Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Умаров Марат Файзуллаевич
Должность: Директор
Дата подписания: 17.02.2026 15:42:52
Уникальный программный ключ:
48505f11ec15acaa386f5219d3113d727fefda78

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Казанский (Приволжский) федеральный университет»
Елабужский институт (филиал) КФУ

УТВЕРЖДАЮ
Заместитель директора по
образовательной деятельности
_____ С.Ю. Бахвалов
«19» 05 2025 г.
МП

Программа дисциплины (модуля)
Программная инженерия

Направление подготовки/специальность: 09.03.03 Прикладная информатика

Направленность (профиль) подготовки (специальности): Прикладная информатика в экономике

Квалификация: бакалавр

Форма обучения: очная

Язык обучения: русский

Год начала обучения по образовательной программе: - 2025

Содержание

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения ОПОП ВО

2. Место дисциплины (модуля) в структуре ОПОП ВО

3. Объем дисциплины (модуля) в зачетных единицах с указанием количества часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся

4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1. Структура и тематический план контактной и самостоятельной работы по дисциплине (модулю)

4.2. Содержание дисциплины (модуля)

5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

6. Фонд оценочных средств по дисциплине (модулю)

7. Перечень литературы, необходимой для освоения дисциплины (модуля)

8. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

9. Методические указания для обучающихся по освоению дисциплины (модуля)

10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости)

11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

12. Средства адаптации преподавания дисциплины (модуля) к потребностям обучающихся инвалидов и лиц с ограниченными возможностями здоровья

13. Приложение №1. Фонд оценочных средств

14. Приложение №2. Перечень литературы, необходимой для освоения дисциплины (модуля)

15. Приложение №3. Перечень информационных технологий, используемых для освоения дисциплины (модуля), включая перечень программного обеспечения и информационных справочных систем

Программу дисциплины разработал(а)(и) доцент, к.н. Минкин А.В. (Кафедра математики и прикладной информатики).

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения ОПОП ВО

Обучающийся, освоивший дисциплину (модуль), должен обладать следующими компетенциями:

Шифр компетенции	Расшифровка приобретаемой компетенции
ОПК-7	Способен разрабатывать алгоритмы и программы, пригодные для практического применения
ОПК-7.1	Знать технологии разработки алгоритмов и программ, пригодных для практического применения.
ОПК-7.2	Уметь разрабатывать алгоритмы и программы, пригодные для практического применения.
ОПК-7.3	Владеть способностью разрабатывать алгоритмы и программы, пригодные для практического применения

Обучающийся, освоивший дисциплину (модуль):

Должен знать:

рациональные технологии разработки алгоритмов и программ, пригодных для практического применения в будущей профессиональной деятельности, основные языки программирования и работы с базами данных.

Должен уметь:

самостоятельно разрабатывать алгоритмы и программы, пригодные для практического применения в будущей профессиональной деятельности, применять языки программирования и работы с базами данных.

Должен владеть:

способностью самостоятельно разрабатывать алгоритмы и программы, пригодные для практического применения в будущей профессиональной деятельности, навыками программирования.

2. Место дисциплины (модуля) в структуре ОПОП ВО

Данная дисциплина (модуль) включена в раздел "Б1.О.05.03 Дисциплины (модули)" основной профессиональной образовательной программы 09.03.03 "Прикладная информатика (Прикладная информатика в экономике)" и относится к обязательным дисциплинам.

Осваивается на 3 курсе в 5, 6 семестрах.

3. Объем дисциплины (модуля) в зачетных единицах с указанием количества часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся

Общая трудоемкость дисциплины составляет 6 зачетных(ые) единиц(ы) на 216 часа(ов).

Контактная работа - 90 часа(ов), в том числе лекции - 36 часа(ов), практические занятия - 0 часа(ов), лабораторные работы - 54 часа(ов), контроль самостоятельной работы - 0 часа(ов).

Самостоятельная работа - 90 часа(ов).

Контроль (зачёт / экзамен) - 36 часа(ов).

Форма промежуточного контроля дисциплины: зачет в 5 семестре; экзамен в 6 семестре.

4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1 Структура и тематический план контактной и самостоятельной работы по дисциплине (модулю)

N	Разделы дисциплины / модуля	Семестр	Виды и часы контактной работы, их трудоемкость (в часах)	Самостоятельная работа

			Лекции	Практические занятия	Лабораторные работы	
1.	Тема 1. Модели и профили жизненного цикла программных средств	5	6	0	6	12
2.	Тема 2. Модели и процессы управлении проектами программных средств.	5	6	0	6	12
3.	Тема 3. Управление требованиями к программному обеспечению	5	6	0	6	12
4.	Тема 4. Конструирование (детальное проектирование) программного обеспечения	6	6	0	12	18
5.	Тема 5. Инструменты и методы программной инженерии	6	6	0	12	18
6.	Тема 6. Качество программного обеспечения	6	6	0	12	18
	Итого: 216 ч. (из них 36 ч. контроль)		36	0	54	90

4.2 Содержание дисциплины (модуля)

Тема 1. Модели и профили жизненного цикла программных средств

Введение. Понятие программной инженерии. Программная инженерия в жизненном цикле программных систем. Модели и профили жизненного цикла программных средств.

Назначение профилей стандартов жизненного цикла в программной инженерии. Модель профиля стандартов жизненного цикла сложных программных систем

Тема 2. Модели и процессы управлении проектами программных средств.

Модели и процессы управлении проектами программных средств. Управление проектами программных средств в системе CMMI. Стандарты менеджмента качеством систем. Тестирование программного обеспечения. Принципы верификации и тестирования программ. Процессы и средства тестирования программных компонентов.

Тема 3. Управление требованиями к программному обеспечению

Организация разработки требований к сложным программным средствам. Процессы разработки требований к характеристикам сложных программных средств. Цели и принципы системного проектирования сложных программных средств. Процессы системного проектирования программных средств. Проектирование программных модулей и компонентов.

Тема 4. Конструирование (детальное проектирование) программного обеспечения

Задачи и особенности объектно-ориентированного проектирования программных средств. Основные понятия и модели объектно-ориентированного проектирования. Процессы тестирования структуры компонентов. Сопровождение программного обеспечения. Организация и методы сопровождения программных средств. Этапы и процедуры при сопровождении программных средств.

Тема 5. Инструменты и методы программной инженерии

Инструменты для поддержки процессов жизненного цикла. Эвристические, формальные и методы прототипирования. Процессы сертификации в жизненном цикле программных средств. Организация сертификации программных продуктов. Документирование процессов и результатов документирования программных продуктов.

Тема 6. Качество программного обеспечения

Организация документирования программных средств. Формирование требований к документации сложных программных средств. Планирование документирования проектов сложных программных средств. Технико-экономическое обоснование проектов программных средств. Цели и процессы технико-экономического обоснования проектов программных средств. Применение различных методик для технико-экономического обоснования.

5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

Самостоятельная работа обучающихся выполняется по заданию и при методическом руководстве преподавателя, но без его непосредственного участия. Самостоятельная работа подразделяется на самостоятельную работу на аудиторных занятиях и на внеаудиторную самостоятельную работу. Самостоятельная работа обучающихся включает как полностью самостоятельное освоение отдельных тем (разделов) дисциплины, так и проработку тем (разделов), осваиваемых во время аудиторной работы. Во время самостоятельной работы

обучающиеся читают и конспектируют учебную, научную и справочную литературу, выполняют задания, направленные на закрепление знаний и отработку умений и навыков, готовятся к текущему и промежуточному контролю по дисциплине.

Организация самостоятельной работы обучающихся регламентируется нормативными документами, учебно-методической литературой и электронными образовательными ресурсами, включая:

Порядок организации и осуществления образовательной деятельности по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры (утвержденный приказом Министерства науки и высшего образования Российской Федерации от 6 апреля 2021 года № 245)

Устав федерального государственного автономного образовательного учреждения "Казанский (Приволжский) федеральный университет"

Правила внутреннего распорядка федерального государственного автономного образовательного учреждения высшего профессионального образования "Казанский (Приволжский) федеральный университет"

Локальные нормативные акты Казанского (Приволжского) федерального университета

6. Фонд оценочных средств по дисциплине (модулю)

Фонд оценочных средств по дисциплине (модулю) включает оценочные материалы, направленные на проверку освоения компетенций, в том числе знаний, умений и навыков. Фонд оценочных средств включает оценочные средства текущего контроля и оценочные средства промежуточной аттестации.

В фонде оценочных средств содержится следующая информация:

- соответствие компетенций планируемым результатам обучения по дисциплине (модулю);

- критерии оценивания сформированности компетенций;

- механизм формирования оценки по дисциплине (модулю);

- описание порядка применения и процедуры оценивания для каждого оценочного средства;

- критерии оценивания для каждого оценочного средства;

- содержание оценочных средств, включая требования, предъявляемые к действиям обучающихся, демонстрируемым результатам, задания различных типов.

Фонд оценочных средств по дисциплине находится в Приложении 1 к программе дисциплины (модулю).

7. Перечень литературы, необходимой для освоения дисциплины (модуля)

Освоение дисциплины (модуля) предполагает изучение учебной литературы. Литература может быть доступна обучающимся в одном из двух вариантов (либо в обоих из них):

- в электронном виде - через электронные библиотечные системы на основании заключенных КФУ договоров с правообладателями;

- в печатном виде - в Научной библиотеке Елабужского института КФУ. Обучающиеся получают учебную литературу на абонементе по читательским билетам в соответствии с правилами пользования Научной библиотекой.

Электронные издания доступны дистанционно из любой точки при введении обучающимся своего логина и пароля от личного кабинета в системе "Электронный университет". При использовании печатных изданий библиотечный фонд должен быть укомплектован ими из расчета не менее 0,25 экземпляра на каждого обучающегося из числа лиц, одновременно осваивающих данную дисциплину

Перечень литературы, необходимой для освоения дисциплины (модуля), находится в Приложении 2 к рабочей программе дисциплины. Он подлежит обновлению при изменении условий договоров КФУ с правообладателями электронных изданий и при изменении комплектования фондов Научной библиотеки Елабужского института КФУ.

8. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

Введение в программную инженерию - https://www.intuit.ru/studies/courses/497/353/info

Основы менеджмента программных проектов - https://www.intuit.ru/studies/courses/38/38/info

Основы тестирования программного обеспечения - https://www.intuit.ru/studies/courses/48/48/info

9. Методические указания для обучающихся по освоению дисциплины (модуля)

Вид работ	Методические рекомендации
лекции	При изучении дисциплины сначала необходимо по каждой теме лекции прочитать

Вид работ	Методические рекомендации
	рекомендованную литературу и составить краткий конспект основных положений, терминов, сведений, требующих запоминания и являющихся основополагающими в этой теме для освоения последующих тем курса. Для расширения знания по дисциплине рекомендуется использовать Интернет-ресурсы; проводить поиски в различных системах и использовать материалы сайтов, рекомендованных преподавателем.
лабораторные работы	При подготовке к лабораторным занятиям необходимо заранее изучить методические рекомендации по его проведению. Обратить внимание на цель работы, на основные вопросы для подготовки к работе, на содержание темы работы. Лабораторное занятие проходит в виде диалога, разбора основных вопросов темы. Также лабораторное занятие может проходить в виде показа презентаций, демонстративного материала (в частности плакатов, слайдов), которые сопровождаются беседой преподавателя со студентами. Студент может сдавать лабораторную работу в виде написания реферата, подготовки слайдов, презентаций и последующей защиты его, либо может написать конспект в тетради, ответив на вопросы по заданной теме. Ответы на вопросы можно сопровождать рисунками, схемами и т.д. с привлечением дополнительной литературы, которую следует указать.
самостоя-тельная работа	Обучающийся самостоятельно определяет режим своей самостоятельной работы и меру труда, затрачиваемого на овладение знаниями и умениями по дисциплине, выполняет внеаудиторную работу по индивидуальному плану, в зависимости от собственной подготовки, бюджета времени и других условий. Ежедневно обучающийся должен уделять выполнению самостоятельной работы в среднем не менее 3 часов. При выполнении самостоятельной работы обучающийся имеет право обращаться к преподавателю за консультацией с целью уточнения задания, формы контроля выполненного задания.
зачет	Для контроля усвоения данной дисциплины предусмотрен зачет, на котором студентам необходимо ответить на вопросы зачетных билетов. При ответе на зачете необходимо: продумать и четко изложить материал; дать определение основных понятий; дать краткое описание явлений; привести примеры. Ответ следует иллюстрировать схемами, рисунками и графиками.
экзамен	Для контроля усвоения данной дисциплины предусмотрен экзамен, на котором студентам необходимо ответить на вопросы экзаменационных билетов. При ответе на экзамене необходимо: продумать и четко изложить материал; дать определение основных понятий; дать краткое описание явлений; привести примеры. Ответ следует иллюстрировать схемами, рисунками и графиками.

10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости)

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем, представлен в Приложении 3 к рабочей программе дисциплины (модуля).

11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Учебная аудитория № 60 (423600, Республика Татарстан, г. Елабуга, ул. Казанская, д. 89) для проведения занятий лекционного типа, занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, помещение для самостоятельной работы. Комплект мебели (посадочных мест) 29 шт. Комплект мебели (посадочных мест) для преподавателя 1 шт. Компьютерный класс: Компьютеры intel core i5 15 шт. Мониторы ViewSonic 22d 15 шт. Проектор EPSON EB-535W 1 шт. Интерактивная доска IQBoard DVT TN082 1 шт. Трибуна 1 шт. Кондиционер 1 шт. Настенные пол ки 6 шт. Шкаф двухстворчатый с полками 1 шт. Веб-камера 1 шт. Выход в Интернет, внутривузовская компьютерная сеть, доступ в электронную информационно-образовательную среду. Набор учебно-наглядных пособий: комплект презентаций в электронном формате по преподаваемой дисциплине 3-5 шт.

12. Средства адаптации преподавания дисциплины к потребностям обучающихся инвалидов и лиц с ограниченными возможностями здоровья

При необходимости в образовательном процессе применяются следующие методы и технологии, облегчающие восприятие информации обучающимися инвалидами и лицами с ограниченными возможностями

здоровья:

- создание текстовой версии любого нетекстового контента для его возможного преобразования в альтернативные формы, удобные для различных пользователей;

- создание контента, который можно представить в различных видах без потери данных или структуры, предусмотреть возможность масштабирования текста и изображений без потери качества, предусмотреть доступность управления контентом с клавиатуры;

- создание возможностей для обучающихся воспринимать одну и ту же информацию из разных источников - например, так, чтобы лица с нарушениями слуха получали информацию визуально, с нарушениями зрения - аудиально;

- применение программных средств, обеспечивающих возможность освоения навыков и умений, формируемых дисциплиной, за счёт альтернативных способов, в том числе виртуальных лабораторий и симуляционных технологий;

- применение дистанционных образовательных технологий для передачи информации, организации различных форм интерактивной контактной работы обучающегося с преподавателем, в том числе вебинаров, которые могут быть использованы для проведения виртуальных лекций с возможностью взаимодействия всех участников дистанционного обучения, проведения семинаров, выступления с докладами и защиты выполненных работ, проведения тренингов, организации коллективной работы;

- применение дистанционных образовательных технологий для организации форм текущего и промежуточного контроля;

- увеличение продолжительности сдачи обучающимся инвалидом или лицом с ограниченными возможностями здоровья форм промежуточной аттестации по отношению к установленной продолжительности их сдачи:

- продолжительности сдачи зачёта или экзамена, проводимого в письменной форме, - не более чем на 90 минут;

- продолжительности подготовки обучающегося к ответу на зачёте или экзамене, проводимом в устной форме, - не более чем на 20 минут;

- продолжительности выступления обучающегося при защите курсовой работы - не более чем на 15 минут.

Программа составлена в соответствии с требованиями ФГОС ВО и учебным планом по направлению 09.03.03 "Прикладная информатика" и профилю подготовки "Прикладная информатика в экономике".

