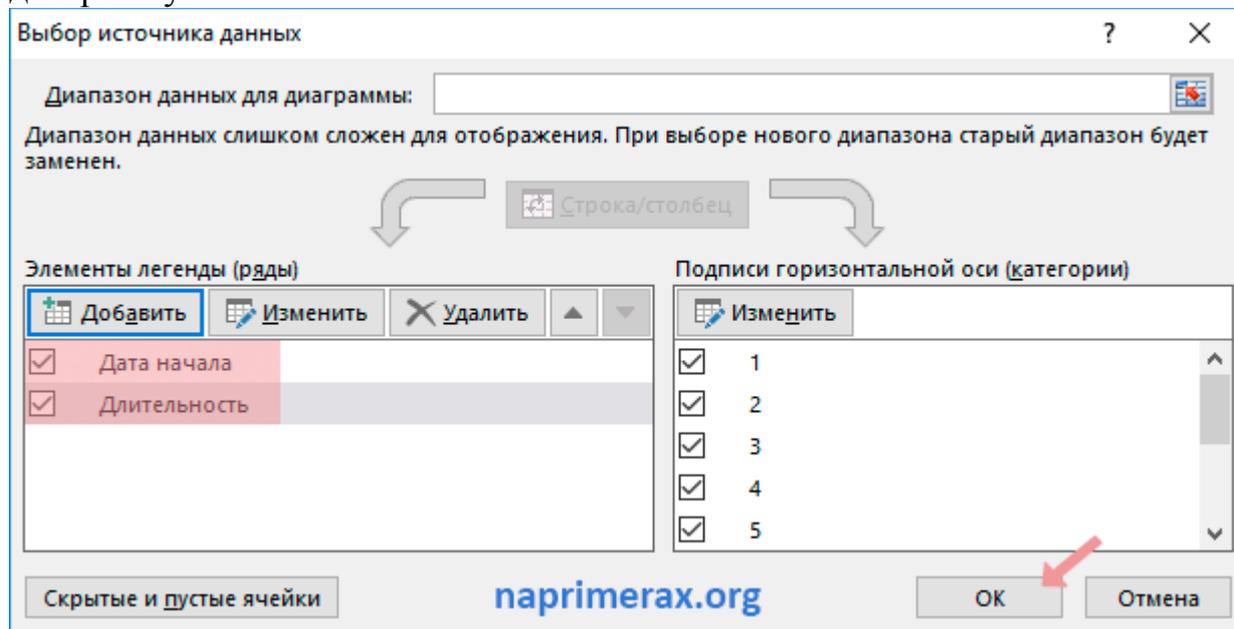


Диаграмма Ганта в Excel – Выбор источника данных – Добавлен новый ряд.

Полученная гистограмма должна выглядеть примерно так:

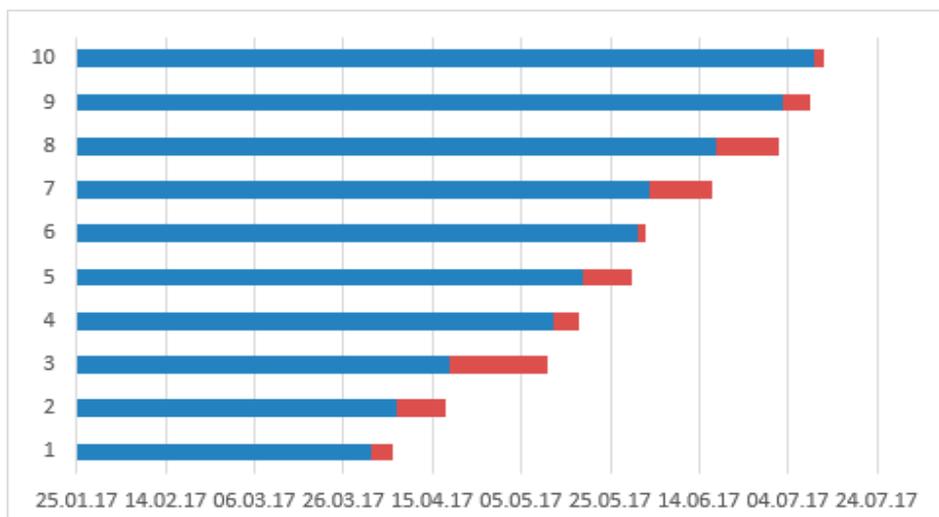


Диаграмма Ганта в Excel – Гисторграмма – Промежуточный вид диаграммы Ганта.

Шаг 4. Добавление описания задач в диаграмме Ганта

Теперь нам нужно заменить дни в левой части диаграммы списком задач.

1. Щелкните правой кнопкой мыши в любом месте области графика (область с синими и красными полосами) и нажмите **«Выбрать данные»**, чтобы снова открыть окно **«Выбор источника данных»**.

2. Убедитесь, что в левой панели выбрана дата начала и нажмите кнопку **«Изменить»** в правой панели под надписью **«Подписи горизонтальной оси (категории)»**.

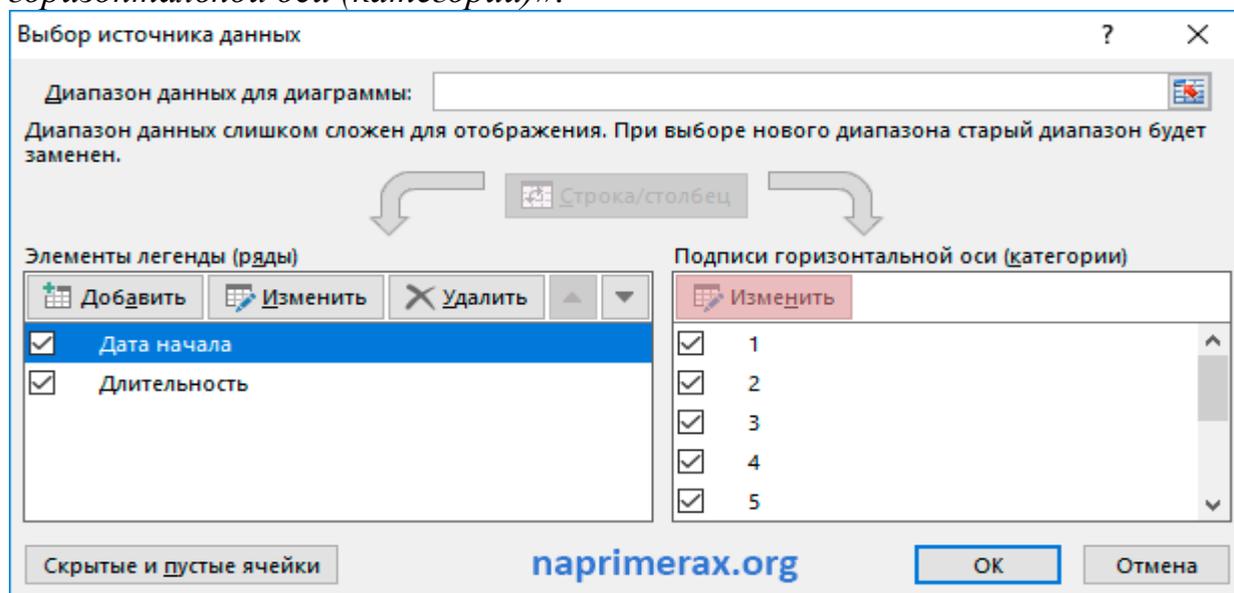


Диаграмма Ганта в Excel – Добавление описания задач в диаграмме Ганта

3. Откроется окно **Подписи оси**, и вы выбираете свои задачи таким же образом, как выбирали «Длительность» на предыдущем шаге. Помните, что заголовок столбца не должен быть включен. Чтобы закрыть открытые окна, два раза нажмите **ОК**.

4. На этом этапе **диаграмма Ганта** должна содержать описание задач в левой части и выглядеть примерно так:

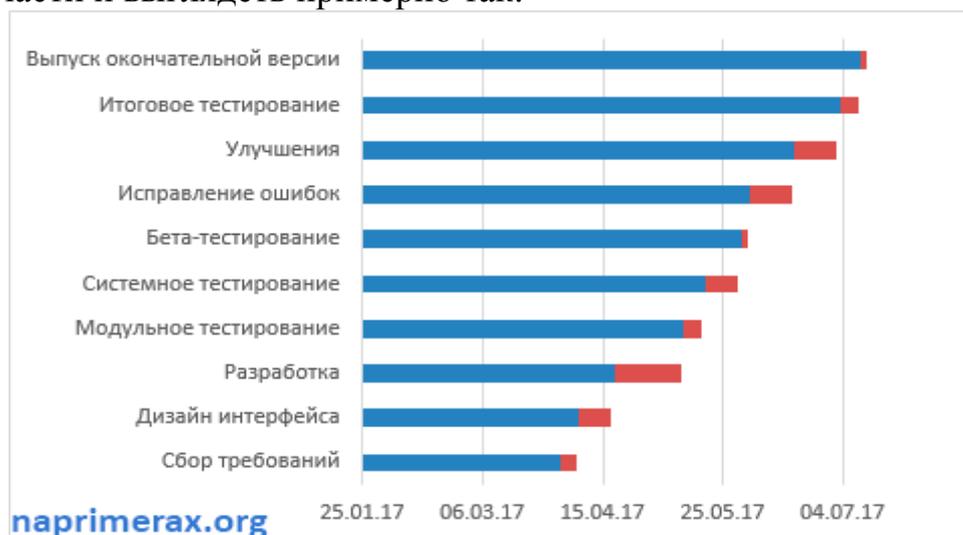


Диаграмма Ганта в Excel – Описание задач в левой части диаграммы Ганта

Шаг 5. Преобразование гистограммы в диаграмму Ганта в Excel

Теперь у нас есть сводная гистограмма. Мы должны добавить правильное форматирование, чтобы данная диаграмма больше напоминала **диаграмму Ганта**. Наша цель – удалить синие полосы, чтобы отображались только оранжевые части, представляющие задачи проекта. С технической точки зрения, мы не будем удалять синие полосы, а просто сделаем их прозрачными и, следовательно, невидимыми.

1. Нажмите на любую **синюю полосу** в **диаграмме Ганта**, чтобы выбрать их все; щелкните правой кнопкой мыши и выберите **«Формат ряда данных»** в контекстном меню.