Приложение №1
к рабочей программе дисциплины (модуля)
Б1.О.05.03 Программная инженерия

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
"Казанский (Приволжский) федеральный университет"
Елабужский институт (филиал)

Фонд оценочных средств по дисциплине
Б1.О.05.03 Программная инженерия

Направление подготовки: 09.03.03 - Прикладная информатика
Профиль подготовки: Прикладная информатика в экономике
Квалификация выпускника: бакалавр
Форма обучения: очное
Язык обучения: русский
Год начала обучения по образовательной программе: 2025

СОДЕРЖАНИЕ

1. Соответствие компетенций планируемым результатам обучения по дисциплине (модулю)
2. Критерии оценивания сформированности компетенций
3. Распределение оценок за формы текущего контроля и промежуточную аттестацию
4. Оценочные средства, порядок их применения и критерии оценивания
4.1. Оценочные средства текущего контроля
4.1.1. Лабораторные работы
4.1.1.1. Порядок проведения.
4.1.1.2 Критерии оценивания
4.1.1.3. Содержание оценочного средства
4.1.2. Тестирование
4.1.2.1. Порядок проведения.
4.1.2.2 Критерии оценивания
4.1.2.3. Содержание оценочного средства
4.2. Оценочные средства промежуточной аттестации
Зачет, экзамен
4.2.1. Устный или письменный ответ на вопрос
4.2.1.1. Порядок проведения.
4.2.1.2. Критерии оценивания.
4.2.1.3. Оценочные средства.
4.2.2. Практическое задание
4.2.2.1. Порядок проведения.
4.2.2.2. Критерии оценивания.
4.2.2.3. Оценочные средства.

1. Соответствие компетенций планируемым результатам обучения по дисциплине (модулю)

Код и наименование компетенции	Индикаторы достижения компетенций для данной дисциплины	Оценочные средства текущего контроля и промежуточной аттестации
ОПК-7. Способен разрабатывать алгоритмы и программы, пригодные для практического применения	Знает рациональные технологии разработки алгоритмов и программ, пригодных для практического применения в будущей профессиональной деятельности, основные языки программирования и работы с базами данных. Умеет самостоятельно разрабатывать алгоритмы и программы, пригодные для практического применения в будущей профессиональной деятельности, применять языки программирования и работы с базами данных. Владеет способностью самостоятельно разрабатывать алгоритмы и программы, пригодные для практического применения в будущей профессиональной деятельности, навыками программирования.	Текущий контроль: Лабораторные работы по темам Тема 1. Модели и профили жизненного цикла программных средств Тема 2. Модели и процессы управлении проектами программных средств. Тема 3. Управление требованиями к программному обеспечению Тема 4. Конструирование (детальное проектирование) программного обеспечения Тема 5. Инструменты и методы программной инженерии Тема 6. Качество программного обеспечения Тестирование по темам Тема 1. Модели и профили жизненного цикла программных средств Тема 2. Модели и процессы управлении проектами программных средств. Тема 3. Управление требованиями к программному обеспечению Тема 4. Конструирование (детальное проектирование) программного обеспечения Тема 5. Инструменты и методы программной инженерии Тема 6. Качество программного обеспечения Промежуточная аттестация: *Зачет, Экзамен*

2. Критерии оценивания сформированности компетенций

Компетенция	Зачтено			Не зачтено
	Высокий уровень (отлично) (86-100 баллов)	**Средний уровень (хорошо) (71-85 баллов)**	**Низкий уровень (удовлетворительно) (56-70 баллов)**	**Ниже порогового уровня (неудовлетворительно) (0-55 баллов)**
ОПК-7	Знает рациональные технологии разработки алгоритмов и программ, пригодных для практического применения в будущей профессиональной деятельности, основные языки программирования и работы с базами данных	Знает основные рациональные технологии разработки алгоритмов и программ, пригодных для практического применения в будущей профессиональной деятельности, основные языки программирования и работы с базами данных. Допускает незначительные ошибки при ответе на вопрос или решении поставленной задачи	Знает отдельные рациональные технологии разработки алгоритмов и программ, пригодных для практического применения в будущей профессиональной деятельности, основные языки программирования и работы с базами данных. Допускает типичные ошибки при ответе на вопрос или решении поставленной задачи	Не знает рациональные технологии разработки алгоритмов и программ, пригодных для практического применения в будущей профессиональной деятельности, основные языки программирования и работы с базами данных
	Умеет самостоятельно разрабатывать алгоритмы и программы, пригодные для практического	Умеет самостоятельно разрабатывать алгоритмы и программы, пригодные для практического	Умеет самостоятельно разрабатывать алгоритмы и программы, пригодные для практического	Не умеет самостоятельно разрабатывать алгоритмы и программы, пригодные

	применения в будущей профессиональной деятельности, применять языки программирования и работы с базами данных	применения в будущей профессиональной деятельности, применять языки программирования и работы с базами данных. Допускает незначительные ошибки при ответе на вопрос или решении поставленной задачи	применения в будущей профессиональной деятельности, применять языки программирования и работы с базами данных. Допускает типичные ошибки при ответе на вопрос или решении поставленной задачи	для практического применения в будущей профессиональной деятельности, применять языки программирования и работы с базами данных
	Владеет способностью самостоятельно разрабатывать алгоритмы и программы, пригодные для практического применения в будущей профессиональной деятельности, навыками программирования	Владеет способностью самостоятельно разрабатывать алгоритмы и программы, пригодные для практического применения в будущей профессиональной деятельности, навыками программирования. Допускает незначительные ошибки при ответе на вопрос или решении поставленной задачи	Владеет способностью самостоятельно разрабатывать алгоритмы и программы, пригодные для практического применения в будущей профессиональной деятельности, навыками программирования. Допускает типичные ошибки при ответе на вопрос или решении поставленной задачи	Не владеет способностью самостоятельно разрабатывать алгоритмы и программы, пригодные для практического применения в будущей профессиональной деятельности, навыками программирования.

3. Распределение оценок за формы текущего контроля и промежуточную аттестацию

5 семестр:
Текущий контроль:
Лабораторные работы по темам
Тема 1. Модели и профили жизненного цикла программных средств
Тема 2. Модели и процессы управлении проектами программных средств.
Тема 3. Управление требованиями к программному обеспечению
Максимальное количество баллов по БРС - 30.

Тестирование
Тема 1. Модели и профили жизненного цикла программных средств
Тема 2. Модели и процессы управлении проектами программных средств.
Тема 3. Управление требованиями к программному обеспечению
Максимальное количество баллов по БРС - 20.

Итого 30+20=50 баллов

6 семестр

Текущий контроль:
Лабораторные работы по темам
Тема 4. Конструирование (детальное проектирование) программного обеспечения
Тема 5. Инструменты и методы программной инженерии
Тема 6. Качество программного обеспечения
Максимальное количество баллов по БРС - 30.

Тестирование
Тема 4. Конструирование (детальное проектирование) программного обеспечения
Тема 5. Инструменты и методы программной инженерии
Тема 6. Качество программного обеспечения
Максимальное количество баллов по БРС - 20.

Итого 30+20=50 баллов
Промежуточная аттестация - зачет – 50 баллов, экзамен-50 баллов.

Промежуточная аттестация проводится после завершения изучения дисциплины или ее части в форме, определяемой учебным планом образовательной программы с целью оценить работу обучающегося, степень усвоения теоретических знаний, уровень сформированности компетенций.
По дисциплине предусмотрен зачет в 5 семестре, экзамен в 6 семестре. Преподаватель, принимающий зачет/экзамен обеспечивает случайное распределение вариантов зачетных/экзаменационных заданий между обучающимися с помощью билетов и/или с применением компьютерных технологий; вправе задавать обучающемуся дополнительные вопросы и давать дополнительные задания помимо тех, которые указаны в билете.
Зачет/экзамен проводится по билетам. В каждом билете два оценочных средства: устный или письменный ответ на вопрос и практическое задание.

Устный или письменный ответ – 20 баллов.
Практическое задание – 30 баллов.
Итого 20+30=50 баллов.
Общее количество баллов по дисциплине за текущий контроль и промежуточную аттестацию: 50+50=100 баллов.

Виды оценок:
Для экзамена
86-100 – отлично.
71-85 – хорошо.
56-70 – удовлетворительно.
0-55 – неудовлетворительно.

Для зачета:
56-100 – зачтено.
0-55 – не зачтено.

4. Оценочные средства, порядок их применения и критерии оценивания
4.1. Оценочные средства текущего контроля
4.1.1. Лабораторные работы
Тема 1. Модели и профили жизненного цикла программных средств
Тема 2. Модели и процессы управлении проектами программных средств.
Тема 3. Управление требованиями к программному обеспечению
Тема 4. Конструирование (детальное проектирование) программного обеспечения
Тема 5. Инструменты и методы программной инженерии
Тема 6. Качество программного обеспечения
4.1.1.1. Порядок проведения.
Лабораторные работы проводятся в часы аудиторной работы.

Перед выполнением каждой работы студенты-бакалавры должны проработать соответствующий материал, используя конспекты теоретических занятий, периодические издания, учебно-методические пособия и учебники.

По окончании занятий студенты оформляют отчет по каждой работе, соблюдая следующую форму:

- Наименование темы;
- Цель работы;
- Задание и содержание выполненной работы,
- Письменные ответы на контрольные вопросы.
- Выводы по проделанной работе.
- Список использованных источников.

4.1.1.2 Критерии оценивания
27-30 баллов ставится, если обучающийся:
Правильно выполнил все задания. Продемонстрировал высокий уровень владения материалом. Проявлены превосходные способности применять знания и умения к выполнению конкретных заданий.
22-26 баллов ставится, если обучающийся:
Правильно выполнил большую часть заданий. Присутствуют незначительные ошибки. Продемонстрирован хороший уровень владения материалом. Проявлены средние способности применять знания и умения к выполнению конкретных заданий.
18-21 баллов ставится, если обучающийся:
Задания выполнил более чем наполовину. Присутствуют серьёзные ошибки. Продемонстрирован удовлетворительный уровень владения материалом. Проявлены низкие способности применять знания и умения к выполнению конкретных заданий.
0-17 баллов ставится, если обучающийся:

Задания выполнил менее чем наполовину. Продемонстрирован неудовлетворительный уровень владения материалом. Проявлены недостаточные способности применять знания и умения к выполнению конкретных заданий.

4.1.1.3. Содержание оценочного средства

Практикум состоит из пяти шагов.

Обзор тем и задач

Лабораторная работа 1. При изучении первой темы (на первой серии занятий) студенты организуются как Scrum-*команда*. Они также знакомятся с условиями модельной задачи, настраивают *инфраструктуру* TFS для будущей разработки (создают командный проект и распределяют *права* на работу с ним).

Лабораторная работа 2. На второй серии занятий студенты практикуются в планировании *работ* на основе методологии Scrum, а также изучают способы использования системы отслеживания задач TFS. Студентам необходимо импортировать *список* пользовательских историй их файла *Excel* в TFS, а затем детально спланировать будущий спринт и распределить задачи.

Лабораторная работа 3. Третья серия занятий предполагает основную работу *по* реализации решения модельной задачи. На занятиях этой серии студенты должны также освоится с *системой контроля версий* TFS, её *интеграцией* с системой отслеживания задач, а также попрактиковаться в создании ветвей и *интеграции* изменений.

Лабораторная работа 4. На занятиях четвертой серии студенты практикуются в разработки *модульных* тестов средствами *Visual Studio Team Developer*. На этих занятиях студенты должны освоить средства автоматической генерации тестов, наполнить сгенерированные тесты содержимым, а также научится изменять конфигурацию запуска *модульных* тестов и считать тестовое покрытие.

Лабораторная работа 5. Пятая серия занятий посвящена системе автоматических сборок TFS. На этих занятиях студенты должны создать несколько определений для автоматической *сборки* (*build definitions*) в разных случаях – с тестами и без, с анализом кода и без и т.д. Кроме того, студенты должны настроить параметры непрерывной *интеграции* и рассылки уведомлений.

Лабораторная работа 6. На заключительной, шестой, серии занятий студенты должны повести ретроспективный *анализ* выполненного Scrum sprint, выявить потенциальные способы *оптимизации*, а затем и применить их, используя средства настройки процесса разработки TFS. На этих занятиях студенты должны освоить изменение настроек системы остлеживания задач средствами *Team Foundation Server Power Tools*.

Лабораторная работа 1. Знакомство и создание проекта

Целями данного занятия является следующее.

1. Разделить студентов на scrum-команды, определить и обсудить Scrum-роли.
2. Провести начальное знакомство с модельной задачей, которую предстоит реализовать.
3. Прояснить открытые вопросы по модельной задаче с Product *Owner*.
4. Создать командный проект на TFS и добавить в него пользователей.

Для разделения студентов на scrum-команды и выделения Scrum-мастеров можно прибегнуть к жеребьевке, после чего перейти к знакомству с задачей. Для этого руководитель курса должен заранее приготовить следующие документы:

- Краткое описание основной концепции разрабатываемого приложения.
- Список задач в формате *product* backlog.

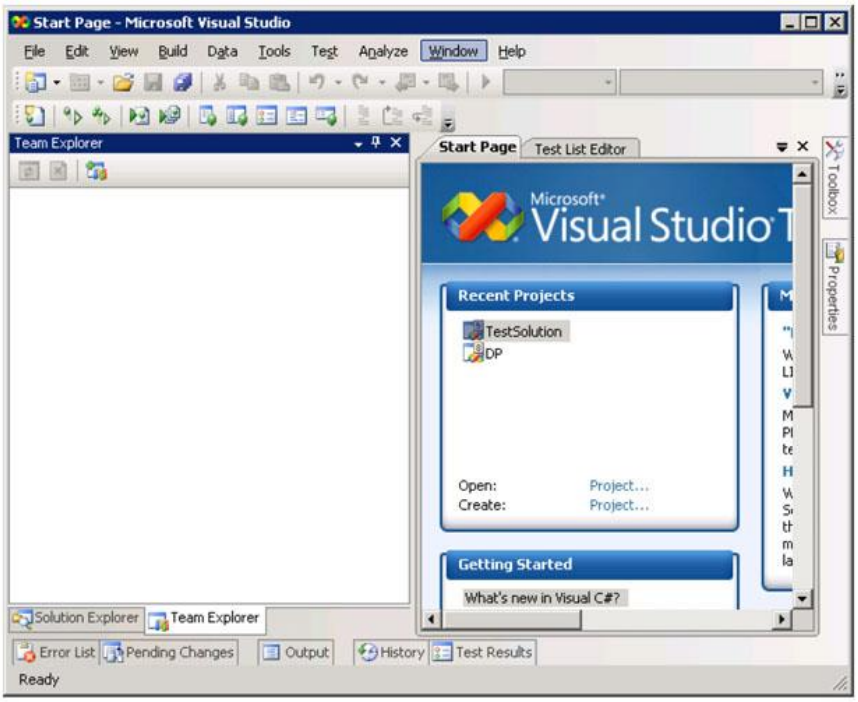
Команды получают эти документы и в течение 20-30 минут обсуждают их в рамках команды, готовя вопросы к хозяину продукта. Затем, в течение 20-30 минут хозяин продукта отвечает на подготовленные командами вопросы.

После того, как команды получили *представление* о разрабатываемом продукте (модельной задаче), им необходимо создать командный проект в MS VSTS и занести всех участников проекта в *список* пользователей.

Шаг 1. Создание проекта

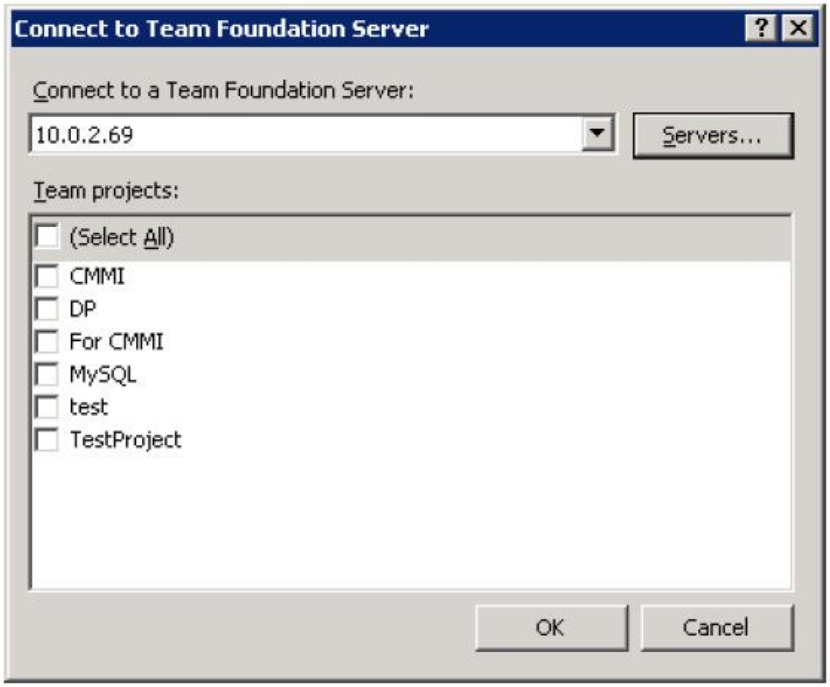
Создание командного проекта осуществляется лидером команды *по* следующему сценарию:

1. Открыть *Visual Studio Team Suite* и окно *Team Explorer* в ней:



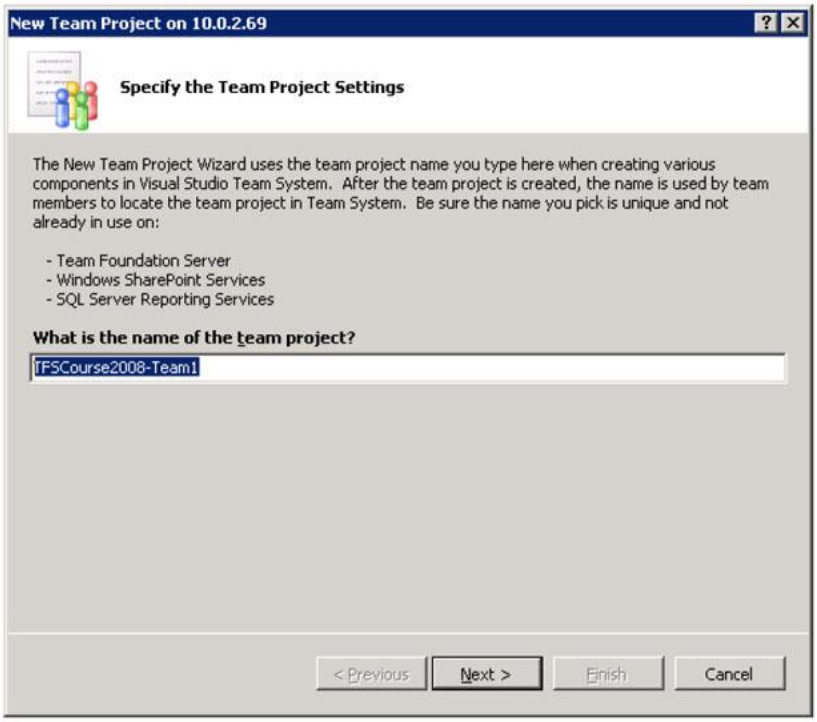
2. Нажать кнопку соединится с сервером

3. Задать имя и *порт* сервера в открывшееся диалоге:

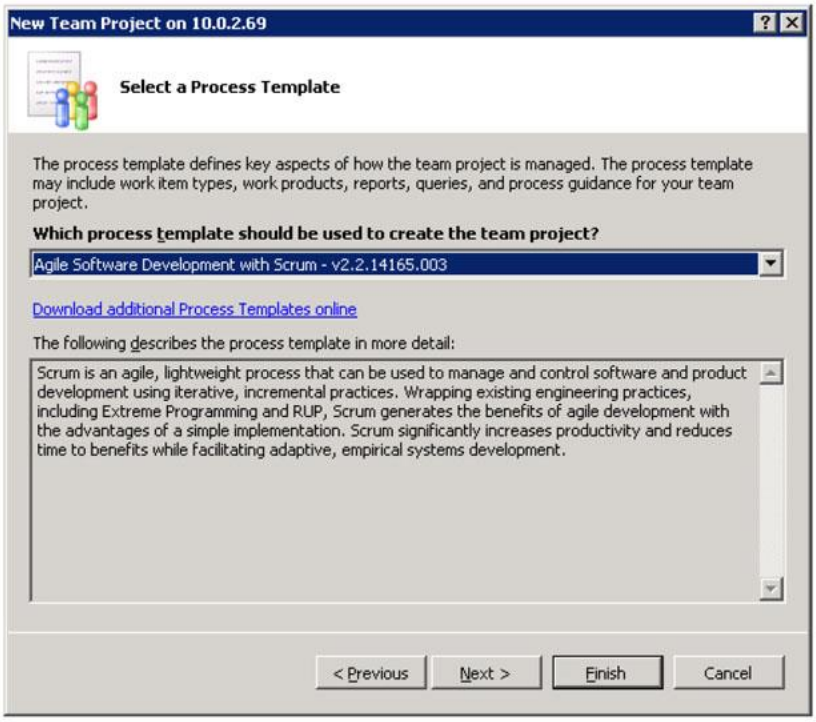


4. После того, как соединение с сервером установлено, запустить процедуру создания проекта, используя соответствующую команду контекстного меню (**New Team Project…**):

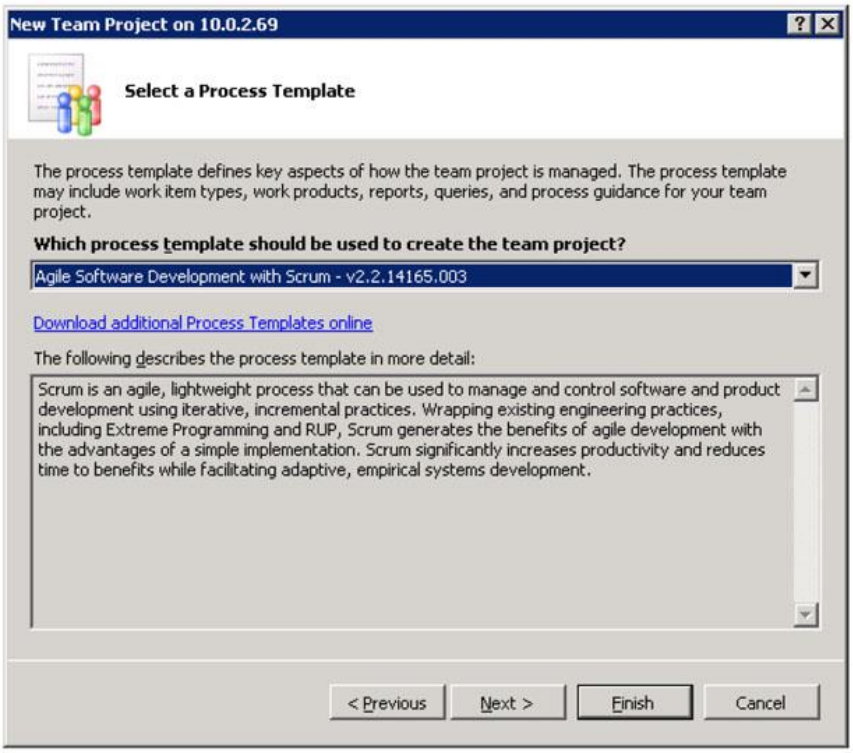
5. В качестве имени проекта необходимо указать **TFSCourse<год>-<имя команды>**:



6. В качестве шаблона процесса разработки выбрать Scrum:



7. Создать новый пустой раздел в *системе контроля версий* для данного проекта:

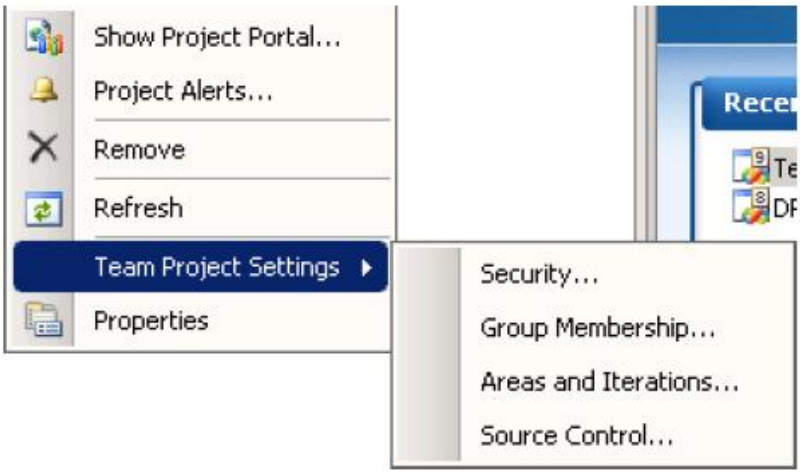


8. После выбора всех настроек, создать проект.

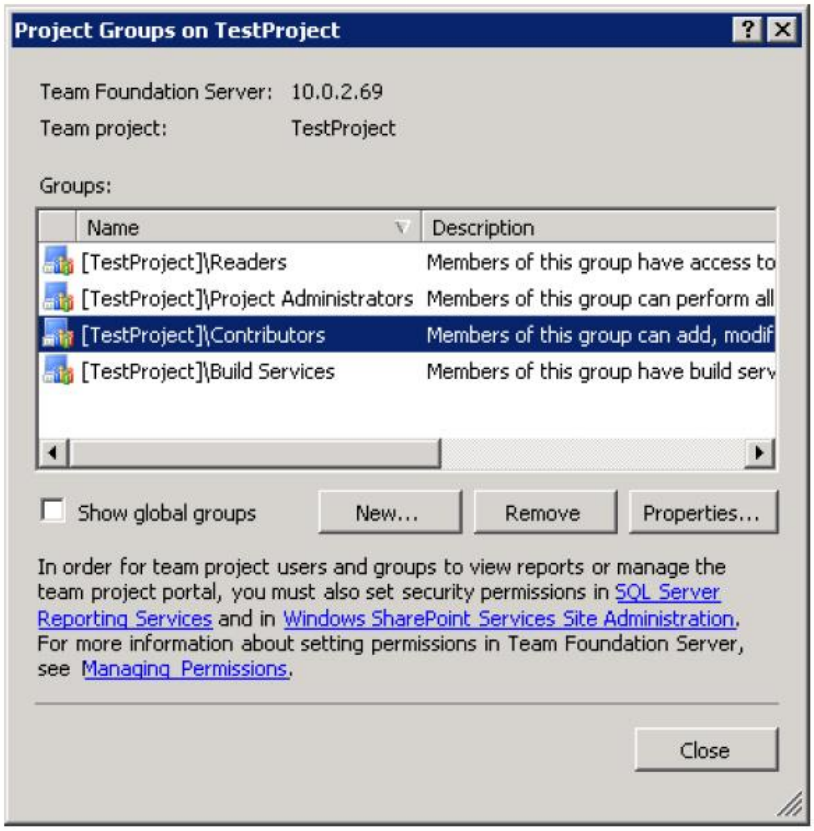
Шаг 2. Настройка прав

После того, как проект был создан лидеру необходимо выделить *права* остальным участникам команды для работы с этим проектом. Для этого ему нужно:

1. Выбрать в свойствах проекта раздел **Group membership**:



2. В открывшемся диалоге включить всех участников в группы **Readers** и **Contributors**:



Шаг 3. Подключение проекта остальными участниками команды

После того, как все получили *права* на работу с проектом, каждый *участник команды* должен на своей машине открыть *Visual Studio* и добавить соединение с этим проектом. Выполняется это аналогично пунктам 1-4 первого шага занятия.

Лабораторная работа 2. Работа с системой отслеживания ошибок

Основной целью данного занятия является знакомство участников с системой отслеживания элементов работы.

1. Создание элементов работы средствами *Visual Studio* и *Team Explorer*.
2. Импорт и экспорт элементов работы из/в Mcrosoft *Excel*.
3. Назначение ответственных за элементы работы.

4. Отслеживание текущего статуса посредством отчетов.

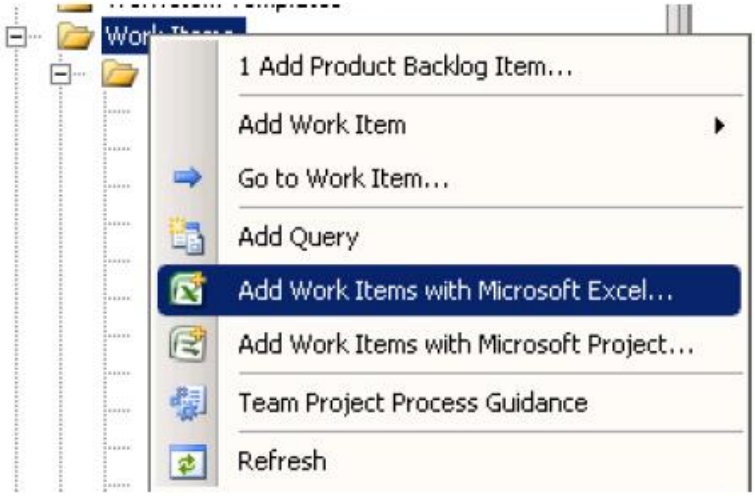
В рамках данного занятия предполагается провести планирование работы команды *по* методологии Scrum. Команды уже провели предварительное знакомство с проектов, а на данном этапе от них требуется следующее.

1. Импортировать содержимое списка требований в TFS, используя средства импорта из *Excel*.
2. Подробно рассмотреть 10 наиболее приоритетных пользовательских историй.
3. Обсудить возникшие вопросы с хозяином продукта.
4. Провести детальное планирование и разбить пользовательские истории на мелкие подзадачи.
5. Распределить подзадачи среди участников проекта.
6. Отчитаться перед хозяином продукта о том, какие пользовательские истории были запланированы. При *отчете* использовать *отчеты* TFS.