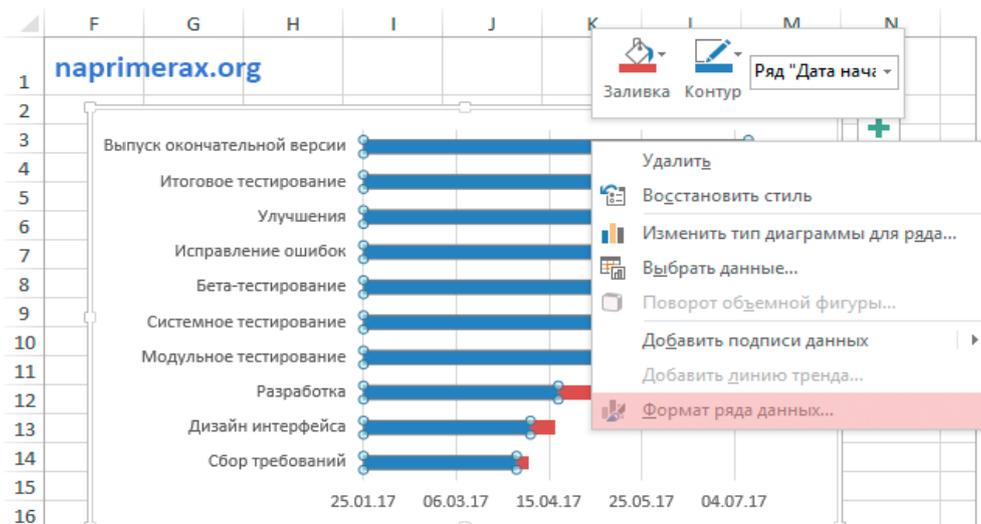


Диаграмма Ганта в Excel – Преобразование гистограммы в диаграмму Ганта

2. Появится окно **«Формат ряда данных»**. Во вкладке **«Заливка и границы»** в группе **«Заливка»** выберите **«Нет заливки»**, и в группе **«Граница»** выберите **«Нет линий»**.

Формат ряда данных

ПАРАМЕТРЫ РЯДА ▾



naprimerax.org

▲ ЗАЛИВКА

- Нет заливки
- Сплошная заливка
- Градиентная заливка
- Рисунок или текстура
- Узорная заливка
- Автовыбор
- Инверсия для чисел <0

▲ ГРАНИЦА

- Нет линий
- Сплошная линия
- Градиентная линия
- Автовыбор

Диаграмма Ганта в Excel – Изменение формата ряда данных в диаграмме Ганта.

3. Как вы, наверное, заметили, задачи в **диаграмме Ганта** перечислены в обратном порядке. И теперь мы это исправим.

Нажмите на список задач в левой части **диаграммы Ганта**, чтобы выбрать их. Появится окно «*Формат оси*». Выберите пункт «**Обратный порядок категорий**» в разделе «*Параметры оси*».

Формат оси ▾

ПАРАМЕТРЫ ОСИ ▾

ПАРАМЕТРЫ ТЕКСТА



naprimerax.org

▲ ПАРАМЕТРЫ ОСИ

Тип оси

- Автовыбор на основе данных
- Ось текста
- Ось дат

Пересечение с горизонтальной осью

- Автовыбор
- В точке со значением категории
- В точке с максимальным значением категории

Положение оси

- Совпадает с делениями
- Находится между делениями

- Обратный порядок категорий

Диаграмма Ганта в Excel – Изменение порядка задач в диаграмме Ганта.

Результаты внесенных изменений:

- Задачи упорядочены в соответствии с **диаграммой Ганта**.
- Маркеры даты перемещаются снизу вверх в начало графика.

Диаграмма Excel начинает выглядеть как обычная **диаграмма Ганта**:

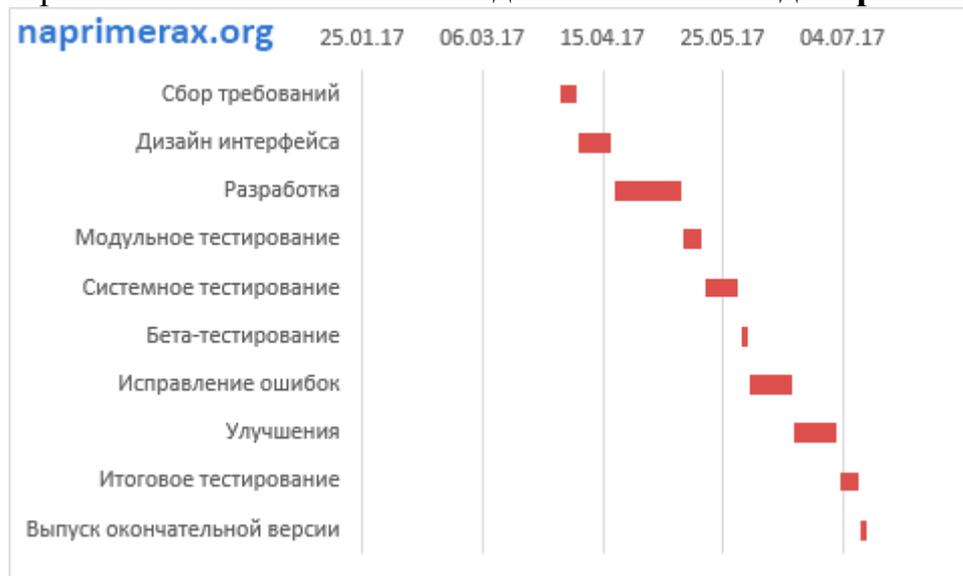


Диаграмма Ганта в Excel – Промежуточная диаграмма Ганта.

Шаг 6. Улучшение дизайна диаграммы Ганта в Excel

Добавим еще несколько штрихов для улучшения внешнего вида **диаграммы Ганта**.

1. Удаление пустого пространства в левой части **диаграммы Ганта**.

Как вы помните, первоначально начальная дата синих полос находилась в начале **диаграммы Ганта**. Теперь вы можете удалить это пустое пространство, чтобы ваши задачи немного приблизились к левой вертикальной оси.

- Щелкните правой кнопкой мыши *по первой начальной дате* в таблице данных, выберите «*Формат ячеек*» --> «*Общий*». Запишите число, которое вы видите – это числовое представление даты, в данном случае 42826. Нажмите «**Отмена**» (!), потому что нам здесь не нужно вносить какие-либо изменения.

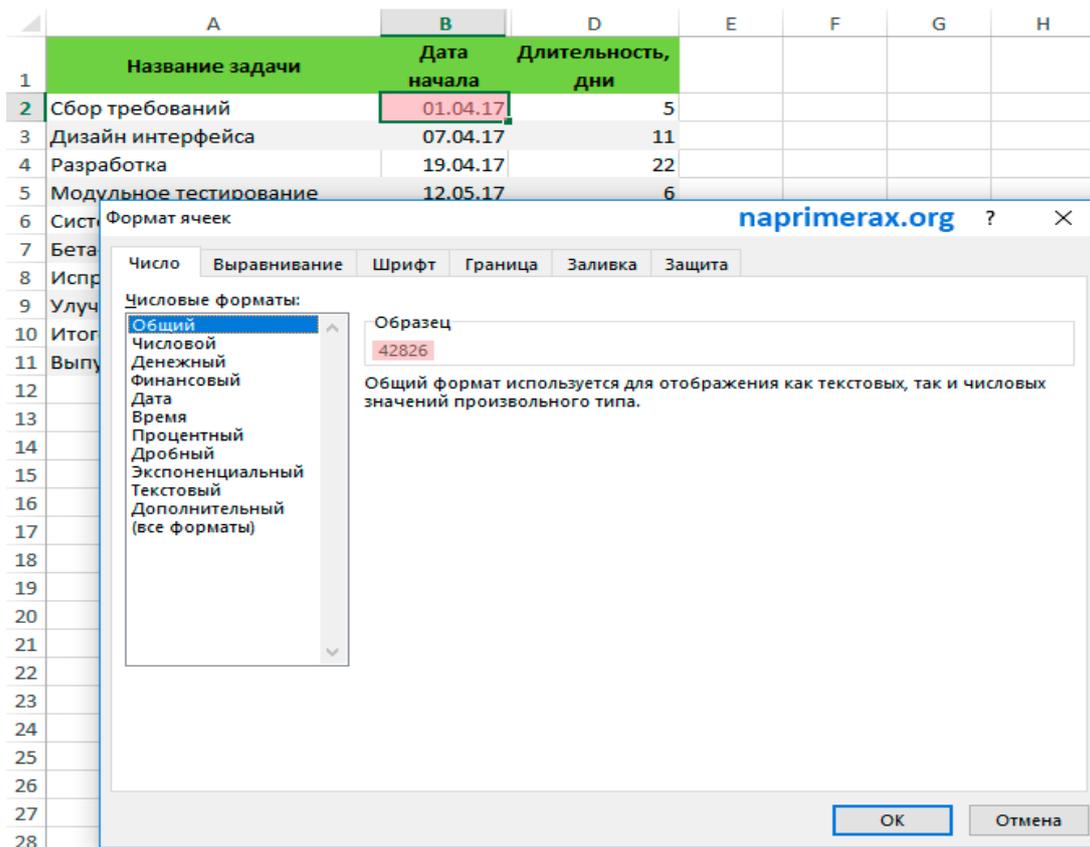


Диаграмма Ганта в Excel – Числовое представление начальной даты.

- Нажмите на любую дату над панелью задач в диаграмме Ганта. Щелкните правой кнопкой мыши в данной области и выберите «**Формат оси**» в контекстном меню.

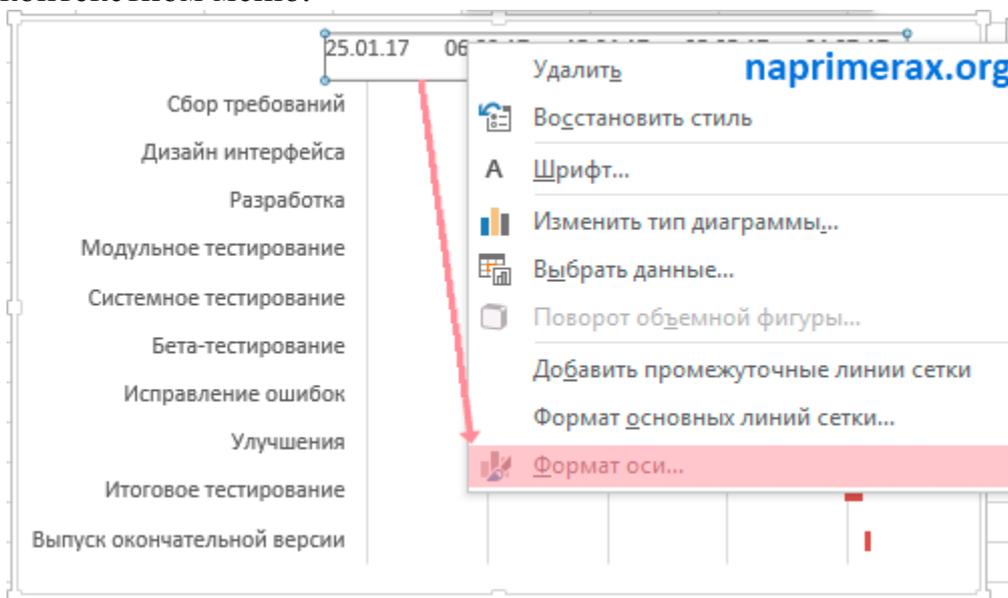


Диаграмма Ганта в Excel – Удаление пустой области в правой части диаграммы Ганта

- В разделе «*Параметры оси*» в поле «**Минимум**» введите число, записанное на предыдущем шаге. Также в этом разделе вы можете изменить *основные* и *промежуточные* деления для интервалов дат. Как

правило, чем короче время вашего проекта, тем меньшее число вы используете. Вы можете увидеть какие настройки мы используем в нашем примере на нижеприведенном изображении.