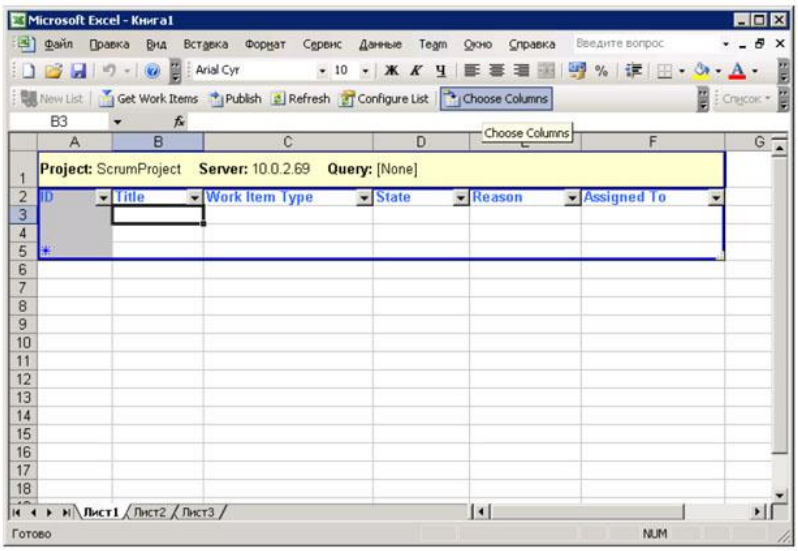
Шаг 1. Импорт списка пользовательских историй

Для того, чтобы загрузить пользовательские истории из *Excel*-файла, полученного командами на прошлом занятии нужно:

1. Выбрать команды **Add work items using Microsoft Excel** в контекстном меню:



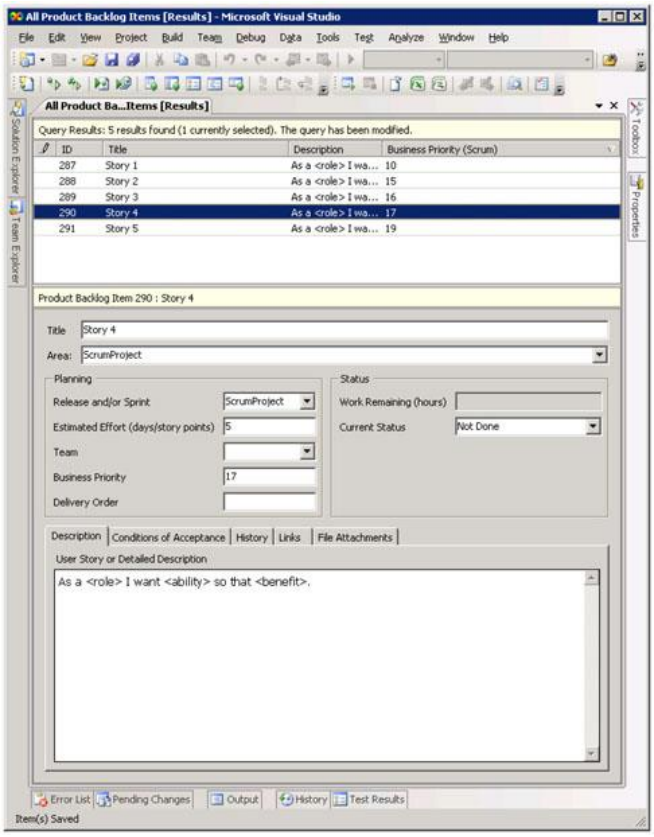
2. В открывшемся окне *Excel* настроить колонки таким образом, чтобы они совпадали порядком и смыслом с колонками в исходном *Excel* документе (для этого можно использовать команды **Choose columns**):



3. После того, как колонки настроены, скопировать значения из исходного *Excel* в редактируемый.

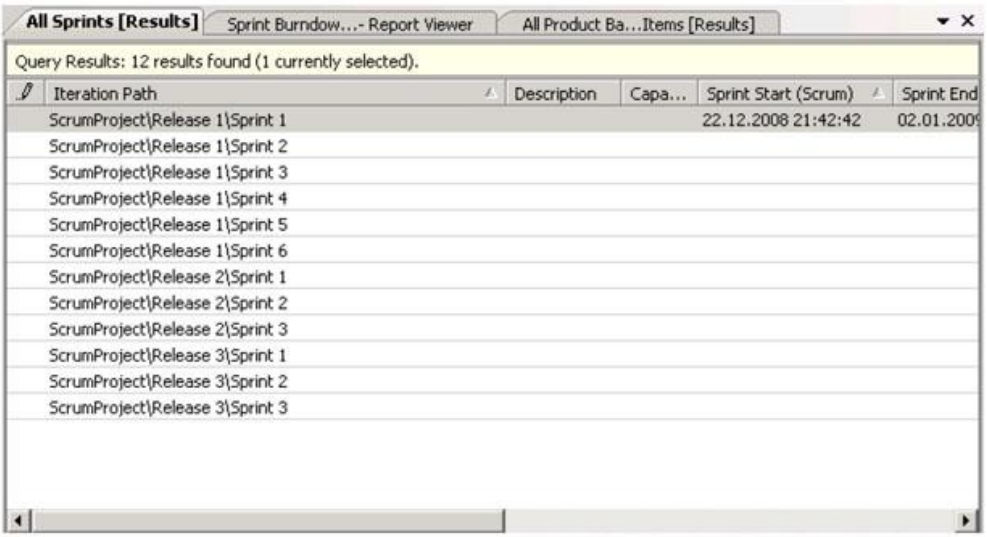
4. Показать колонку с именем **Work Item Type** и задать для все строчек значение **Product backlog item**.
5. Нажать кнопку

6. Убедится, что при выполнении запроса **All product backlog items**, видны все вновь загруженные пользовательские истории:



Шаг 2. Создание sprint

После импорта списка пользовательских историй *команда* должна создать элемент работы, соответствующий начинающемуся sprint. Некоторое количество sprints уже создано *по* умолчанию при создании проекта:



Для активации первого спринта ему необходимо установить дату начала, дату окончания и количество часов, которые *команда* может потратить в этом спринте.

Шаг 3. Формирование Sprint backlog

После обсуждения открытых вопросов *по* 10 наиболее приоритетным пользовательским *историям команда* должна приступить к планированию текущего **sprint** и формированию **sprint backlog**. Для этого ей необходимо рассмотреть *список* всех пользовательских историй и разбить его на *список* более мелких задач. При этом для каждой задачи необходимо создать элемент работы типа **sprint backlog item** и проставить следующие атрибуты:

1. В качестве **sprint** указать **Release1/Sprint1**.
2. Добавить связь с соответствующим элементом **product backlog**, а также со всеми связанными элементами работы.
3. Установить **Estimated efforts** и **Work remaining** в соответствии с оценкой команды.
4. Задать ответственного за задачу (**Owned By**).

Выполнить все *операции* нужно средствами *Visual Studio* и *Team Explorer*.

После создания и распределения задач каждый член команды должен на своей машине убедиться, что выданные ему задачи отображаются в результатах запроса **My Sprint Backlog Items**.

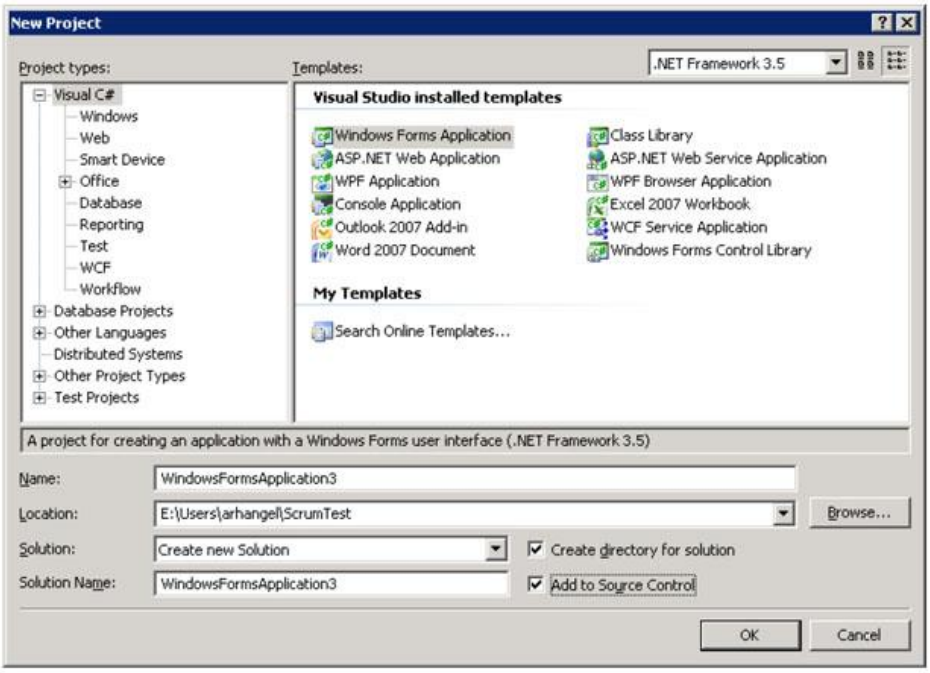
Лабораторная работа 3. Работа с системой контроля версий

Основной целью данного занятия является освоение *системы контроля версий Team Foundation Server* и её *интеграции* с системой отслеживания задач. Занятие предполагает выполнение следующих действий.

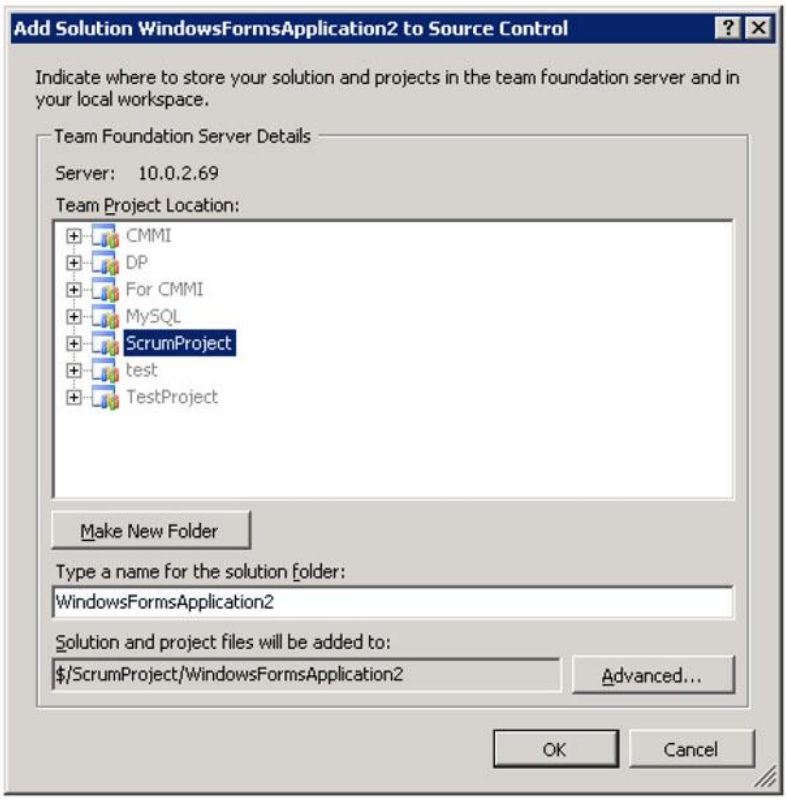
1. Разработка кода модельной задачи средствами *Visual Studio* и внесение его в систему управления версиями.
2. Проставление связей между вносимыми изменениями и элементами системы отслеживания задач.
3. Создание параллельно поддерживаемых веток кода.
4. Интеграция изменений, сделанных параллельно в одном файле или в разных ветках кода.

Шаг 1. Разработка кода

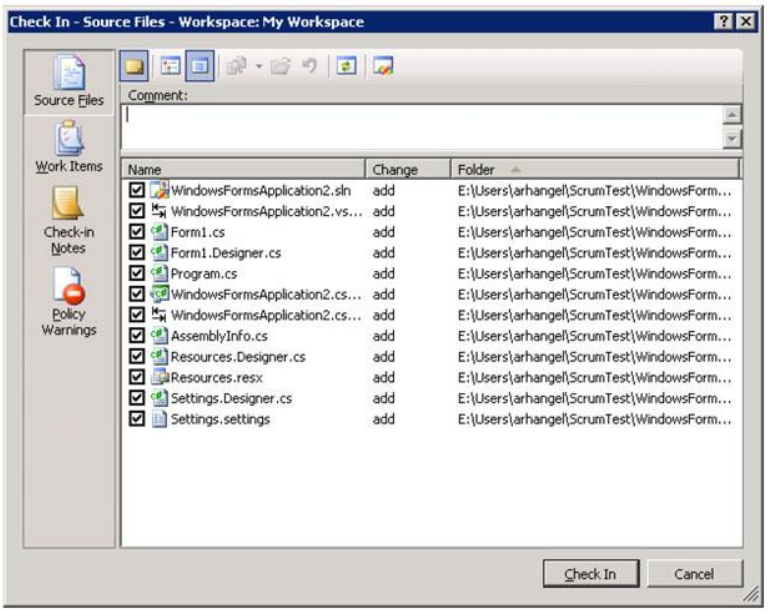
Перед началом работы команде необходимо создать решение (**solution**) средствами *Visual Studio*, включив опцию **Add to Source Control**:



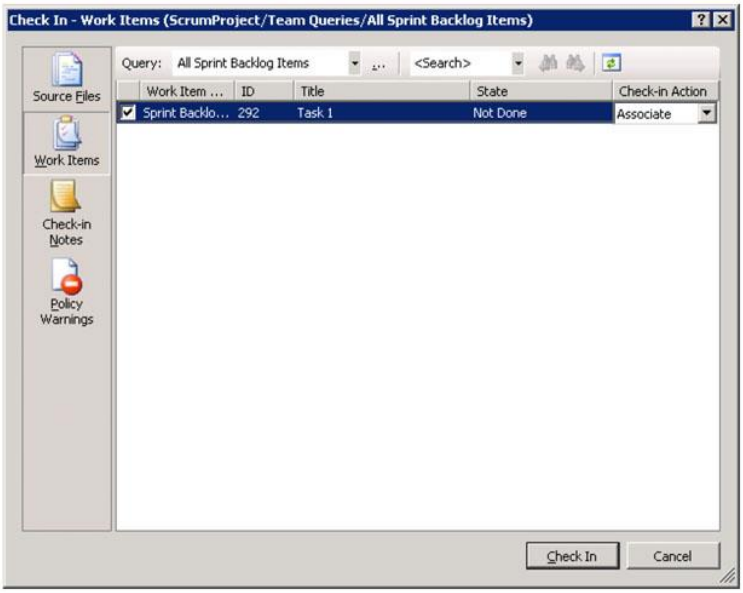
В открывшемся после создания проекта окне необходимо выбрать командный проект, в систему *контроля версий* которого нужно добавить данное решение:



Затем необходимо внести все данные в систему *контроля версий*, используя команду **Check-in**, открывающую диалог:



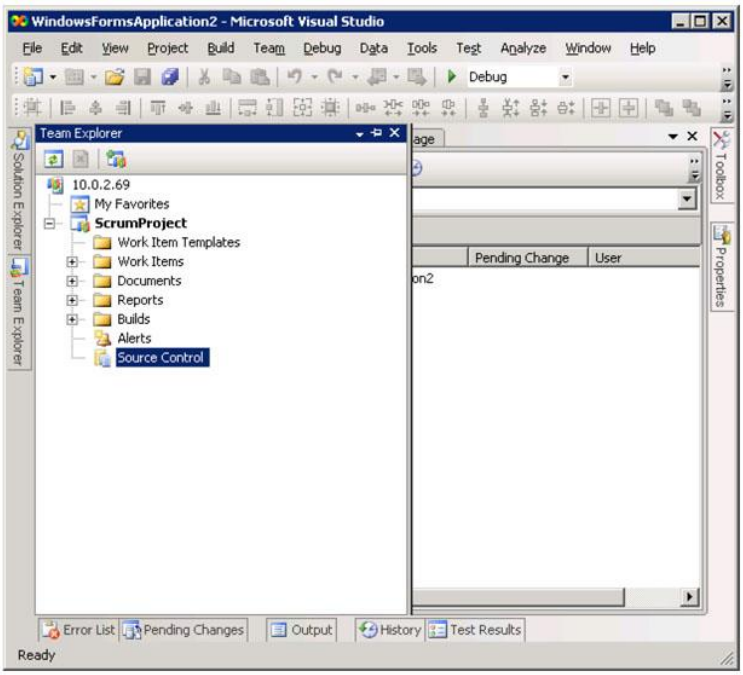
В этом диалоге необходимо внести комментарии к вносимому коду, а также, на вкладке **Work items**, связать вносимое изменение с элементами работы:



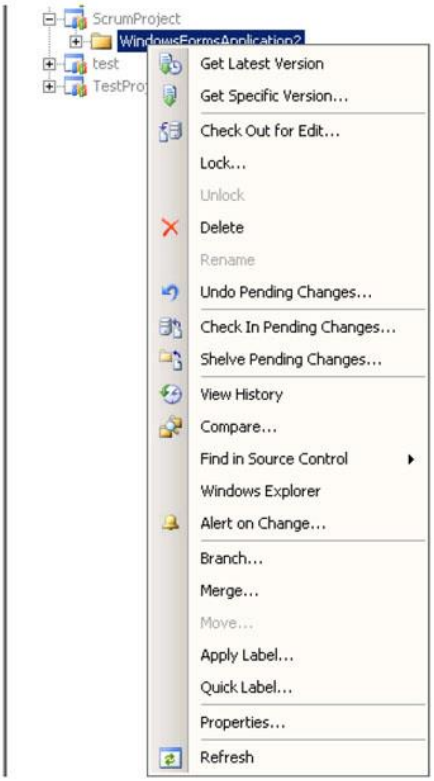
Шаг 2. Создание ветки кода

Для того, чтобы освоится с практикой *конфигурационного управления*, команды должны создать *ветвь* в *системе контроля версий*, следуя приведенной ниже *инструкции*.

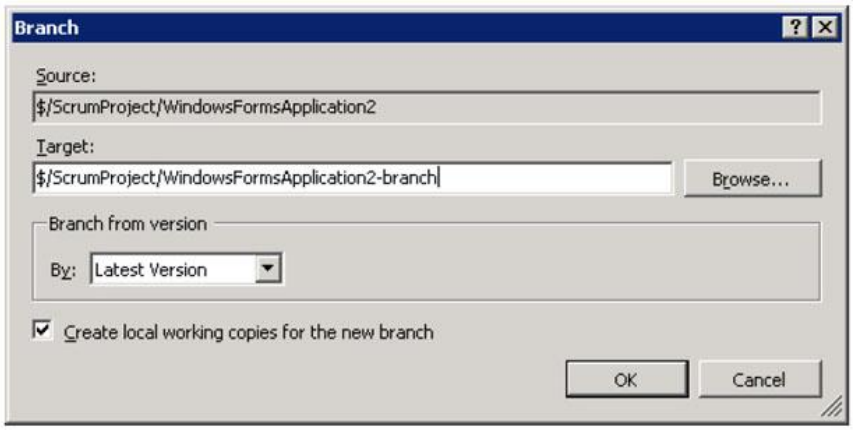
1. Открыть *Source control explorer*:



2. Выбрать нужный проект и в контекстном меню команду **Branch**:



3. В открывшемся окне задать целевую папку, куда необходимо скопировать данные для новой ветви:

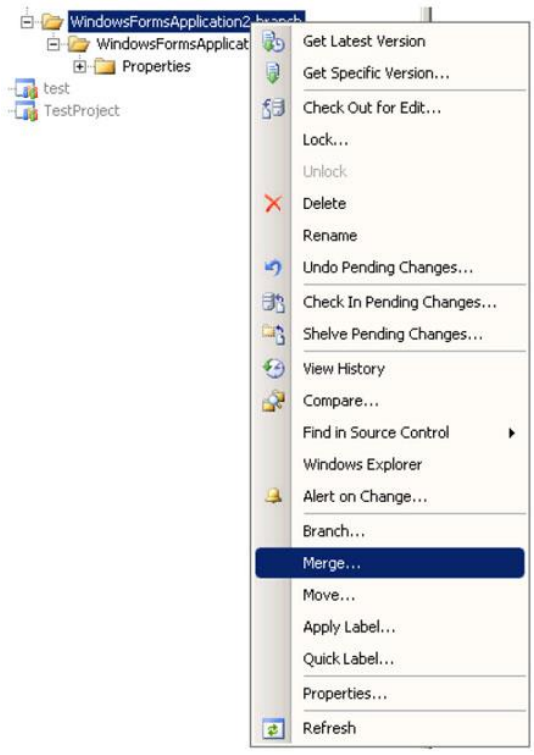


4. Внести изменения с помощью команды **Check-in**

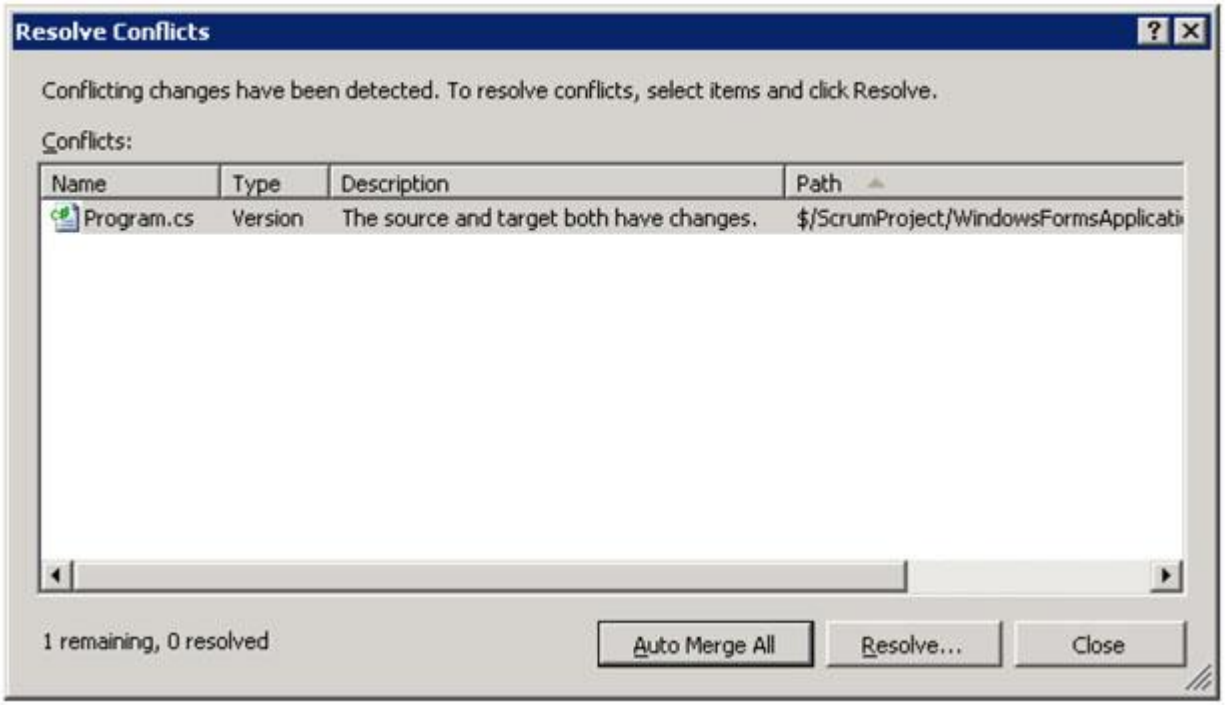
После того, как создана *ветка*, разные *участники команды* вносят изменения в разные ветки кода, реализую необходимую функциональность приложения.

Шаг 3. Объединение изменений

После того, как в отдельные ветви было внесено некоторое количество изменений, необходимо перенести изменения из отделенной ветви в основную, используя команду **Merge**:



В процессе *объединения* изменений могут возникнуть *конфликты*, *информация* о которых будет включена в сообщение следующего вида:



Все *конфликты* необходимо разрешить, используя команду **Resolve** и *утилиту* для *объединения* результатов.

После *разрешения конфликтов* все изменения внести в систему *контроля версий* посредством *операции* **Check-in**.

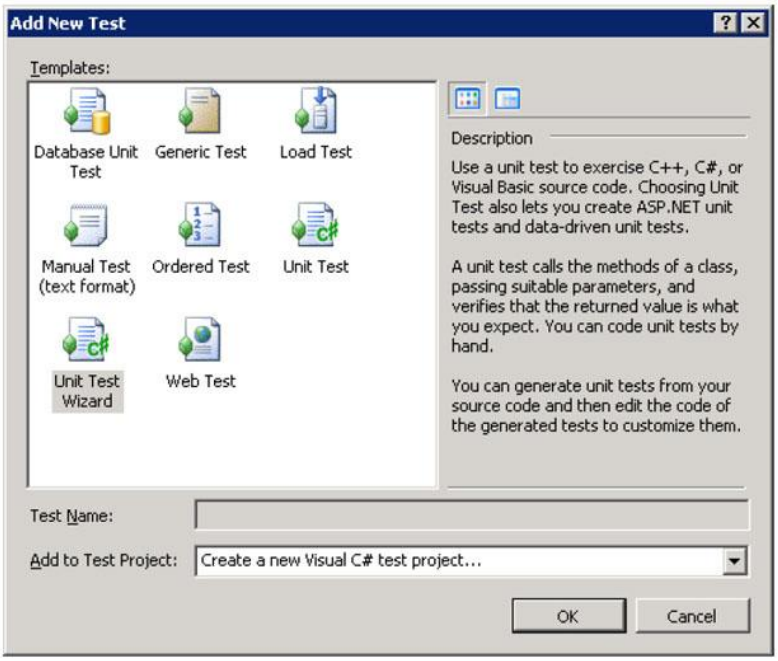
Лабораторная работа 4. Разработка модульных тестов

На данном занятии команды должны разработать набор *модульных* тестов, покрывающих функциональность, разработанную на занятии предыдущем. В рамках данного занятие предполагается освоить следующие возможности MS VSTS.

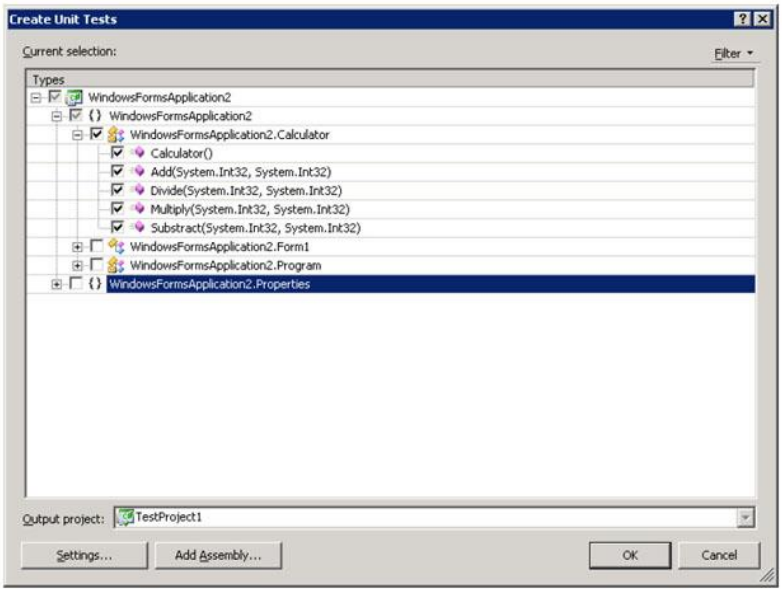
1. Автоматическая генерация тестов.
2. Наполнение тестов содержимым.
3. Запуск тестов и просмотр результатом.
4. Изменение *конфигурации* работы тестов.

Шаг 1. Автоматическая генерация тестов

Для ускорения разработки тестов команды могут воспользоваться возможность *Visual Studio no* автоматической генерации тестов. Для этого необходимо воспользоваться командой **Test/New Test** и выбрать **Unit Test wizard** в открывшиеся окне:



После создания тестового проекта, будет предложен выбор из тех типов и методов, для тестирования которых необходимо создать *заглушки*:



В этом диалоге *команда* должна выбрать все основные классы и методы, которые планируется покрыть модульными тестами.

Шаг 2. Наполнение тестов содержимым

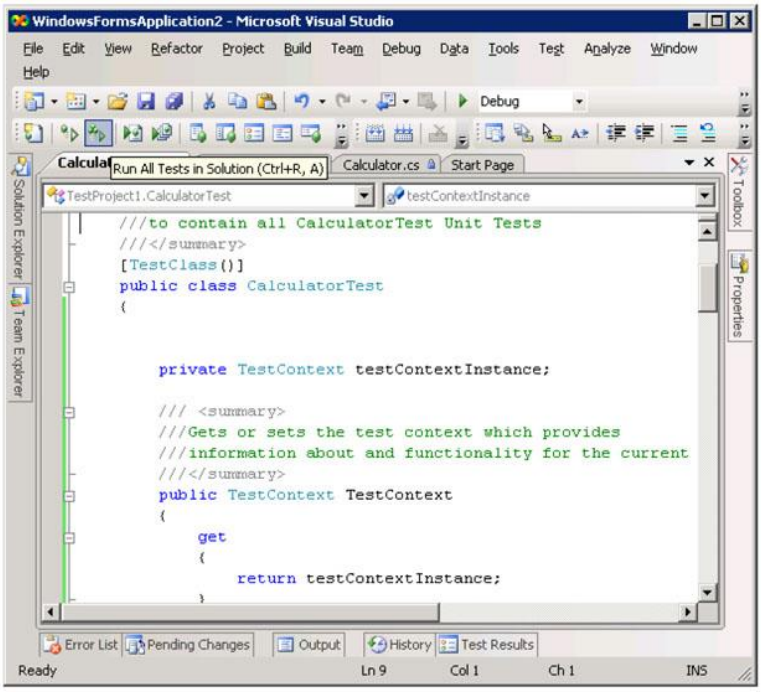
После генерации тестового покрытия *команда* получить набор скелетов тестов для всех методов, которые были выбраны для тестирования. Однако, эти тесты имеют достаточно простую структуру и пока лишены смысла:

```
/// <summary>
///A test for Multiply
///</summary>
[TestMethod()]
public void MultiplyTest()
{
  // TODO: Initialize to an appropriate value
  Calculator target = new Calculator();
  int a = 0; // TODO: Initialize to an appropriate value
  int b = 0; // TODO: Initialize to an appropriate value
  int expected = 0; // TODO: Initialize to an appropriate value
  int actual;
  actual = target.Multiply(a, b);
  Assert.AreEqual(expected, actual);
  Assert.Inconclusive("Verify the correctness of this test
  method.");
}
```

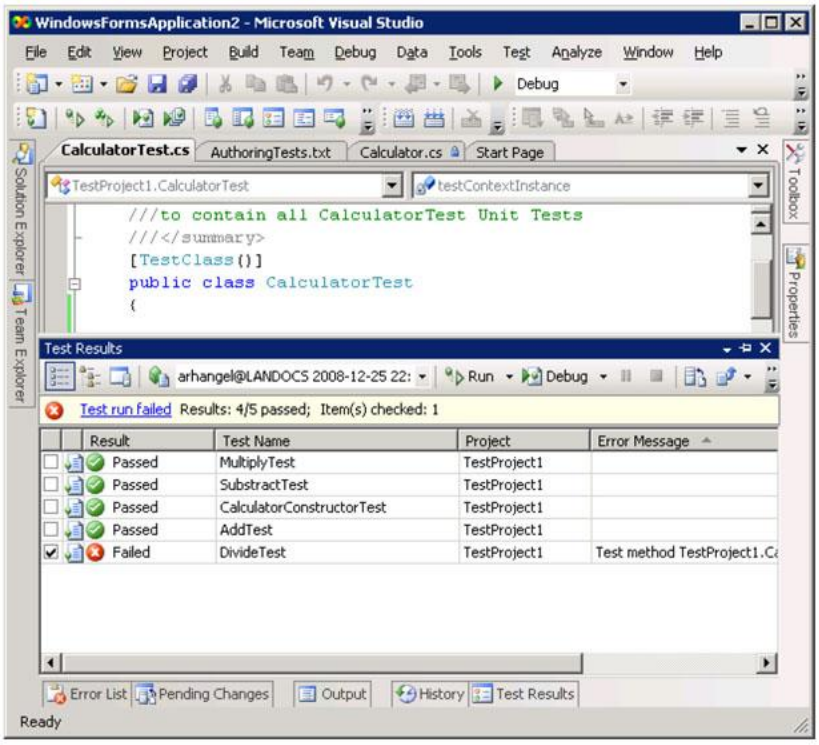
На следующем шаге *команда* должна заполнить эти тесты необходимым содержимым, используя функциональность *по валидации* (*Assert*), предоставляемую тестовой платформой.