Примечание. Если вы пользуетесь Excel 2010, для того чтобы ввести число, вам необходимо выбрать вариант «*фиксированное*».

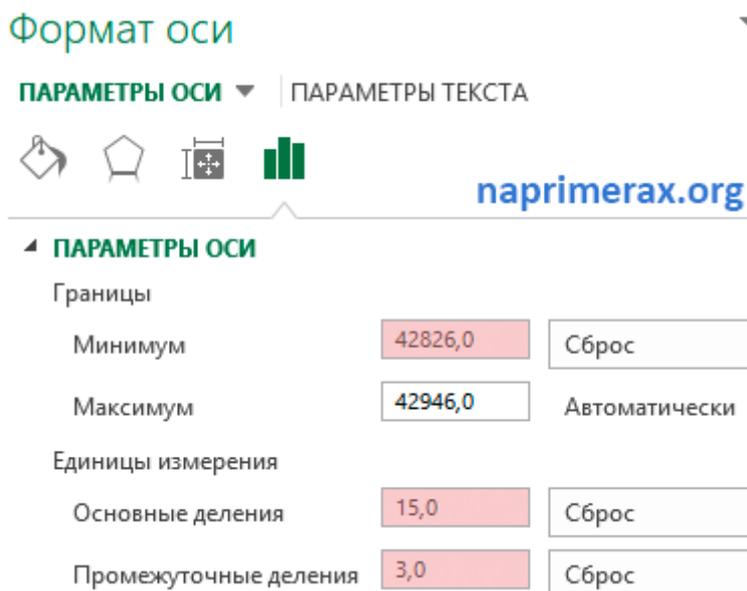


Диаграмма Ганта в Excel – Изменение дат в диаграмме Ганта.

2. Удаление лишнего пустого пространства между полосами.

Уплотнение полос задач сделает **диаграмму Ганта** еще лучше.

- Нажмите на любую *красную полосу*, чтобы выбрать их все; щелкните правой кнопкой мыши и выберите «**Формат ряда данных**».

- В диалоговом окне «Формат ряда данных» установите **Перекрывание рядов** на 100%, а **Боковой зазор** - на 0% (или близкий к 0%).

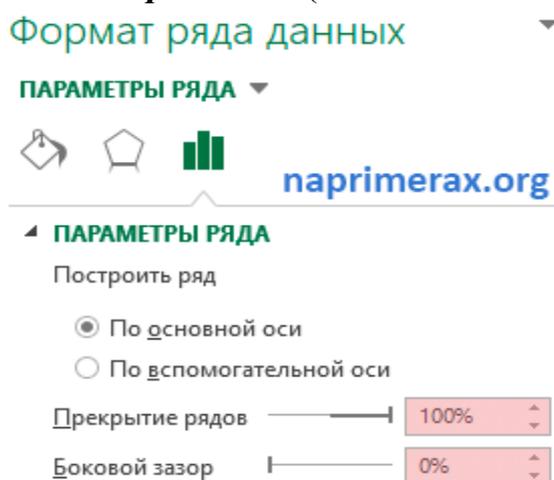


Диаграмма Ганта в Excel – Удаление пустого пространства в диаграмме Ганта.

И вот результат наших усилий – простая, но симпатичная **диаграмма Ганта в Excel**:

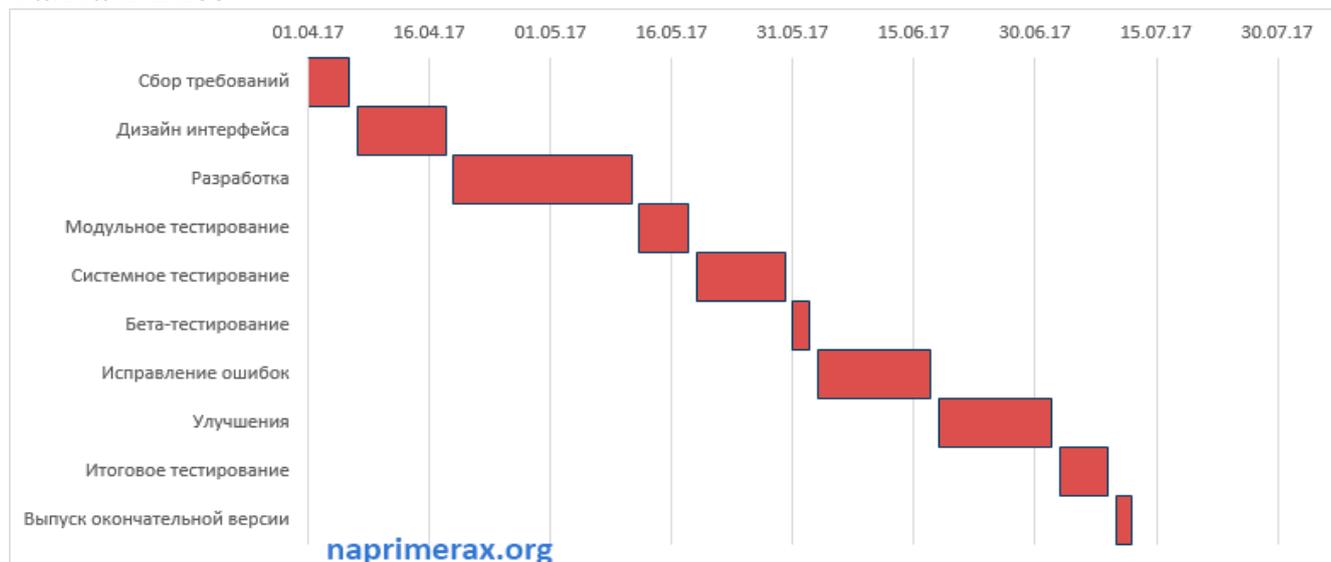


Диаграмма Ганта в Excel – Финальная версия диаграммы Ганта.

Помните, что, хотя диаграмма Excel очень хорошо имитирует диаграмму Ганта, она по-прежнему сохраняет основные функции стандартной **диаграммы Excel**:

- **Диаграмма Ганта в Excel** изменит размер при добавлении или удалении задач.
- Вы можете изменить Дату начала или Длительность, диаграмма будет отражать изменения и автоматически настраиваться.

Диаграмма Ганта в Excel шаблон

Вы можете видоизменить свою **диаграмму Ганта в Excel** различными способами, изменяя цвет заливки, цвет границы, тень и даже применяя трехмерный формат. Все эти параметры доступны в окне «*Формат данных*».

Когда вы создали свой дизайн, хорошей идеей будет сохранить **диаграмму Ганта в Excel** как шаблон для будущего использования. Для этого щелкните правой клавишей мыши в области диаграммы, и выберите пункт «*Сохранить как шаблон*» в контекстном меню.

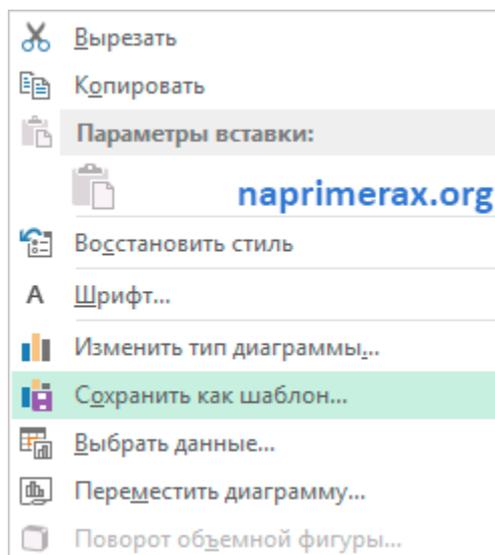


Диаграмма Ганта в Excel – Сохранить шаблон диаграммы Ганта в Excel.

Задание.

Выполнить разработку графика разработки и внедрения информационной системы с помощью диаграммы Ганта в Excel для своей информационной системы.

ПРАКТИЧЕСКАЯ РАБОТА № 4 «Сравнительный анализ методологий проектирования»

Цель работы: приобретение навыков анализа методологий проектирования

Методика выполнения работы

Краткие теоретические сведения

На сегодняшний день существует множество разнообразных методологий построения процесса разработки ПО, и у каждой из них есть свои плюсы и минусы, области применения, в которых определенные из них наиболее эффективны. Все эти методологии преследуют своей первой целью улучшение производственного процесса, который позволил бы наиболее эффективно и качественно производить программные продукты.

Модели (или методологии) процессов разработки программного продукта принято классифицировать по «весу» – количеству формализованных процессов (большинство процессов или только основные) и детальности их регламентации. Чем больше процессов документировано, чем более детально они описаны, тем больше «вес» модели.

Модель процессов MSF

Microsoft Solutions Framework (MSF) – это гибкая и достаточно легковесная модель, построенная на основе итеративной разработки.

Модель процессов MSF описывает общие подходы к разработке и внедрению ПО. Благодаря своей гибкости она может быть применена при разработке весьма широкого круга проектов. Эта модель сочетает в себе свойства двух стандартных моделей: каскадной и спиральной. Модель процессов MSF охватывает весь жизненный цикл создания решения, начиная с его отправной точки и заканчивая внедрением.

Модель процессов MSF опирается на следующие базовые принципы:

- подход, основанный на фазах и вехах;
- итеративный подход;
- интегрированный подход к созданию и внедрению решений.

Модель процессов включает такие основные фазы процесса разработки как:

- выработка концепции;
- планирование;
- разработка;
- стабилизация;
- внедрение.

Процесс MSF ориентирован на «вехи» (milestones) – ключевые точки проекта, характеризующие достижение в его рамках какого-либо существенного (промежуточного либо конечного) результата, причем этот результат может быть оценен и проанализирован. Критерии оценки результата должны быть сформулированы еще до начала проекта.

Для декомпозиции больших этапов работы может быть введено также большое количество промежуточных вех.

Модель процессов MSF учитывает постоянные изменения проектных требований. Она исходит из того, что разработка решения должна состоять из коротких циклов, реализующих поступательное движение от начальных версий системы к ее окончательному виду.

В рамках MSF программный код, документация, дизайн, планы и другие рабочие материалы создаются, как правило, итеративными методами. MSF рекомендует начинать разработку решения с построения, тестирования и внедрения его базовой функциональности. Затем к решению добавляются все новые и новые возможности, т. е. с каждой новой версией эволюционирует функциональность решения. Для малых проектов может быть достаточным выпуск одной версии.

Итеративный подход к процессу разработки требует использования гибкого способа ведения документации. Документация должна изменяться по мере реализации проекта вместе с изменениями требований к конечному продукту. В рамках MSF предлагается ряд шаблонов стандартных документов для каждой стадии разработки продукта, которые могут быть

использованы для планирования и контроля процесса разработки. Решение не представляет ценности, пока оно не внедрено. Именно по этой причине модель процессов MSF содержит весь жизненный цикл создания решения, включая его внедрение, вплоть до момента, когда решение начинает давать отдачу.

ТЕХНОЛОГИЯ XP

Технология XP (англ. Extreme Programming) базируется на спиральной модели жизненного цикла. Авторами технологии являются К. Бек, У. Каннингем, М. Фаулер. Название технологии связано со стремлением авторов поднять существующие методы разработки ИС (и программного обеспечения в целом) на новый, «экстремальный» уровень.

Для оценки проектов с точки зрения применимости XP применяются два показателя – критичность и масштаб. Критичность определяется последствиями, вызываемыми дефектами ПО, и может иметь один из четырех уровней:

1. С – дефекты вызывают потерю удобства.
2. D – дефекты вызывают потерю возместимых ресурсов.
3. E – дефекты вызывают потерю невозместимых ресурсов.
4. L – дефекты могут создавать угрозу для человеческой жизни.

Масштаб определяется количеством разработчиков, участвующих в проекте:

- от 1 до 6 человек
- от 6 до 20 человек
- свыше 20 человек

- малый масштаб;
- средний масштаб;
- большой масштаб.

По оценке специалиста по разработке ПО А. Коберна, XP применима в проектах малого и среднего масштаба с низкой критичностью (С или D).

Разработка в XP ведется небольшими трехнедельными итерациями, в течение которых уточняются и реализуются требования к системе. При разработке применяется **рефакторинг** – методика улучшения кода без изменения его функциональности. Программный код в процессе работы над проектом неоднократно переделывается, в том числе и на поздних стадиях проекта.

Для того чтобы эти переделки не привели к неработоспособности системы, используется методика TDD (**Test-Driven Development** – разработка через тестирование). Технология XP предполагает написание автоматических тестов – специальных программ, написанных для тестирования других программ. Ручная прогонка тестов здесь невозможна, т. к. количество тестов

слишком велико. Тесты пишутся еще до того, как начинается создание системы, поэтому риск получения неработоспособной версии из-за постоянно вносимых изменений существенно снижается – любую новую версию тут же можно протестировать.

В процессе создания системы применяются такие характерные для XP методы, как парное программирование, непрерывная интеграция, упрощенное проектирование.

Парное программирование предполагает, что программы создаются парами программистов, работающих за одним компьютером. Один из них пишет непосредственно текст программы, другой оценивает его работу, благодаря чему становится возможной постоянная проверка программного кода. В течение работы над проектом пары не фиксируются: это делается с той целью, чтобы каждый программист в команде имел хорошее представление обо всей системе. Повышение эффективности при работе парой программистов подтверждено специальными исследованиями.

Непрерывная интеграция (сборка) системы позволяет поддерживать ее целостность в течение всего процесса разработки. В традиционных методиках интеграция выполняется в самом конце работы над продуктом, когда все составные части разрабатываемой системы полностью готовы. Интеграционные проблемы обладают способностью накапливаться и наслаиваться друг на друга, что может даже привести к провалу проекта. В XP интеграция системы выполняется несколько раз в день, после того, как все модули прошли положенные для них тесты. Это позволяет выявить проблемы интеграции на возможно более ранней стадии разработки и заблаговременно принять необходимые шаги к их преодолению.

Упрощенное проектирование применяется в XP из-за того, что в процессе работы требования к системе могут неоднократно меняться, что снижает ценность проекта, выполненного целиком в самом начале разработки. Для XP характерно непрерывное проектирование, выполняемое в течение всего времени работы над проектом. Проектирование должно выполняться небольшими этапами, с учетом постоянно изменяющихся требований. В каждый момент времени следует использовать наиболее простые решения, которые подходят для решения текущей задачи, и менять его по мере того, как условия задачи меняются. Согласно К. Беку, упрощенное проектирование обеспечивает корректное выполнение всех тестов, не порождает дублирующего кода, включает наименьшее количество классов и методов, ясно выражает цель программиста.

Помимо перечисленного, все члены команды в ходе работы должны соблюдать *общие требования стандартов программирования*, что существенно облегчает рефакторинг и снижает риски проекта, связанные с текучкой кадров. В идеале соблюдение стандартов программирования должно полностью исключить индивидуальные черты стиля разработки – программный продукт должен выглядеть как результат работы одного человека.

Коллективное владение означает, что каждый член команды несет ответственность за весь исходный код. Каждый вправе вносить изменения в любой участок программы. Сопутствующие риски от вносимых изменений устраняются мощной системой тестирования. Однако это не порождает безответственности, поскольку существует требование, согласно которому каждый программист должен сам исправить сделанные им ошибки. Важное преимущество коллективного владения кодом состоит в том, что оно ускоряет процесс разработки, поскольку при необходимости любой программист может оперативно внести изменения в любую часть кода.

Существенным считается и наличие *метафоры системы* – простой аналогии, понятной всем участникам проекта, которая с достаточной точностью описывает функционирование и внутреннюю структуру ИС.

Каждая итерация обычно длится две-три недели и выглядит как программный проект в миниатюре, включая все этапы жизненного цикла системы:

- планирование;
- анализ требований;
- проектирование;
- программирование;
- тестирование;
- документирование.

Хотя отдельная итерация, как правило, недостаточна для выпуска новой версии продукта, подразумевается, что гибкий программный проект готов к работе в конце каждой итерации. По окончании каждой итерации команда выполняет переоценку приоритетов разработки.

МЕТОДОЛОГИЯ RUP

Rational Unified Process (RUP) – методология разработки программного обеспечения, созданная компанией Rational Software, с 2003 года входящей в корпорацию IBM. Методология RUP основана на спиральной модели жизненного цикла ИС. В качестве языка моделирования в RUP используется язык Unified Modelling Language (UML).

Наибольшее внимание RUP уделяет начальным стадиям разработки проекта – анализу и моделированию, – что призвано снизить риски проекта за счет раннего обнаружения возможных ошибок. Последовательный выпуск версий организован таким образом, чтобы наиболее существенные риски устранялись в первую очередь.

Методология RUP широко использует так называемые *прецеденты*, или сценарии использования – описание последовательностей действий, которые может осуществлять система, взаимодействуя с внешними действующими факторами. Прецеденты создаются при помощи UML и включают варианты как правильных, так и ошибочных последовательностей (исключений). Прецеденты служат для документирования требований заказчика к проектируемой информационной системе. Прецедент описывает целостный фрагмент поведения системы в виде последовательности сообщений, которыми система обменивается с действующими лицами.

Другие существенные черты методологии RUP:

- заранее предусматриваются изменения в требованиях и проектных решениях в течение всего процесса разработки;
- постоянное обеспечение качества на всех этапах разработки ИС;
 - компонентная архитектура используется начиная с ранних стадий проекта.

Процессы и стадии RUP

RUP использует итеративную модель разработки. В конце каждой итерации (продолжительностью в несколько недель) команда разработчиков должна получить функционирующую версию конечного продукта, позволяющую достичь запланированных на данную итерацию целей. Итеративная разработка позволяет быстро реагировать на меняющиеся требования, обнаруживать и устранять риски на ранних стадиях проекта, а также эффективно контролировать качество создаваемого продукта.

Полный жизненный цикл разработки ИС состоит из четырех фаз, описываемых ниже.

Каждая фаза может включать в себя одну или несколько итераций процесса создания системы.

Начальная стадия

В фазе начальной стадии:

- создается экономическое обоснование разработки,
- определяются основные требования, ограничения,
- создается базовая версия модели прецедентов,
- оцениваются риски.

Уточнение

В фазе «Уточнение» производится анализ исходных данных для проектирования и выбор архитектуры ИС.

Фаза включает в себя:

- анализ требований заказчика, включая детальное описание для большинства прецедентов, интерфейса,
 - проектирование архитектуры ИС, спецификация функций и пользовательского
- планирование работ по проекту и всех необходимых ресурсов.

Построение

В фазе «Построение» происходит итеративная реализация требуемых функций ИС. Это основная фаза проектирования и создания программного кода. Фаза завершается выпуском бета- версии системы.

Внедрение

В фазе «Внедрение» финальная версия системы внедряется у ее заказчика. Фаза включает проведение испытаний системы, обучение пользователей, а также оценку качества ИС. В том случае, если качество системы не соответствует требованиям заказчика, фаза «Внедрение» выполняется повторно. Выполнение всех требований и достижение целей проекта означает завершение полного цикла разработки.

Выполнение работы

1. Изучите методологии проектирования ИС
2. Дайте характеристику каждой методологии:

- 1) Назначение;
- 2) «вес» модели;
- 3) Модель жизненного цикла;
- 4) Фазы и стадии методологии

3. Перечислите преимущества и недостатки разработки ПО по «весу», заполнив таблицу:

<i>Вес модели</i>	<i>Преимущества</i>	<i>Недостатки</i>
Тяжелые		
Легкие		

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №5 Анализ бизнес - процессов подразделения

Цель работы: Получить практические навыки в выделении и описании компонент бизнес-процесса, в построении структурной модели процесса и процессной организационной структуры управления.

Порядок выполнения работы.

1. Выбор задания.

Выберите бизнес-процесс, для которого будете формировать модель. Вы можете выбрать один из вариантов процессов, описанных в приложении, или предложить свой вариант.

2. Общая характеристика процесса.

Дайте краткую характеристику процесса, указав:

- организацию (компанию, фирму, учреждение), использующую процесс;

- является этот процесс производственным или процессом управления;

- для производственных процессов – является ли он основным (связанным с производством конечных продуктов для внешнего потребителя) или вспомогательным (связанным с обеспечением основных процессов ресурсами, с поддержанием ресурсов);

- для процессов управления – является ли он процессом текущего управления (направленным на управление существующими производственными процессами) или процессом совершенствования (направленным на обновление существующих процессов или на разработку

новых бизнес-процессов).

3. Характеристика компонент процесса.

Опишите компоненты и характеристики процесса:

- что является результатом процесса;
- кто является потребителем процесса (это могут быть не только внешние потребители – клиенты, заказчики, другие организации, но подразделения или процессы той же компании, которая выполняет исследуемый процесс);
 - как инициируется процесс, т.е. какое событие запускает выполнение процесса (например, обращение клиента);
 - периодичность выполнения процесса;
 - входы процесса (сырье, материалы, входная информация, т.е. объекты, которые перерабатываются для получения выхода – результата процесса);
 - кто является поставщиком процесса (процесс может получать входы не только от внешних поставщиков, от клиентов, но и от других процессов, подразделений той же компании, которая выполняет процесс);
 - ключевые показатели результативности процесса (метрики), например: среднее время выполнения заказа; объем продукции в месяц; среднее количество клиентов в месяц; себестоимость продукции (услуги); качество продукции; процент брака; удобство обслуживания клиента.

4. Функциональная декомпозиция бизнес-процесса.

Разбейте исследуемый процесс сначала на крупные функции-этапы; затем каждую из этих функций – на более мелкие функции-подэтапы. Выделенные функции- подэтапы (или некоторые из них) разбейте на еще более мелкие функции -операции. Можно еще детализировать отдельные операции.

Представьте иерархию функций процесса в виде схемы. Пример схемы приведен на рис. 1.1.





Рисунок 1.1– Иерархия функций процесса

5. Выделение структурных элементов функций.