Шаг 3. Запуск тестов

Для того, чтобы исполнить созданные тесты необходимо использовать соответствующую *панель инструментов*:



После этого результаты выполнения тестов будут видны в окне результатов:

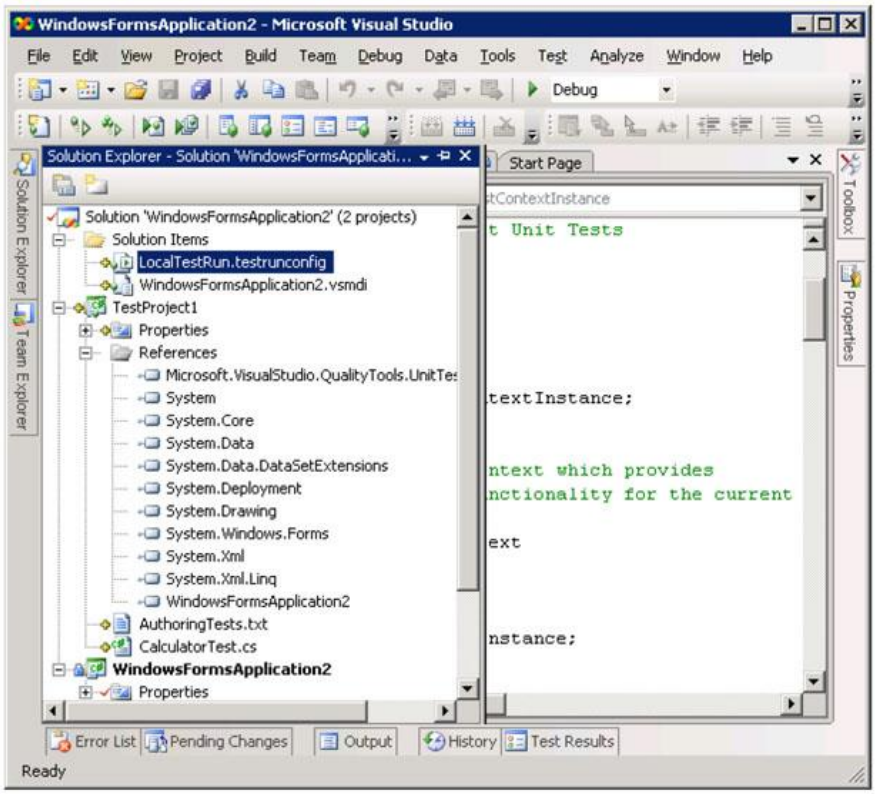


Команды должны добиться того, чтобы все разработанные тесты проходили успешно.

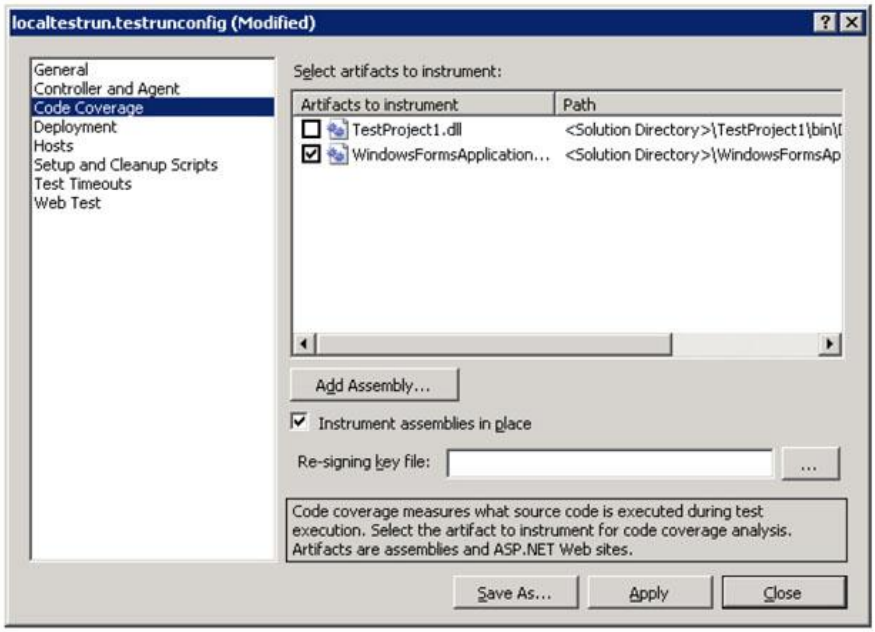
Шаг 4. Изменение конфигурации тестов

Для того, чтобы проанализировать качество разработанных тестов, *команда* должна вычислить тестовое покрытие. Для этого её необходимо:

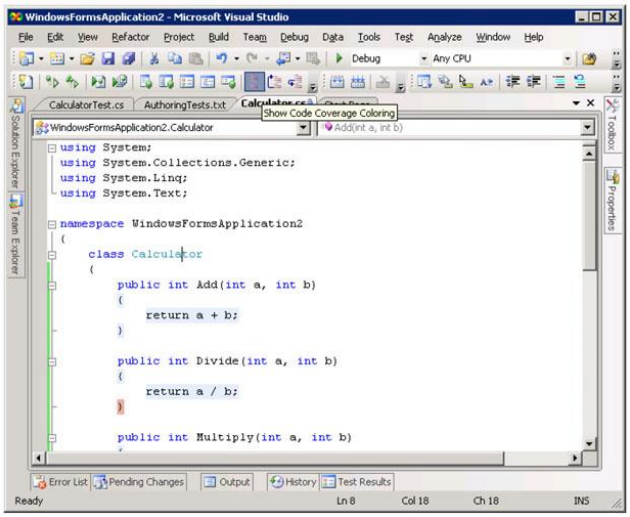
1. Открыть файл *конфигурации* запуска тестов, автоматически добавленный к решению при создании тестов:



2. На открывшемся диалоге выбрать вкладку **Code Coverage** и установить то, какие именно проекты нужно анализировать:



3. После сохранения *конфигурации* запустить тесты и активировать опцию **Show Code Coverage Coloring**:



Лабораторная работа 5. Создание и конфигурация автоматической сборки

На данном этапе *команда* должна создать в своем проекте процедуру автоматической *сборки*. При этом должно быть создано несколько процедур:

1. Простая процедура, включающая только сборку.
2. Полная процедура, включающая тесты и *анализ* кода.

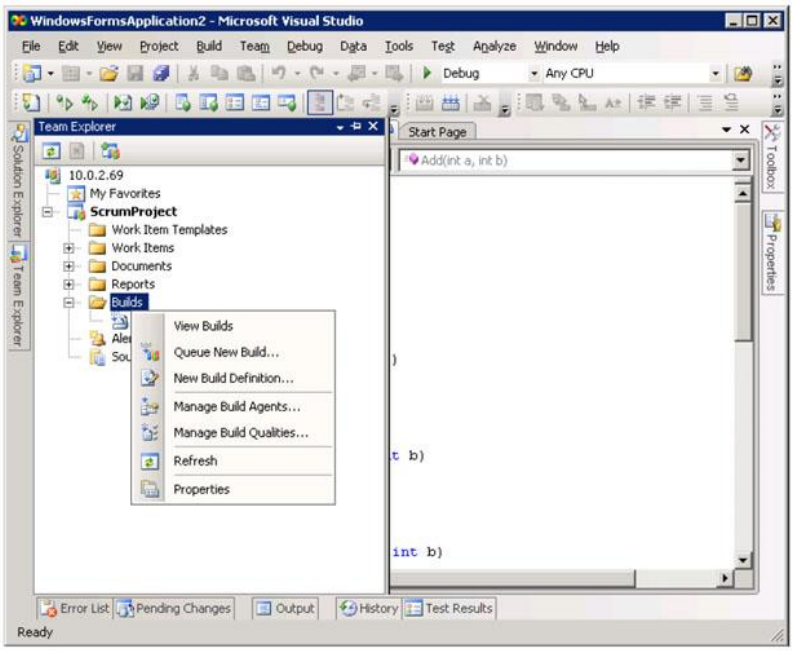
После создания сборок необходимо настроить параметры непрерывной *интеграции*:

1. Простая сборка должна запускаться после каждого внесенного изменений, но не чаше чем раз в 5 минут.
2. Полная сборка должна запускаться каждую ночь.

Шаг 1. Создание простой сборки

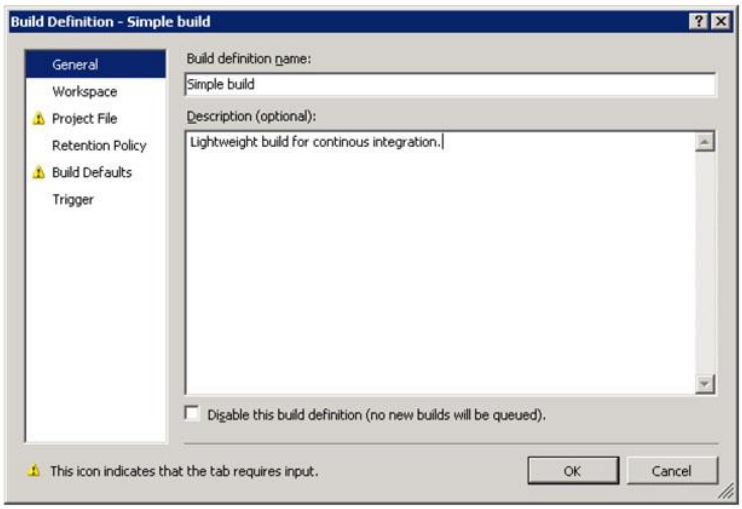
Все результаты *сборки*, проведенной TFS, выкладываются в разделяемую папку, на *запись* в которую есть *права* у пользователя, с правами которого работает *сервер* автоматических сборок (обычно – TFSBuild), а также у пользователя, с правами которого работает сам TFS (обычно – TFSService). Как правило, учащиеся не обладают достаточным количеством прав для создания такого рода папок, поэтому они должны быть заранее подготовлены преподавателем.

Для создания простой *сборки* необходимо обратится к окну **Team Explorer**, после чего в разделе **Builds** выбрать команду **Build Definitions**:

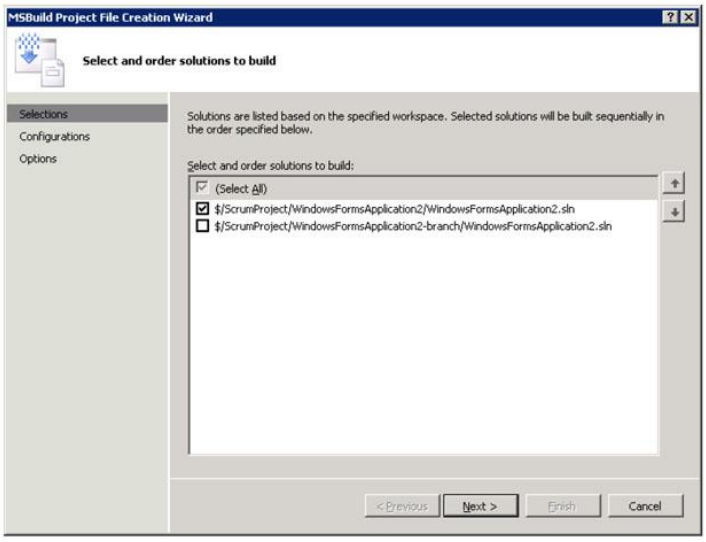


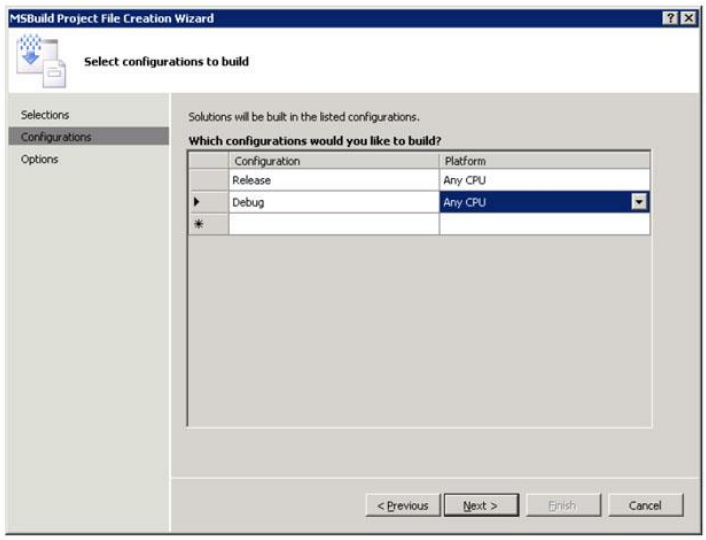
Вызов этой команды приведет к открытию мастера, в котором нужно задать следующие параметры *сборки*:

1. Задать имя и описание *сборки*:

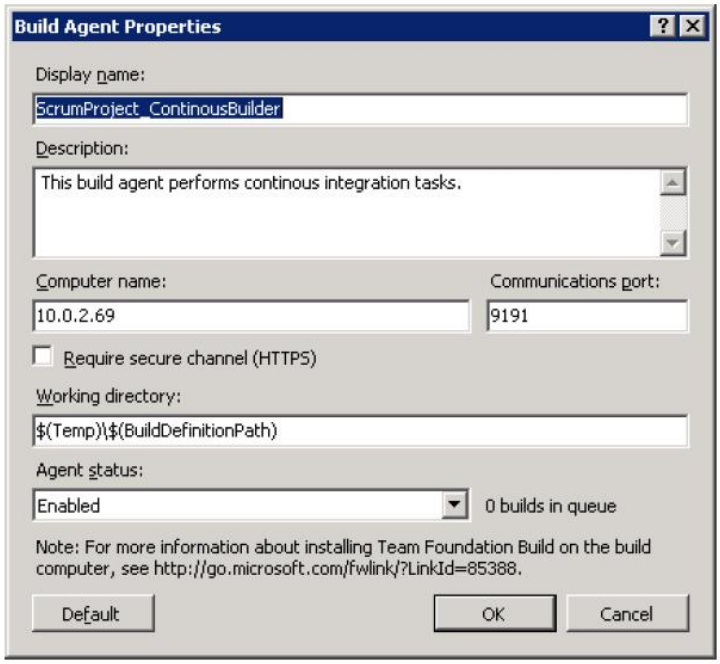


2. На закладке **Project File**, создать новое описание *сборки* используя кнопку **Create**, после чего задав проекты и *конфигурации* для *сборки*:

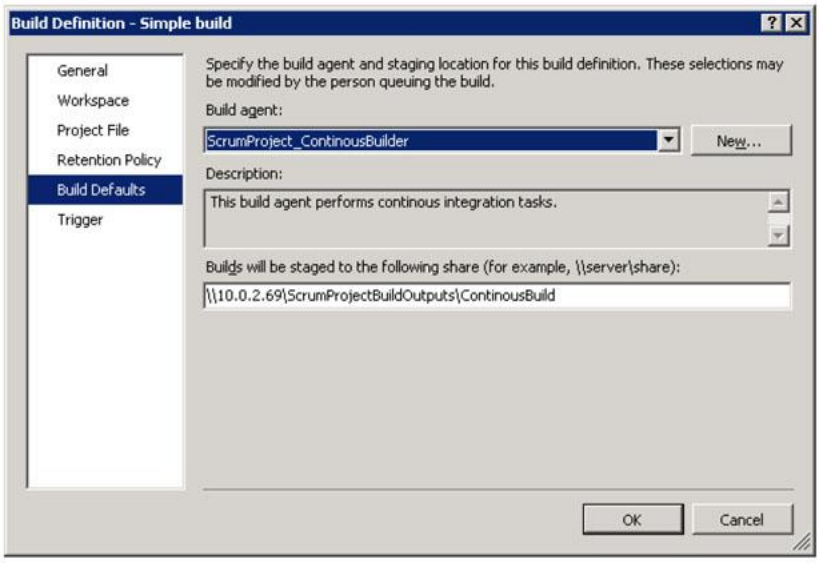




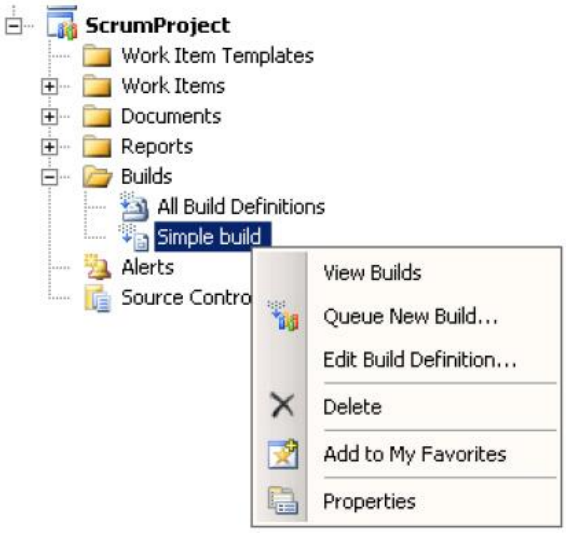
3. На закладке **Build Defaults** создать определение *агента*-сборщика используя кнопку **New**.
   Для *агента* указать имя, описание, и *IP адрес* сервера сборок (как правило, совпадает с сервером TFS):