Для каждой из полученных в результате декомпозиции функций (всех уровней, включая процесс в целом), выделите структурные элементы следующих типов:

- входы – объекты, которые преобразуются в выходы, например, *сырье, материалы, заявка клиента*;
- выходы – объекты, являющиеся результатом выполнения функции, например, *изготовленный продукт, выполненная услуга, обработанная заявка*;
- исполнители – подразделения компании или отдельные исполнители, выполняющие функцию, например, *продавец, отдел приема заявок, бригада рабочих, дизайнер*;
- оборудование, инструменты, используемые при выполнении функции, например, *станок, компьютер, торговое оборудование, офисное оборудование*;
- управляющая, регламентирующая информация, показывающая, как выполняется функция, или принимаемая во внимание при выполнении функции, например, *план, проект, инструкция, чертеж*.

Представьте перечень структурных элементов каждой функциональной подсистемы в виде таблицы (см. таблицу 1.1).

Некоторые элементы у разных функций могут совпадать. Чем выше уровень функциональной подсистемы, тем более обобщенными являются элементы. У функций нижних уровней элементы более конкретные, они могут детализировать элементы вышестоящих уровней. Например, в качестве исполнителя функции "Прием заявок" можно указать отдел продаж, а для подфункций "Прием заявки", "Прием оплаты" можно указать конкретных сотрудников этого отдела – продавца-консультанта, кассира.

Таблица 1.1 – Структурные элементы функциональных подсистем

Функция	Вход	Исполнитель	Оборудование инструменты	Управляющая информация	Выход
Продажа мебели на заказ	Заявка клиента, материалы	Персонал фирмы	Оборудование	Спецификации изготавливаемой мебели	Доставленная клиенту мебель
Оформление заказа	Заявка клиента	Отдел продаж	Торговое оборудование	Каталоги мебели	Оформленный заказ, договор
Изготовление мебели	Материалы	Производственный цех	Производственное оборудование	Оформленный заказ, спецификации изготавливаемой мебели	Готовая мебель
Доставка мебели клиенту	Готовая мебель	Отдел доставки	Транспорт	Адрес клиента	Доставленная клиенту мебель
...					

Задание.

1. Описать краткую характеристику бизнес-процесса.
2. Выполнить иерархию функций процесса в виде схемы (рис.1.1).
3. Выполнить структуру предприятия с помощью SmartArt в Word.
(<https://yandex.ru/video/preview/?filmId=5274912515345877899&text=создание%20бизнес%20процесса%20с%20помощью%20смартарт%20в%20ворде&path=wizard&parent-reqid=1590733624531804-1029048885749624578200292-production-app-host-man-web-yp-74&redircnt=1590733629.1>)
4. Создать схему бизнес процесса для информационной системы. Выполнить 4 пункта. (<https://rzbpm.ru/knowledge/sozдание-sxemy-biznes-processa-dlya-neterpelivyx.html>).

ПРАКТИЧЕСКАЯ РАБОТА №6
«Разработка и оформление предложений по расширению функциональности информационной системы»

Цель: научиться разрабатывать и оформлять приложения по расширению функциональности информационной системы.

Форма отчета:

- выполнить задание;
- показать преподавателю;
- ответить на вопросы преподавателя.

Программное обеспечение (ПО) от англ. <i>software</i>	совокупность программ, хранящихся на устройствах долговременной памяти компьютера и предназначенных для решения поставленных задач.
Прикладное ПО	совокупность программ, посредством которых пользователь решает свои информационные задачи.
Системное ПО	совокупность программных средств, предназначенных для организации диалога с пользователем, поддержания функционирования компьютера и управления устройствами компьютера.
Операционная система (ОС) от англ. <i>operating system</i>	комплекс управляющих и обрабатывающих программ, осуществляющих диалог с пользователем, управление компьютером, его ресурсами и другими программами.
Драйверы от англ. <i>driver</i> – «водитель»	программы, обеспечивающие взаимодействие прикладных программ и операционной системы с внешними устройствами.
Утилиты от англ. <i>Utility</i> «служебный»	программы вспомогательного назначения, расширяющие возможности операционной системы.
Система программирования	система для разработки новых программ на конкретном языке программирования.
Лицензия на программное обеспечение	правовой инструмент, определяющий использование и распространение программного обеспечения, защищённого авторским правом.
Файл	поименованная совокупность записей информации, размещенных на диске.
Программа	последовательность инструкций (команд), описывающая алгоритм решения с помощью компьютера соответствующей задачи, для реализации которой эта программа была разработана.

ПРАКТИКУМ

Задание № 1

Соотнесите данные программы к своему классу программного обеспечения. Запишите в таблице под каждой буквой необходимые программы и опишите их назначение.

Paint, Windows Media Player, Калькулятор, Dr Web, Фортран, Си, Лисп, Windows Vista, Pascal, WinRar, Касперский, Ассемблер, Avast, Блокнот, Skype, Алгол, ISQ, Linux, MS Office Word, операционные системы, WinZip, Пролог, драйвера, C++, MS Office Excel, игры, переводчики, проигрыватели, Adobe PhotoShop, утилиты, Basic, WordPad, Linux, Autocad, CCleaner, Scandisk, Delphi, MS DOS, FineReader

А системное	Б прикладное	В системы программирования

Контрольные вопросы по теме:

1. Продолжите фразу: Компьютер представляет собой единство двух составляющих...
2. Что такое программное обеспечение?
3. Какая разница между ПО и собственно программой?
4. Виды программного обеспечения?
5. К какому виду ПО относятся утилиты?
6. Какова цель использования прикладных программ?
7. Перечислите несколько примеров прикладного ПО для создания текстовых и графических документов.
8. Какие программы называются прикладными программами специального назначения?

Задание № 2

По данной теме составьте кроссворд или тест от 10-15 слов или вопросов.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №7

«Разработка перечня обучающей документации на информационную систему»

Цель работы: Ознакомиться с процедурой разработки руководства пользователя на создание информационной системы (ИС) с применением РД 50-34-698-90 «Автоматизированные системы требования к содержанию документов»

http://www.rugost.com/index.php?option=com_content&view=article&id=98:50-34698-90&catid=22&Itemid=53

Основные теоретические сведения

Формальные требования к документации программного обеспечения описаны в ЕСПД (Единая система программной документации), неформально: состав документации к программному обеспечению состоит из описания его внешнего эффекта и описания его внутреннего устройства.

Первая часть документации, так называемая «Инструкция пользователю» или «Руководство пользователю» предназначена для того, кто собирается использовать программное обеспечение (для пользователя), не вникая в подробности его внутреннего устройства.

Вторая часть – «Руководство программисту» необходима при модификации программного обеспечения или при необходимости исправить в нем ошибку.

В целом, документация к программному обеспечению может содержать ниже перечисленные сведения:

1. Наименование ПО и описание задачи, которую оно решает.
2. Область применимости ПО.

3.Режим работы ПО, сообщения, выдаваемые по ходу его работы, ответы пользователя на них (если это необходимо).

4.Исходные данные, необходимые для работы ПО; а также выдаваемые им результаты.

5.Правила подготовки исходных данных на внешних носителях (если они применяются) и вид выдаваемой информации.

6.Описание структуры данных. Для любой переменной описывается ее назначение, атрибуты (тип, размер массива и т.д.), структура информации в ней, если она не очевидна. Описание переменных должно начинаться с тех, которые служат исходными данными и результатами.

7.Описания форм, объектов. Опись свойств форм и объектов.

8.Тексты программ, процедур (в виде распечатки ЭВМ) с комментариями.

9.Тесты.

10.Инструкция (руководство) пользователю.

Инструкция по использованию программы (или просто «Инструкция пользователю», или «Руководство для пользователя») - это выдержка из полной документации, предназначенная для эксплуатации программы. Она представляет собой независимый документ для пользователя программы, в котором описывается: что делает программа и как им пользоваться.

«Инструкция пользователю» должна содержать всю необходимую для пользователя информацию и должна быть ему понятна без дополнительных материалов (без обращения к другим спецификациям). Следовательно, необходимая для этой инструкции информация переписывается полностью из соответствующих спецификаций.

Первая часть инструкции является описательной и должна содержать:

- наименование программы;
- краткое описание программы;
- перечень выполняемых программой функций;
- краткую характеристику метода (или методов) решения поставленной задачи, его достоинство и недостатки;
- полную библиографическую ссылку на полное описание метода;
- описание входных и выходных данных.
- описание структуры базы данных (если она имеется), всех ее таблиц в словесной (вербальном) форме.

Вторая часть документа должна описывать порядок работы с информационной системой. Она должна содержать описание всех режимов работы ИС, а также содержание всех печатей и диагностических сообщений, которые выдаются по ходу выполнения ИС.

Следует помнить, что пользователь по своей квалификации не является программистом и поэтому его работа с программой описывается на понятном ему языке и достаточно подробно, а именно:

- как запустить ИС;

- как продолжить работу с ИС (описывается подробный интерактивный режим работы пользователя с программой);
- подготовка и ввод исходных данных в ИС;
- как реагировать на запросы ИС;
- как вести работу в исключительных ситуациях;
- как реагировать на ошибки;
- как восстановить работу программы в случае аварийного его завершения;
- как получить требуемый результат;
- как правильно закончить работу с программой (запланированный программой выход);
- другие сведения, необходимые пользователю программы.

Задания для выполнения.

1. Составить документ «Руководство пользователю» к разработанной ранее информационной системе
2. Работа должна быть оформлена в виде документа «Руководство пользователя» согласно РД 50-34-698-90 «Автоматизированные системы требования к содержанию документов».
3. Сдать и защитить работу.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №8 «Разработка руководства оператора»

Руководство оператора

Руководство оператора – это основной документ в составе эксплуатационной документации на Информационную систему (ГОСТ 19).



Назначение руководства оператора

В соответствии с государственными стандартами, Руководство оператора входит в состав комплекта эксплуатационной документации на программное обеспечение. Для чего нужен такой документ? Чтобы ответить

на этот вопрос, необходимо понять, какую роль в использовании системы играет оператор.

Мы знаем, что **администратор** отвечает за настройку системы и поддержку работы пользователей. Пользователь же, в свою очередь, выполняет с помощью системы определенные прикладные функции, решает прикладные задачи. Роль оператора по своим функциям ближе всего к **роли пользователя**, однако, отличается от нее тем, что перед оператором не ставятся прикладные задачи, которые он может решить с помощью программы тем или иным способом, в том или ином порядке. Его работа заключается в выполнении отдельных операций (согласно инструкции), то есть конкретных последовательностей действий, приводящих к конкретному результату (например, ежедневный запуск вспомогательных программ).

Состав типового руководства оператора

Так, например, требования к содержанию и оформлению Руководства оператора представлены в ГОСТ 19.505. В соответствии с требованиями стандарта, документ должен содержать следующие разделы:

– **Назначение программы**, где указывают область применения ПО и общие сведения о ней.

– **Условия выполнения программы**, где должны быть указаны условия, необходимые для работы ПО.

– **Выполнение программы**, где описывают последовательность действий оператора, обеспечивающих выполнение его обязанностей, а также ожидаемые реакции программы на эти действия.