4. На закладке **Build Defaults** также необходимо задать имя разделяемой папки, в которую будут сложены результаты:



После того, как *определение сборки* было создано, её необходимо запустить, используя команду **Queue new build**:

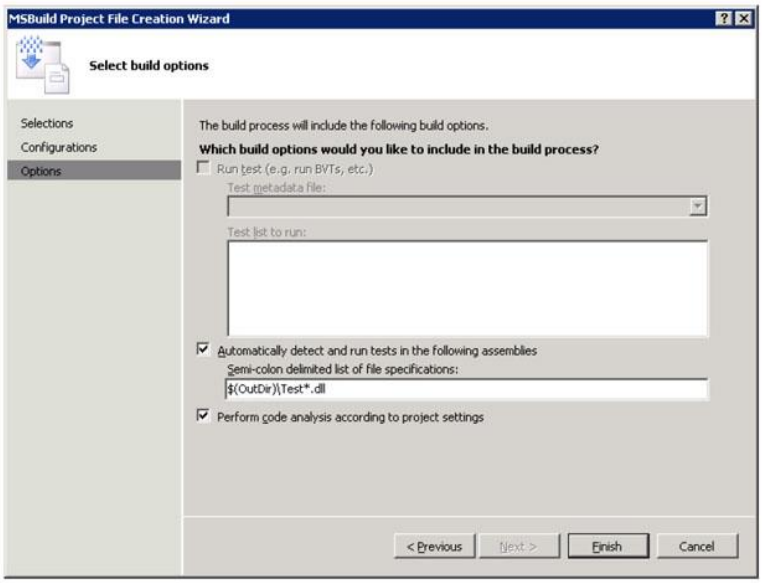


После запуска *сборки* необходимо дождаться ее завершения, и убедится, что на соответствующей сетевой папке появились результаты *сборки*.

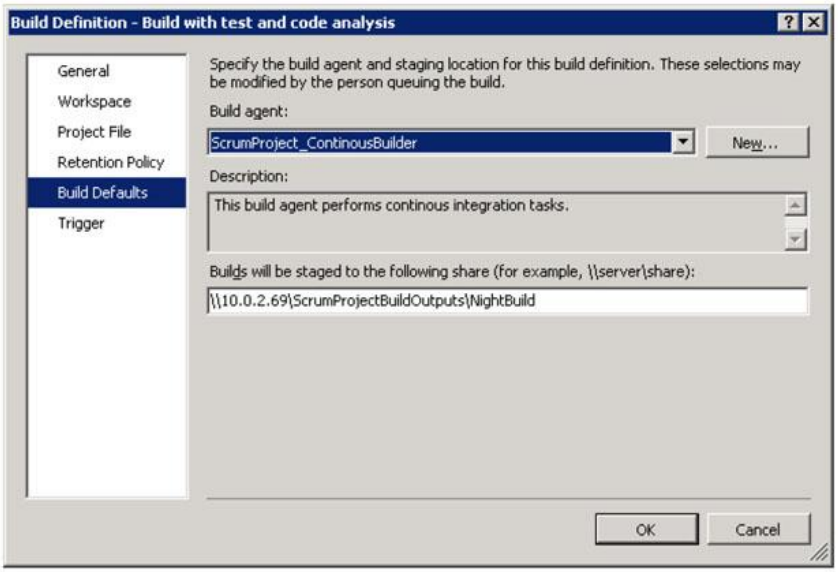
Шаг 2. Создание сложной сборки

Создание сложной *сборки* проходит во многом аналогично созданию *сборки* простой, за *исключением* нескольких шагов:

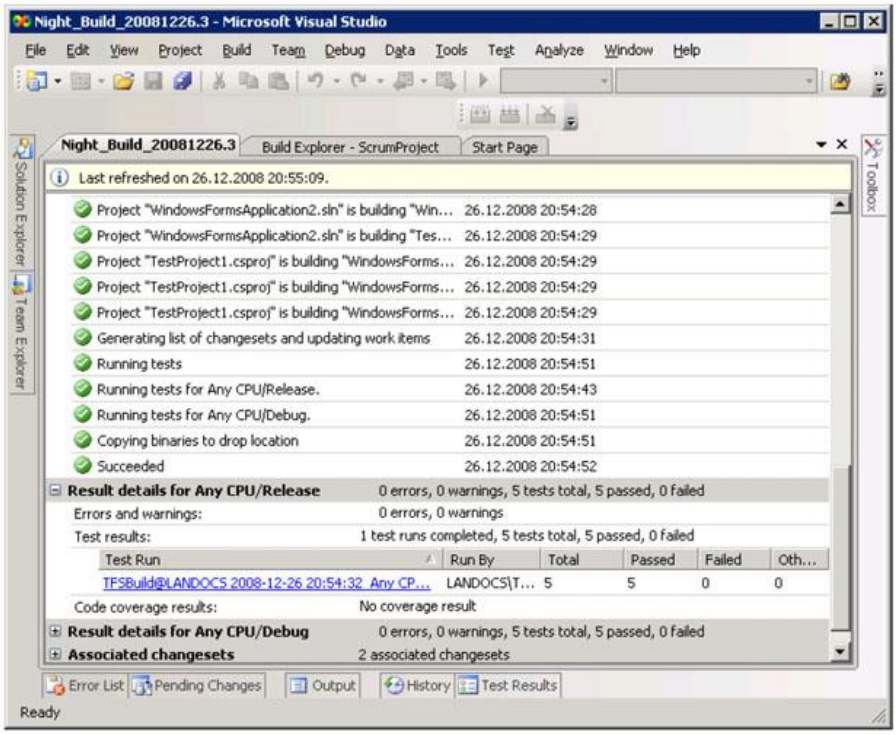
1. На закладке опций при создании проекта *сборки* необходимо включить автоматический запуск *модульных* тестов и *анализ* кода:



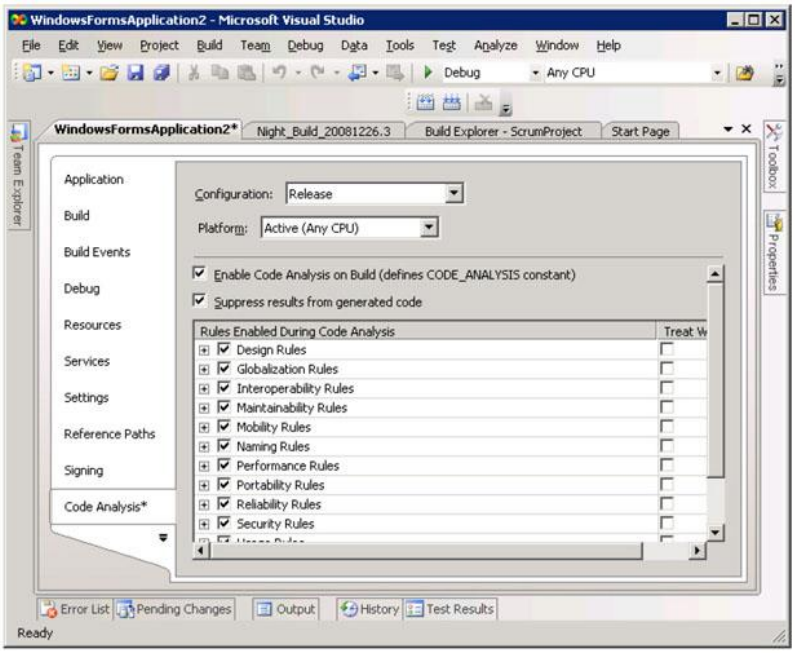
2. На закладке **Build Defaults** необходимо задать другую папку для сбора результатов:



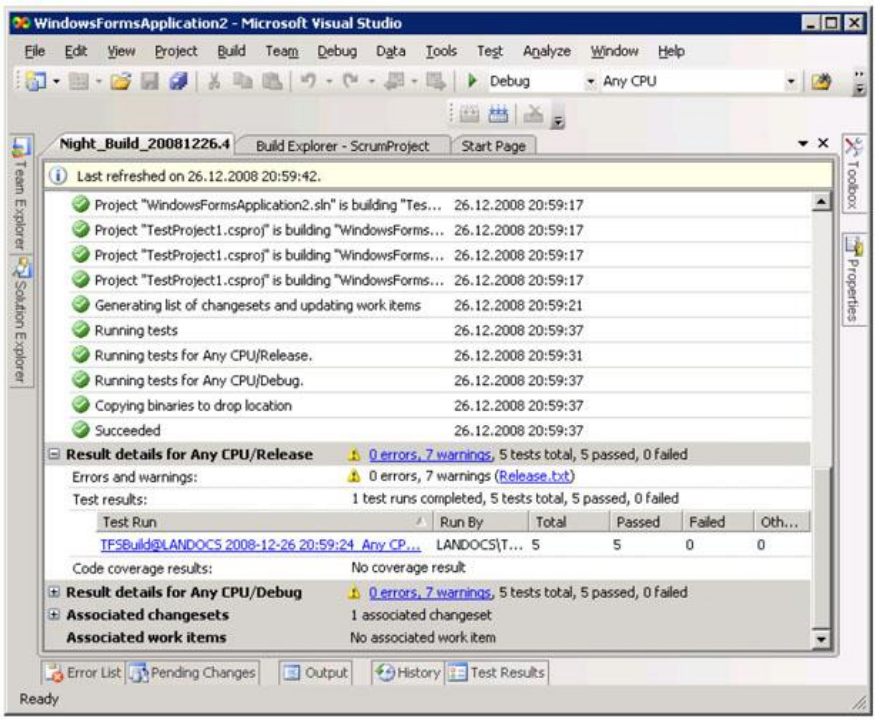
После создания *сборки* необходимо запустить её, и обратить внимание, что результаты *сборки* включают и результаты тестов:



Теперь нужно добиться выполнения статического *анализа* кода во время ночной *сборки*. Для этого необходимо активировать *анализ* кода в настройках соответствующих проектов:



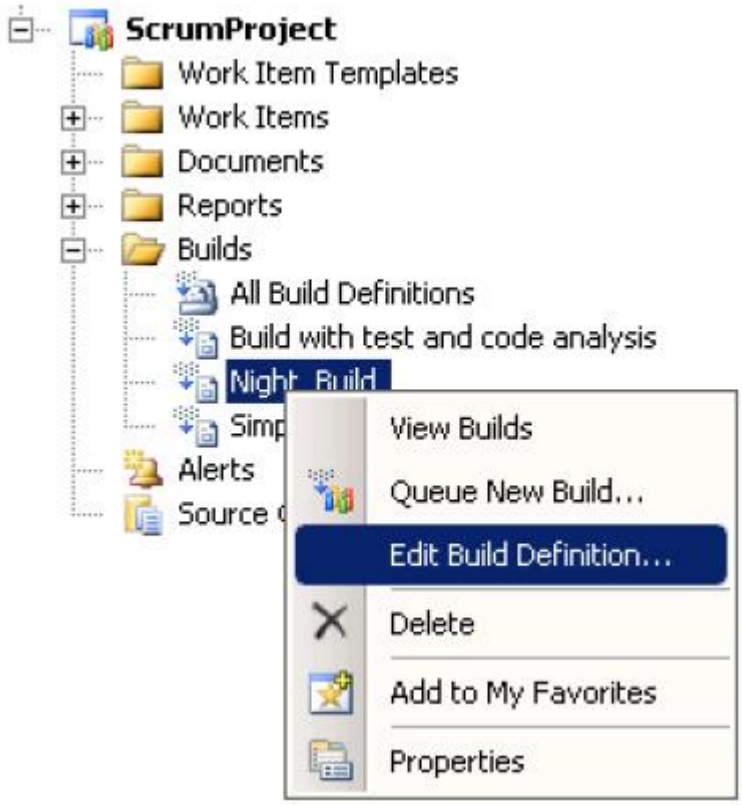
Следующая же собранная *сборка* будет содержать большое количество предупреждений от анализатора:



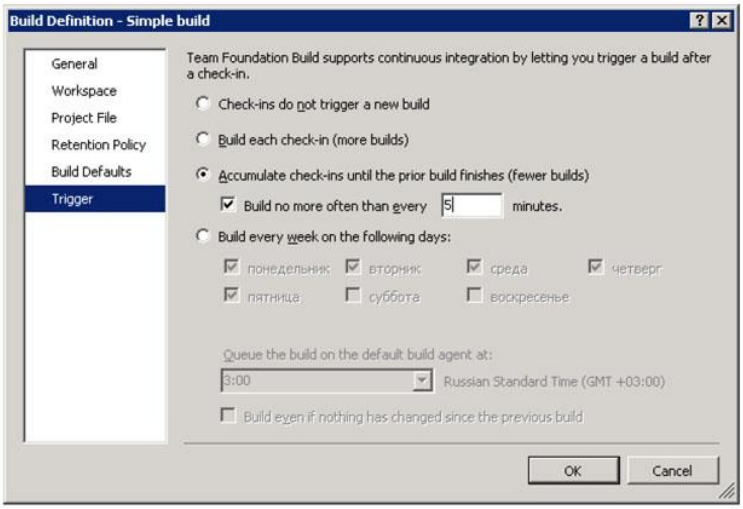
Шаг 3. Настройка непрерывной интеграции

На данном шаге учащимся необходимо исправить описания сборок таким образом, чтобы они выполнялись автоматически при определенных условиях. Простой вариант *сборки* должен запускаться автоматически после каждого внесения изменений, но не чаще, чем в пять минут. Для того, чтобы добиться этого необходимо:

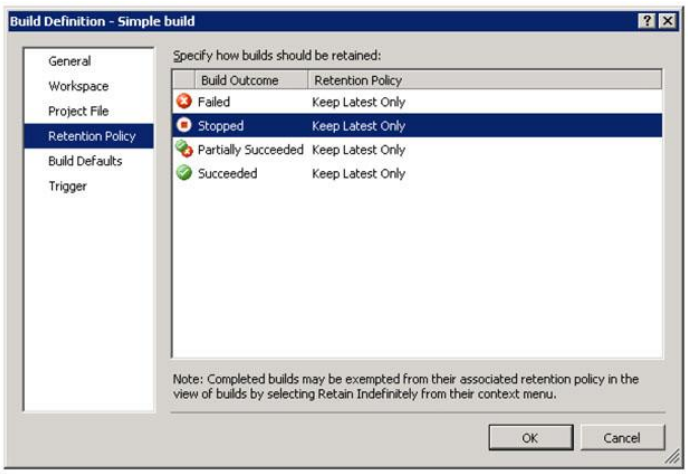
1. Вызвать команду **Edit build definition**:



2. Задать настройки автоматического запуска на закладке **Trigger**:



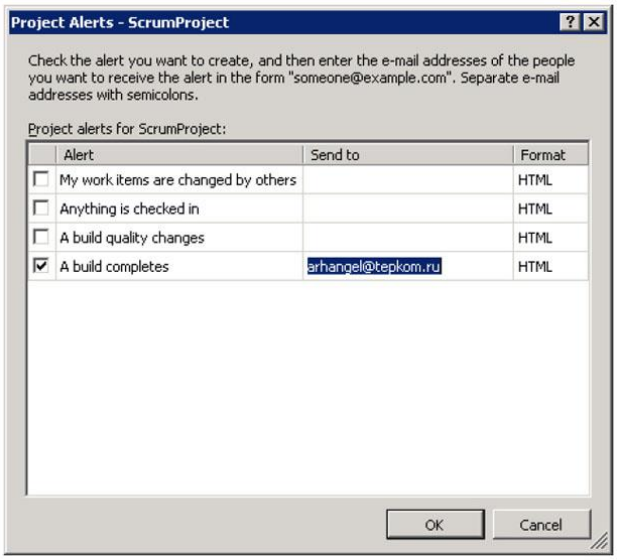
3. Настроить политику очистки сборок на закладке **Retention Policy**. Это необходимо для того, чтобы избежать быстрого исчезновения места на машине-сборщике и для удаления из базы TFS информации о второстепенных сборках:



Проведение *сборки* после внесения изменений наиболее эффективно в том случае, если участники проекта получают нотификации о том, что *сборка* была проведена. Для того чтобы этого добиться необходимо:

1. В контекстном меню проекта выбрать команду **Project Alerts**:

2. В списке событий, о которых нужно слать нотификации, выбрать **"a build completes"** и задать список адресов электронной почты, на которые нужно отправить сообщение:



После настройки простой *сборки* для запуска при внесении изменения необходимо проверить работу системы – внести некоторое изменение и дождаться сообщения о сборке.