– **Сообщения оператору**, где приводят тексты сообщений, выдаваемых в ходе выполнения программы, а также действия оператора в случае, если реакция программы не соответствует ожидаемой.

Такая структура документа обычно позволяет сделать его удобным, понятным и отвечающим тем задачам, которые необходимо решить с его помощью. Однако, кроме официальных требований, на основании практического опыта можно сформулировать несколько принципов создания Руководства оператора:

– не стоит включать в документ теоретические описания и отступления, лучше собрать всю теорию в отдельный раздел, а лучше, по возможности, обойтись без нее совсем;

– лучше не ссылаться в документе на какие-либо внешние или внутренние источники, информацию предпочтительнее продублировать в том месте, где она необходима;

– описывать нужно не только действия оператора, но и те результаты, которые он должен получить.

Стандарты для руководства оператора

Наличие Руководства оператора регламентируется ГОСТ 19.101, а структура и содержание – ГОСТ 19.505. Однако, в зависимости от сложности, назначения и области применения ПО, различные

Руководства оператора могут отличаться друг от друга по способу, методике и стилю изложения.

Стоимость разработки руководства оператора

Наименование документа	Наименование стандарта ГОСТ	Стоимость разработки	Срок выполнения	Пример выполнения
Руководство оператора на программное обеспечение	ГОСТ 19.505	040-160 тыс. руб.	до 1 месяца	Пример

Несмотря на все принципы и правила, у каждой системы всегда существуют особенности эксплуатации, которые нужно учесть в документации.

Введение

Руководство оператора должно состоять из следующих частей:

- Титульной;
- Информационной;
- Основной.

Титульная часть оформляется согласно ГОСТ 19.104-78 ЕСПД. Основные надписи.

Информационная часть должна состоять из аннотации и содержания. В аннотации приводят сведения о назначении документа и краткое изложение основной части.

Содержание включает перечень записей о структурных элементах основной части документа.

Основная часть руководства оператора должна содержать следующие разделы:

(ГОСТ 19.505-79 ЕСПД. Руководство оператора. Требования к содержанию и оформлению)

– **Назначение программы** содержит сведения о назначении программы и информацию, достаточную для понимания функций программы.

– **Условия выполнения программы** должны содержать минимальный и максимальный состав аппаратурных и программных средств.

– **Выполнение программы** представляет собой последовательность действий оператора, обеспечивающих загрузку, выполнение и завершение программы, возможные варианты команд, которыми оператор может управлять выполнением программы, а также ответы программы на эти команды.

– **Сообщения оператору** содержат тексты сообщений, выдаваемых в ходе выполнения программы и соответствующие действия оператора, его действия в случае сбоя, повторного запуска программы.

Задание

Составить руководство оператора в соответствии с ГОСТ 19.505-79 ЕСПД. Руководство оператора. Требования к содержанию и оформлению.

Требования к отчёту

Отчёт должен содержать титульный лист, аннотацию, содержание и основную часть, оформленную в соответствии с ГОСТ 19.505-79 ЕСПД. Руководство оператора. Требования к содержанию и оформлению.

ПРАКТИЧЕСКАЯ РАБОТА № 9 «Разработка моделей интерфейсов пользователей»

Цель работы: изучить процесс создания интерфейса пользователя.

Теоретические положения

Интерфейс пользователя является одним из важнейших элементов программы, это та часть программы, которая находится у всех на виду. Недочеты в пользовательском интерфейсе могут серьезно испортить впечатление даже о самых многофункциональных программах. Именно поэтому разработке и проектированию пользовательского интерфейса нужно уделять особое внимание.

Некоторые программисты склонны оставлять дизайн интерфейса пользователя на потом, считая, что реальное достоинство приложения его программный код, который и требует большего внимания. Однако часто возникает недовольство пользователей из-за неудачно подобранных шрифтов, непонятного содержимого экрана и скорости его прорисовывания, поэтому работу над интерфейсом также нужно воспринимать серьезно.

Для чего нужна разработка пользовательского интерфейса?

Как минимум, для этого есть две причины:

- чтобы пользователи работали более продуктивно, программа должна быть простой в использовании;
- хороший интерфейс может стать преимуществом против конкурентов, плохой послужит причиной неудачи всего проекта. Процесс создания интерфейса начинается с определения целей проекта, а также внутренних и внешних обстоятельств, которые вы должны принять во внимание. Для того чтобы правильно расставить приоритеты, необходимо учитывать:
 - опыт работы пользователей с компьютером, типовые ситуации использования;
 - какая информация необходима и когда, какие результаты должны быть получены;
 - технологию разработки и платформа, на которой будут работать пользователи.

1.1.1 Начальная фаза разработки. Концептуальный дизайн

В этой фазе разработки Вы должны решить, какой интерфейс лучше всего будет подходить для достижения ваших целей текстовый, графический или мультимедиа. Затем необходимо выбрать структуру взаимодействия, которая обеспечивает разные степени гибкости для пользователей. Обычно, чем гибче структура, тем больше она требует от пользователя обучения, понимания, и времени на работу с окнами (открыть, закрыть, разместить и т. д.).

Для выполнения начальной фазы разработки необходимо погрузиться целиком в задачи пользователей и создать бумажный прототип навигационной модели. Навигационная модель показывает, как необходимо распределять функции или задачи между окнами вашей программы, она определяет, как пользователи смогут перемещаться как между различными задачами, так и внутри отдельной задачи.

Для того чтобы создать хороший интерфейс, на каждой стадии разработки необходима обратная связь от пользователей. Чтобы оценить концептуальную модель программы, необходимо показать ее схему пользователям и попросить объяснить ее вам. Если у пользователя возникнут трудности, значит, вы еще не достигли точки зрения пользователя в понимании проблемы.

1.1.2 Визуальный дизайн. Использование компонентов

Хорошо выполненный дизайн выглядит чистым, простым и аккуратным. Его можно понять одним взглядом. Пользователь должен сразу распознавать, какие данные можно редактировать, какие нет; по каким объектам можно щелкать мышью и какие объекты можно перетаскивать.

Работа с несколькими формами. Если интерфейс пользователя должен содержать несколько форм, вам предстоит принять самое важное решение: какой использовать вид интерфейса одно документный (SDI) или многодокументный (MDI).

В SDI-приложениях окна форм появляются совершенно независимо друг от друга (примером таком интерфейса может служить программа «Блокнот (Notepad)» или графический редактор MS Paint), в MDI-приложениях вы можете одновременно работать с несколькими объектами (примером таком интерфейса может служить текстовый процессор MS Word). Однако не имеет значения какой тип интерфейса SDI или MDI выбран; взаимодействие пользователя с формами происходит одинаково посредством обработки событий, поступающих от элементов управления формы. Поэтому, если в вашем приложении предусмотрено несколько форм, программу необходимо написать так, чтобы у пользователей не было возможности нарушить предписанный ход ее выполнения (например, у пользователя не должно быть средств вывести форму, для которой еще не готова информация).

1.1.3 Эффективные меню

Еще одна важная часть разработки форм создание содержательных и эффективных меню. Приведем некоторые важные рекомендации:

- Следуйте стандартным соглашениям о расположении пунктов меню принятым в Windows File, Edit, View, и т. д.
- Группируйте пункты меню в логическом порядке и по содержанию.
- Для группировки пунктов в раскрывающихся меню используйте разделительные линии.
- Избегайте избыточных меню.
- Избегайте пунктов меню верхнего уровня, не содержащих раскрывающихся меню.
- Не забывайте использовать символ троеточия для обозначения пунктов меню, активизирующих диалоговые окна.
- Обязательно используйте клавиатурные эквиваленты команд и «горячие» клавиши.
- Помещайте на панель инструментов часто используемые команды меню.

Когда есть видимость работы приложения, пользователи более легко переносят длительное ожидание в работе программы.

Один из способов информирования пользователя о ходе выполнения работы использовать в форме индикатор процесса.

1.2 ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

Данное практическое занятие предполагает выполнение следующих этапов:

- изучить методические указания;
- ответить на контрольные вопросы.

1.3 КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Почему разработке пользовательского интерфейса уделяется особое внимание?
2. В чем заключается начальная стадия разработки интерфейса?
3. Рекомендации по созданию эффективного меню.
4. Особенности работы с несколькими формами.
5. Виды интерфейсов пользователей.

ПРАКТИЧЕСКАЯ РАБОТА №10 «Настройка доступа к сетевым устройствам»

Цели:

1. Изучение основных способов подключения и настройки соединения к сети Интернет в различных операционных системах
2. Продолжить формирование умений и навыков работы в сети Интернет
3. Формирование причинно-следственных связей

Форма отчета:

- выполнить задание;
- показать преподавателю;
- ответить на вопросы преподавателя.

Время выполнения: 2 ч

Материально техническое и методическое обеспечение

ПК, Интернет, методические указания по практической работе.

Краткие сведения из теории

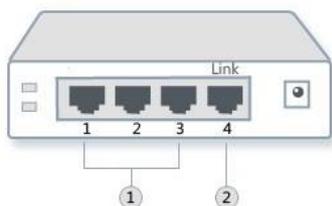
Сетевые технологии

Наиболее распространенными сетевыми технологиями являются беспроводная технология, Ethernet, HomePNA и Powerline.

Существует несколько видов оборудования, используемого в домашних сетях.

- **Сетевые адаптеры.** Эти адаптеры (также называемые сетевыми интерфейсными платами (NIC)) подключают компьютеры к сети, чтобы те могли обмениваться данными. Сетевой адаптер можно подключить к порту USB или Ethernet на компьютере или установить внутри компьютера в свободное гнездо расширения PCI.

- **Сетевые концентраторы и коммутаторы.** Концентраторы и коммутаторы подключают два или большее число компьютеров к сети Ethernet.



1 Порты для компьютеров

2 Порт для широкополосного модема или концентратора (обычно с меткой «Link» или «Out»)

Концентратор Ethernet

Порядок выполнения работы (занятия)

Задание 1. Настройка общего доступа к подключению интернета

Для того чтобы настроить общий доступ и подключение к сети интернет необходимо:

1. На сервере:

Войдите на сервер с учетной записью администратора или владельца. Нажмите кнопку Пуск и выберите пункт «Панель управления». Щелкните пункт «Сеть и подключения к Интернету». Щелкните ссылку «Сетевые подключения». Щелкните правой кнопкой мыши подключение, которое должно использоваться для выхода в интернет. Например, если доступ в Интернет осуществляется через модем, щелкните правой кнопкой мыши

требуемое подключение в разделе «Удаленный доступ». Нажмите кнопку «Свойства». Откройте вкладку «Дополнительно». В разделе Общий доступ к подключению Интернета установите флажок «Разрешить другим пользователям сети использовать подключение к Интернету данного компьютера».

При использовании удаленного общего подключения к Интернету установите флажок «Устанавливать вызов по требованию», чтобы разрешить компьютеру автоматическое подключение к Интернету. Нажмите кнопку «ОК».

Когда общий доступ к Интернету будет разрешен, сетевой плате локальной сети будет назначен IP-адрес 192.168.0.1. При этом связь с другими компьютерами сети может быть потеряна. Если другие компьютеры используют статические IP-адреса, следует настроить их на использование динамических адресов. Вы действительно хотите разрешить общий доступ к подключению Интернета? Нажмите «Да».

Для сетевой платы локальной сети устанавливается статический IP-адрес 192.168.0.1 и маска подсети 255.255.255.0.

2. На клиентском компьютере:

Для подключения к Интернету через общее соединение, необходимо проверить настройки IP для сетевой платы локальной сети и затем настроить клиентский компьютер. Для проверки настроек IP для сетевой платы локальной сети, выполните указанные ниже действия.

Войдите на клиентский компьютер с учетной записью администратора или владельца.

Нажмите кнопку «Пуск» и выберите пункт «Панель управления». Щелкните пункт «Сеть и подключения к Интернету».

Щелкните ссылку «Сетевые подключения».

Щелкните правой кнопкой мыши значок «Подключение по локальной сети» и выберите команду «Свойства».

На вкладке «Общие» выберите параметр «Протокол Интернета (TCP/IP)» в списке

«Компоненты, используемые этим подключением» и нажмите кнопку «Свойства». В диалоговом окне «Свойства»: Протокол Интернета (TCP/IP) выберите пункт

«Получить IP-адрес автоматически» (если он еще не выбран) и нажмите «ОК». Примечание. Можно также назначить уникальный статический IP-адрес в диапазоне от 192.168.0.2 до 192.168.0.254. Например, возможно назначение следующей комбинации статического IP-адреса, маски подсети и шлюза по умолчанию: IP-адрес 192.168.0.2 Маска подсети: 255.255.255.0

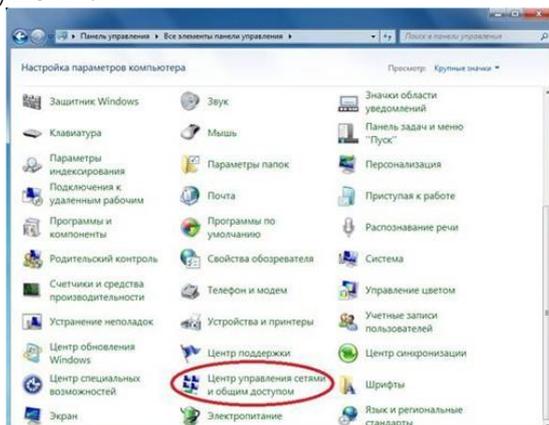
Шлюз по умолчанию: 192.168.0.1. В диалоговом окне Подключение по локальной сети – свойства нажмите кнопку «ОК». Закройте панель управления.

Задание 2. Настройка интернет на Windows 7

Прежде чем приступать к настройке интернет соединения, Вам необходимо установить драйвера на модем, сетевую карту или другое устройство, с помощью которого Вы осуществляете доступ в Интернет. Для начала Вам необходимо запустить Панель управления (Пуск и выбрать Панель управления):

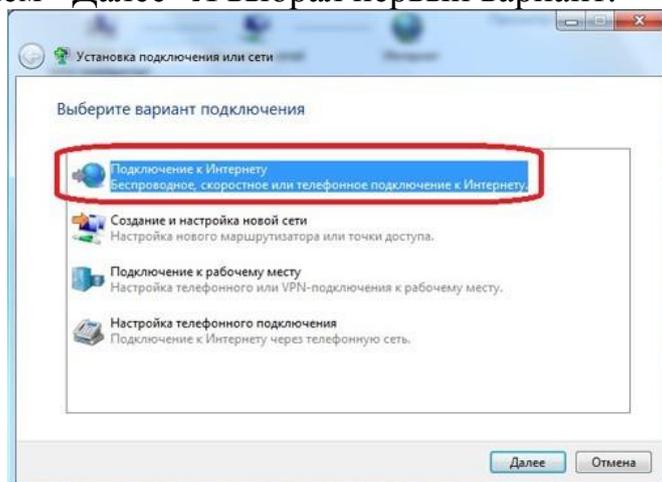
У вас появится окно Настройки параметров компьютера, в котором нужно переключить вид по категориям:

После этого в появившемся списке выберите "Центр управления сетями и общим доступом":



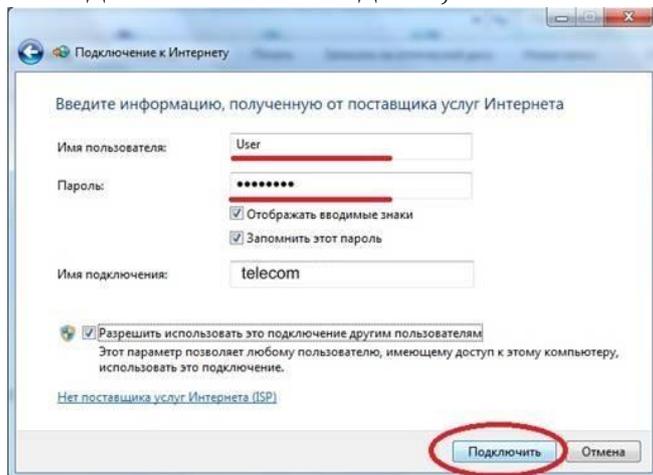
В "Центре управления сетями и общим доступом" Вам необходимо выбрать пункт "Настройка нового подключения или сети":

На следующем этапе установки нужно быть внимательным! Здесь операционная система предлагает нам выбрать вариант подключения к Интернету. В случае если Вы используете ADSL подключение, то Вам необходимо выбрать первый пункт: "Подключение к Интернету". Если Вы используете например 3G Интернет, то вам необходимо выбрать пункт "Настройка телефонного подключения". Выбираем необходимый Вам вариант и нажимаем "Далее" Я выбрал первый вариант:



В следующем окне нам нужно просто нажать на "Высокоскоростное (с PPPoE)" (Если Вы настраиваете 3G интернет, то на этой стадии у Вас появится окно выбора модема):

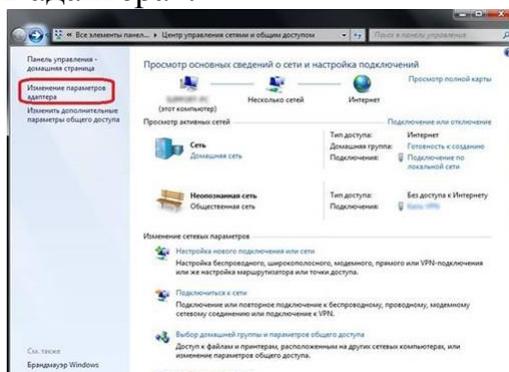
После этого необходимо ввести информацию от поставщика интернет услуг. Здесь пишем имя пользователя и пароль. Если Вы настраиваете 3G интернет, то у Вас будет ещё одно дополнительное поле "Набираемый номер". После того как данные были введены, нажимаем кнопку подключить:



Если все настроено правильно, то у вас появится надпись: "Подключение к Интернету готово к использованию":

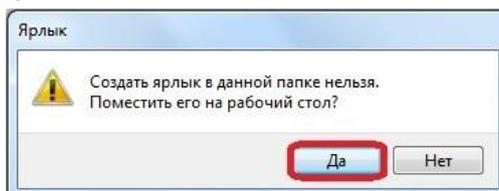
Следующим этапом **настройки интернета на Windows 7** будет создание ярлыка подключения на рабочем столе. Для этого опять переходим в панель управления -

>Центр управления сетями и общим доступом и кликаем по пункту "Изменение параметров адаптера":



В появившемся окне кликаем правой кнопкой мыши на созданном соединении и выбираем пункт "Создать ярлык":

Теперь кликаем на кнопку "Да" тем самым, подтверждая помещение ярлыка на рабочий стол:



ПРАКТИЧЕСКАЯ РАБОТА № 11

Настройка политики безопасности

Цель работы: ознакомиться с методами ограничения доступа к информации

Теоретическая часть

Разграничение доступа является достаточно эффективным средством предупреждения возможного ущерба вследствие нарушения целостности или конфиденциальности информации. В том случае, если доступ к самому компьютеру или к его ресурсам может получить пользователь, который имеет злой умысел или недостаточный уровень подготовки, он может случайно или преднамеренно исказить информацию или уничтожить ее полностью или частично.

Это же обстоятельство может привести к раскрытию закрытой информации или несанкционированному тиражированию открытой, например, программ, баз данных, разного рода документации, литературных произведений и т. д. в нарушение прав собственников информации, авторских прав...

С точки зрения разграничения доступа, в информационных системах следует различать *субъекты доступа* и *объекты доступа*. В число *субъектов доступа* могут войти либо персонал информационной системы, либо посторонние лица. *Объектами доступа* являются аппаратно-программные элементы информационных систем. Чаще всего в качестве объектов доступа рассматриваются файлы (в том числе папки и файлы программ). Доступ к объекту может рассматриваться либо как чтение (получение информации из него), либо как изменение (запись информации в него). Тогда виды доступа определяются следующими возможными сочетаниями этих операций:

- ни чтение, ни изменение;
- только чтение;
- только изменение;
- и чтение, и изменение.

Очевидно, что различие функциональных обязанностей субъектов обуславливает необходимость предоставления им соответствующих видов доступа.

Управление доступом пользователей и глобальными параметрами на членах домена осуществляется на двух уровнях: локальной системы и домена. На отдельных компьютерах доступ пользователей конфигурируют на уровне локальной системы, а одновременно для нескольких систем или ресурсов, входящих в домен, — на уровне домена.

Права доступа пользователя определяются руководителем организации и прописываются на рабочей станции системным администратором (администратором домена).

Процедура проверки прав доступа включает авторизацию и аутентификацию. Авторизация предполагает проверку уровня доступа к объекту, а аутентификация — проверку подлинности пользователя. Для аутентификации обычно используются имя пользователя (login) и пароль (password). Системный администратор осуществляет разграничение прав доступа в соответствии с заданной системной политикой, которая предполагает:

- ограничения на минимальную длину, сложность и срок действия пароля;
- требование уникальности паролей;
- блокировку пользователя при неудачной аутентификации;
- ограничение времени и места работы пользователя.

Система разграничения доступа реализована так, что при повседневной работе пользователи не должны замечать, что любое обращение к любому объекту проходит проверку на соответствие установленным правам доступа. Списки прав доступа можно задавать на каждый документ отдельно. При создании документа автоматически задается такой доступ на него, чтобы создатель имел все права.

Система разграничения доступа предназначена для реализации определенных администратором защиты правил на выполнение операций пользователями над объектами хранилища.

Система ограничения прав доступа не может дать полной гарантии безопасности информации. Дело в том, что злоумышленник может получить или подобрать пароль легального пользователя. Кроме того, опытный специалист может обойти систему разграничения доступа. Средствами обнаружения несанкционированного доступа к ресурсам служат системы аудита, которые автоматически фиксируют доступ к файлам и папкам и системные события.

Модели разграничения доступа

Наиболее распространенные модели разграничения доступа:

- *дискреционная* (избирательная) модель разграничения доступа;
- полномочная (*мандатная*) модель разграничения доступа.

Дискреционная модель характеризуется следующими правилами:

- ❖ любой объект имеет владельца;
- ❖ владелец имеет право произвольно ограничивать доступ субъектов к данному объекту;
- ❖ для каждого набора субъект – объект – метод право на доступ определен однозначно;
- ❖ наличие хотя бы одного привилегированного пользователя (например, администратора), который имеет возможность обращаться к любому объекту с помощью любого метода доступа.

В дискреционной модели определение прав доступа хранится в матрице доступа: в строках перечислены субъекты, а в столбцах – объекты. В каждой ячейке матрицы хранятся права доступа данного субъекта к данному объекту.

Тип метаданных	Объект синоним	Изменение	Чтение	Работа с внутренними документами	Изменение	Чтение	Регистрация внутренних документов	Изменение	Чтение	Чтение видов внутренних документов	Чтение
Справочники	Виды состояний документов в СВД		+								
Справочники	Виртуальные пользователи	+	+								
Справочники	Внешние информационные базы	+	+	+		+	+		+	+	+
Справочники	Внешние пользователи		+								
Справочники	Внутренние документы		+	+	+	+	+	+	+		
Справочники	Вопросы деятельности		+	+		+					
Справочники	Входящие документы		+								

Полномочная модель характеризуется следующими правилами:

- ❖ каждый объект обладает грифом секретности. Гриф секретности имеет числовое значение: чем оно больше, тем выше секретность объекта;
- ❖ у каждого субъекта доступа есть уровень допуска. Допуск к объекту в этой модели субъект получает только в случае, когда у субъекта значение уровня допуска не меньше значения грифа секретности объекта.

Преимущество полномочной модели состоит в отсутствии необходимости хранения больших объемов информации о разграничении доступа. Каждым субъектом выполняется хранение лишь значения своего уровня доступа, а каждым объектом – значения своего грифа секретности.

Методы разграничения доступа

Виды методов разграничения доступа:

1. Разграничение доступа по спискам

Суть метода состоит в задании соответствий: для каждого пользователя задается список ресурсов и права доступа к ним или для каждого ресурса определяется список пользователей и права доступа к этим ресурсам. С помощью списков возможно установление прав с точностью до каждого пользователя. Возможен вариант добавления прав или явного запрета доступа. Метод доступа по спискам используется в подсистемах безопасности операционных систем и систем управления базами данных.

2. Использование матрицы установления полномочий

При использовании матрицы установления полномочий применяется матрица доступа (таблица полномочий). В матрице доступа в строках

записываются идентификаторы субъектов, которые имеют доступ в компьютерную систему, а в столбцах – объекты (ресурсы) компьютерной системы. В каждой ячейке матрицы может содержаться имя и размер ресурса, право доступа (чтение, запись и др.), ссылка на другую информационную структуру, которая уточняет права доступа, ссылка на программу, которая управляет правами доступа и др. Данный метод является достаточно удобным, так как вся информация о полномочиях сохраняется в единой таблице. Недостаток матрицы – ее возможная громоздкость.

3. Разграничение доступа по уровням секретности и категориям

Разграничение по степени секретности разделяется на несколько уровней. Полномочия каждого пользователя могут быть заданы в соответствии с максимальным уровнем секретности, к которому он допущен. При разграничении по категориям задается и контролируется ранг категории пользователей. Таким образом, все ресурсы компьютерной системы разделены по уровням важности, причем каждому уровню соответствует категория пользователей.

4. Парольное разграничение доступа

Парольное разграничение использует методы доступа субъектов к объектам с помощью пароля. Постоянное использование паролей приводит к неудобствам для пользователей и временным задержкам. По этой причине методы парольного разграничения используются в исключительных ситуациях.

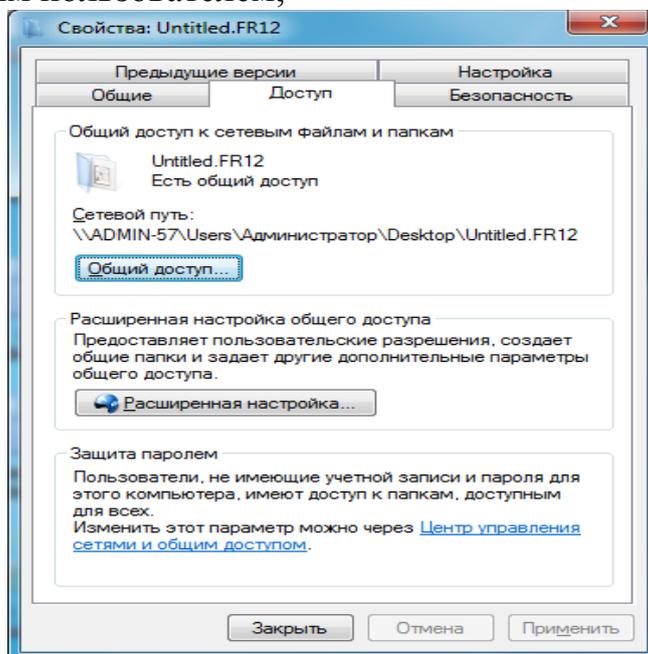
На практике принято сочетать разные методы разграничений доступа. Например, первые три метода усиливаются парольной защитой. Использование разграничения прав доступа является обязательным условием защищенной информационной системы.

Практическая часть – можно ничего не менять в своих настройках. Просто прогуляйтесь по указанным путям и сделайте соответствующие скрин-шоты.

1. Освоить средства разграничения доступа пользователей к папкам:

- выполнить команду «Общий доступ и безопасность» контекстного меню папки (если эта команда недоступна, то выключить режим «Использовать простой общий доступ к файлам» на вкладке «Вид» окна свойств папки) или команду «Свойства»;
- открыть вкладку «Безопасность» и включить в отчет сведения о субъектах, которым разрешен доступ к папке и о разрешенных для них видах доступа;
- с помощью кнопки «Дополнительно» открыть окно дополнительных параметров безопасности папки (вкладка «Разрешения»);

- включить в отчет сведения о полном наборе прав доступа к папке для каждого из имеющихся в списке субъектов;
- открыть вкладку «Владелец», включить в отчет сведения о владельце папки и о возможности его изменения обычным пользователем;
- открыть папку «Аудит», включить в отчет сведения о назначении параметров аудита, устанавливаемых на этой вкладке, и о возможности их установки обычным пользователем;



- закрыть окно дополнительных параметров безопасности.

2. Освоить средства разграничения доступа пользователей к файлам:

- выполнить команду «Свойства» контекстного меню файла;
- повторить все задания п. 1, но применительно не к папке, а к файлу.

3. Освоить средства разграничения доступа к принтерам:

- выполнить команду «Принтеры и факсы» меню «Пуск»;
- выполнить команду «Свойства» контекстного меню установленного в системе принтера;
- повторить все задания п. 1, но применительно не к папке, а к принтеру.

4. Освоить средства разграничения доступа к разделам реестра операционной системы:

- с помощью команды «Выполнить» меню «Пуск» запустить программу редактирования системного реестра regedit (regedt32);
- с помощью команды «Разрешения» меню «Правка» редактора реестра определить сведения о правах доступа пользователей к корневым разделам реестра, их владельцах и параметрах политики аудита;

- включить в отчет сведения о правах доступа пользователей к данной папке и о ее владельце.

Так как версии операционных систем у всех разные, то возможны отклонения от указанных маршрутов. Придется догадываться самим

Отчет должен содержать:

1. Тему и цель работы
2. Экранные копии выполнения работы
3. Составьте матрицу доступа по своей ИС.
4. Ответы на контрольные вопросы.

Контрольные вопросы:

1. В чем достоинства дискреционной политики безопасности?
2. В чем недостатки мандатной политики безопасности?
3. Кто определяет права доступа к папкам, файлам, принтерам при использовании дискреционной политики безопасности?
4. В чем отличие определения прав на доступ к файлам по сравнению с определением прав на доступ к папкам?
5. В чем отличие определения прав на доступ к принтерам по сравнению с определением прав на доступ к папкам и файлам?
6. В чем отличие определения прав на доступ к разделам реестра по сравнению с определением прав на доступ к папкам и файлам?

ПРАКТИЧЕСКАЯ РАБОТА №12

«Выполнение задач тестирования в процессе внедрения информационной системы»

Цель работы: изучить элементы плана тестирования; приобрести навыки разработки плана тестирования.

1.1 Краткие теоретические сведения

Тест-план, или план тестирования (test plan), – часть проектной документации, описывающая и регламентирующая процесс тестирования.

1.1.1 Ключевые секции тест-плана.

1.1.1.1 Перечень работ.

Эта секция включает перечень функциональных областей приложений, которые будут подвергаться тестированию.

Здесь же может быть перечень компонентов или функциональности, которые не будут тестироваться по тем или иным причинам.

1.1.1.2 Критерии качества.

Одна из наиболее важных секций, содержащая перечень критериев, по которым оценивается текущее и финальное качество продукта.

1.1.1.3 Оценка рисков.

Риск (risk) – сочетание вероятности наступления события и последствий, вызванных этим событием.

В секции приводится перечень рисков, которые могут (вполне вероятно) возникнуть в процессе работы с проектом. По каждому риску даётся оценка представляемой им угрозы и приводятся варианты выхода из ситуации.

1.1.1.4 Документация.

Здесь приводится полный перечень используемой документации, а также указывается, кто и когда должен её готовить, кому передавать и т.п.

1.1.1.5 Стратегия тестирования.

Стратегия тестирования подразумевает описание процесса тестирования с точки зрения применяемых методов, подходов, инструментальных средств и т. п.

1.1.1.6 Ресурсы.

В данной секции перечисляются необходимые для успешного проведения тестирования ресурсы:

- программные;
- аппаратные;
- человеческие;
- временные;
- финансовые.

1.1.1.7 Метрики.

Секция, в которой приводятся числовые характеристики показателей качества, способы их оценки, формулы и т. п.

1.1.1.8 Расписание и ключевые точки.

Самая простая для понимания секция. Фактически это календарь, в котором указано, что и к какому моменту должно быть сделано.

Задание

1 Приведите по пять примеров тестов для вашего разработанного ПО (или концепции вашего разработанного ПО)

2 Приведите пример риска, который, может быть, отмечен при планировании тестовых испытаний. Дайте рекомендации по недопущению такой ситуации и выходу из неё в случае возникновения.

3 Перечислите основные секции тестового плана и дайте краткое пояснение того, что размещается в каждой из них.

Контрольные вопросы:

1. Что подлежит тестированию?
2. Что такое тест-план?
3. Как определить критерии качества?
4. Что такое «риски» и как они оцениваются?

ЧАГАРОВА Зухра Нуржигидовна

ВНЕДРЕНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ

Практикум для обучающихся по специальности
09.02.07 Информационные системы и программирование

Корректор Чагова О.Х.
Редактор Чагова О.Х.

Сдано в набор 29.07.2024 г.
Формат 60x84/16
Бумага офсетная
Печать офсетная
Усл. печ. л. 3,02
Заказ № 4923
Тираж 100 экз.

Оригинал-макет подготовлен
в Библиотечно-издательском центре СКГА
369000, г. Черкесск, ул. Ставропольская 36