Аналогичным образом можно настроить и автоматический *запуск* сложной *сборки* каждый день в определенное время.

Лабораторная работа 6. Настройка шаблона процесса

На завершающем этапе, соответствующем окончанию спринта, команды должны провести ретроспективу своей деятельности и сформировать *список* возможных изменений в процессе и работе с TFS. Все замечания должны быть занесены в TFS как соответствующие элементы работы.

В рамках ретроспективы *команда* должна предложить некоторые изменения к элементам работы, вовлеченным в процесс, а затем и реализовать эти изменения.

Шаг 1. Ретроспектива

На ретроспективе *команда* в течение 20-30 минут обсуждает то, как прошел данный спринт, выделяя позитивные и негативные моменты, а также предложения *по* изменениям.

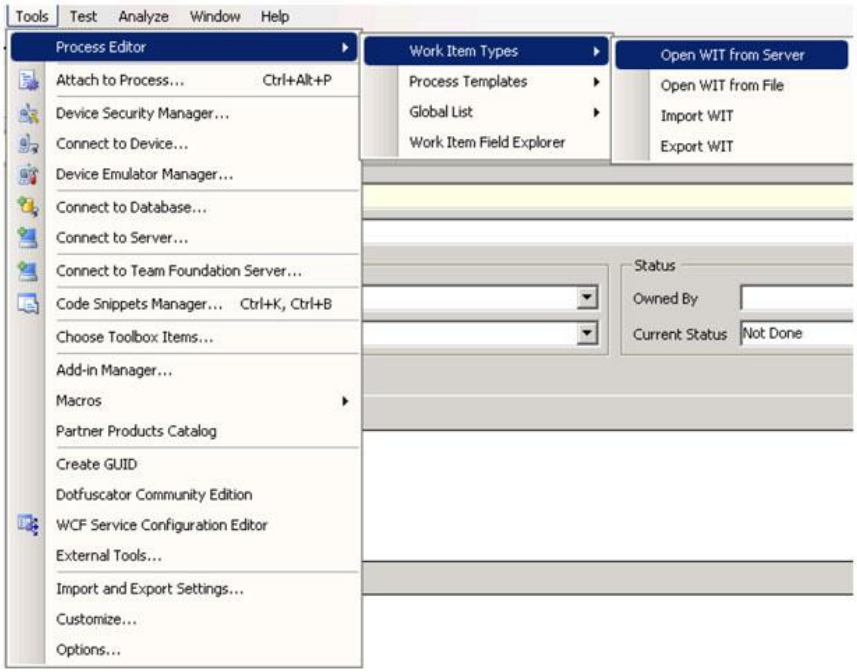
Все комментарии должны быть внесены в соответствующий элемент работы типа Sprint retrospective, а для каждого предложения *по* улучшению заведены элементы работы типа Sprint backlog *item*, а для каждого идентифицированного негативного момента, требующего устранения – элемент работы типа Impediment.

Результаты ретроспективы необходимо обсудить с хозяином продукта.

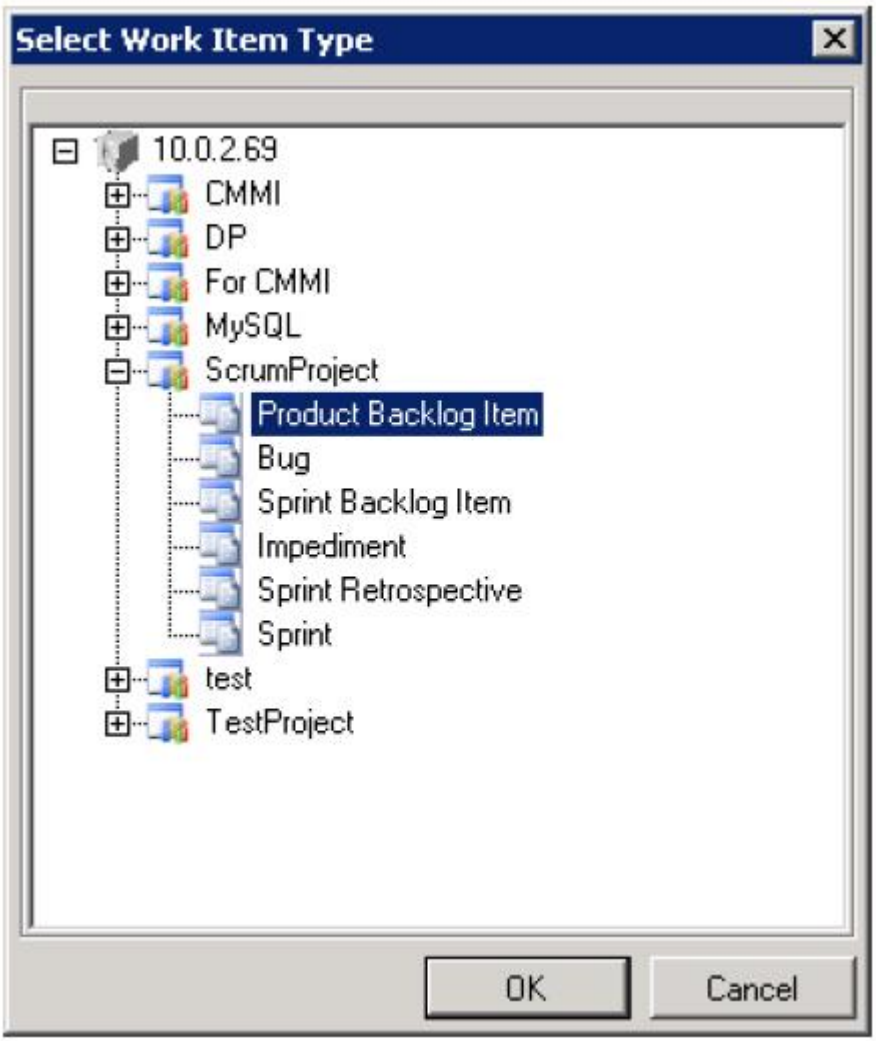
Шаг 2. Изменение элемента работы

На этапе ретроспективы *команда* должна выявить некоторые изменения в формате и жизненном цикле элементов работы, которые помогут повысить эффективность команды. На следующем шаге им нужно воплотить эти изменения в жизнь. Для этого необходимо:

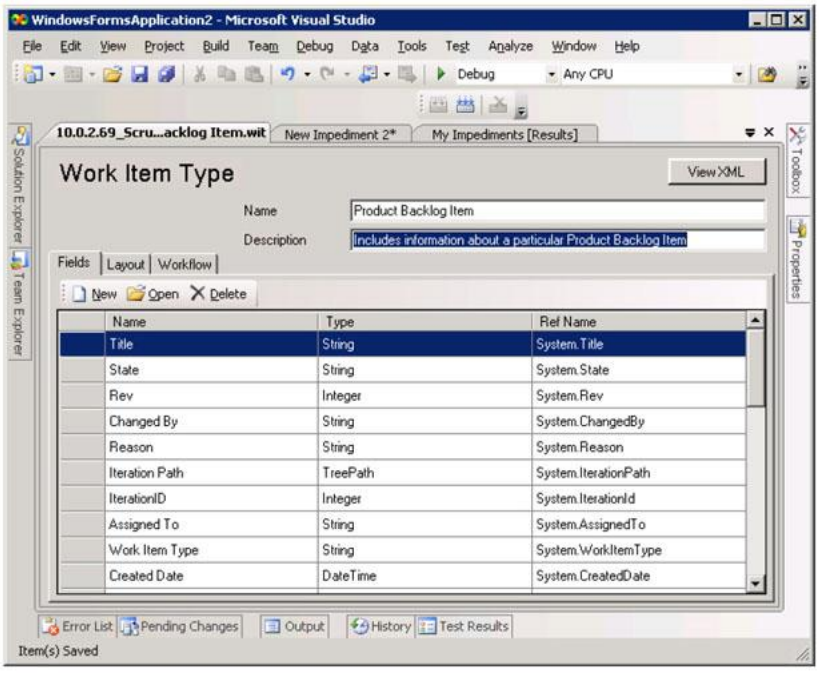
1. Открыть тип элемента работы на редактирование с помощью команды меню **Tools**:



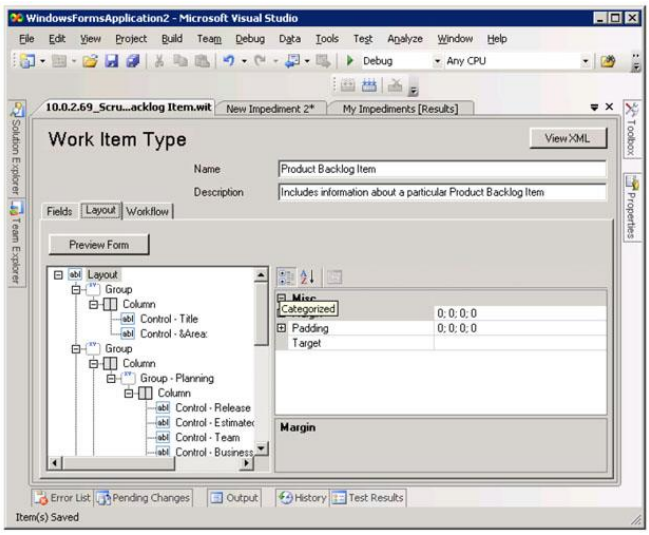
2. Выбрать нужный элемент работы в открывшемся диалоге:



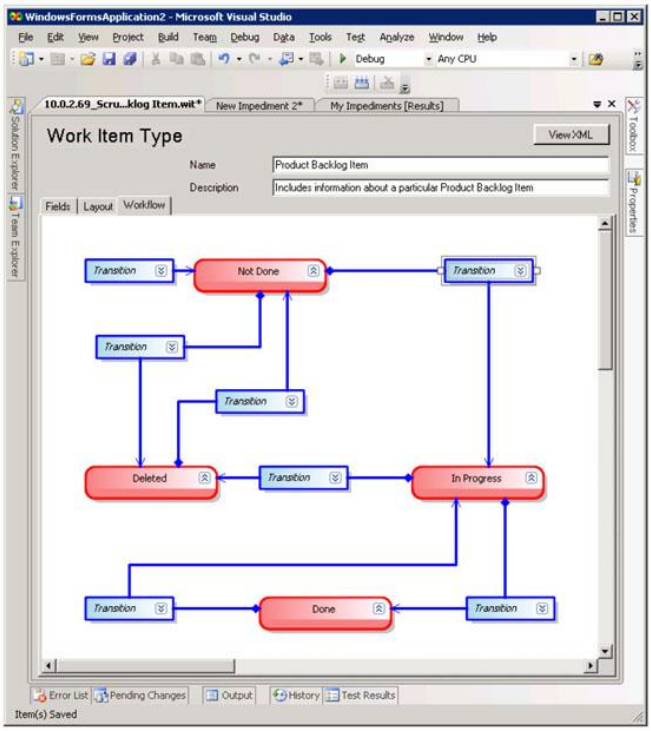
3. На закладке **Fields** добавить, удалить или изменить необходимые поля:



4. На закладке **Layout** изменить соответствующим образом визуальное представление элемента работы:



5. На закладке **Workflow** внести необходимые изменения в жизненный цикл элемента работы:



После внесения всех изменений их нужно сохранить, а затем убедиться, что они применились к существующим и к вновь создаваемым элементам работы.

4.1.2. Тестирование

Тема 1. Модели и профили жизненного цикла программных средств
Тема 2. Модели и процессы управлении проектами программных средств.
Тема 3. Управление требованиями к программному обеспечению
Тема 4. Конструирование (детальное проектирование) программного обеспечения
Тема 5. Инструменты и методы программной инженерии
Тема 6. Качество программного обеспечения

4.1.2.1. Порядок проведения.

Тестирование проходит в письменной форме или с использованием компьютерных средств. Обучающийся получает определённое количество тестовых заданий. На выполнение выделяется фиксированное время в зависимости от количества заданий. Оценка выставляется в зависимости от процента правильно выполненных

заданий. Тестирование проводится по вариантам. В каждом варианте – 10 тестовых заданий. За каждый правильный ответ начисляется 1 балл. Итого за тестирование студент может заработать до 10 баллов.
Ниже приведены примерные задания. Полный банк тестовых заданий хранится на кафедре.

4.1.2.2 Критерии оценивания

17-20 баллов ставится, если обучающийся набрал 86% правильных ответов и более.
14-16 баллов ставится, если обучающийся набрал от 71% до 85 % правильных ответов.
11-15 баллов ставится, если обучающийся набрал от 56% до 70% правильных ответов.
0-10 баллов ставится, если обучающийся набрал 55% правильных ответов и менее.

4.1.2.3. Содержание оценочного средства

1. Классификация автоматизированных информационных технологий по способу реализации автоматизированных информационных систем включает:
1 традиционные технологии;
2 новые информационные технологии;
3 технологии поддержки принятия решений;
4 технологии работы с СУБД
A. Верно только 1, 2
B. Верно только 3, 4
C. Верно только 1, 2, 3
D. Все верно
2. Классификация автоматизированных информационных технологий по степени охвата задач управления включает:
1 технологии электронной обработки данных;
2 технологии автоматизации функций управления;
3 технологии электронного офиса;
4 технологии экспертных систем
A. Верно только 1, 2
B. Верно только 3, 4
C. Верно только 1, 2, 3
D. Все верно
3. Классификация автоматизированных информационных технологий по классу реализуемых технологических операций включает:
1- технологии работы с текстовыми и табличными процессорами;
2 - технологии работы с СУБД;
3 - технологии работы с графическими объектами и мультимедийными системами;
4 - технологии работы с экспертными системами
A. Верно только 1, 2
B. Верно только 3, 4
C. Верно только 1, 2, 3
D. Все верно
4. Классификация автоматизированных информационных технологий по способу построения сети включает:
1- распределенные;
2- многоуровневые;
3- виртуальные;
4 - локальные.
A. Верно только 1, 2
B. Верно только 3, 4
C. Верно только 1, 2, 4
D. Все верно
5. Укажите правильное утверждение в отношении понятия новая информационная технология
A. Это технология, которая основывается на применении новейших компьютеров
B. Это технология, которая основывается на применении доступа

пользователя к удаленным базам данных и программам.
C. Это технология, которая основывается на активном участии пользователей в информационном процессе, высоком уровне дружественного пользовательского интерфейса, широком использовании пакетов прикладных программ общего и проблемного назначения, доступе пользователя к удаленным базам данных и программам благодаря вычислительным сетям ЭВМ.
D. Это технология, которая основывается на применении компьютеров, широком использовании пакетов прикладных программ общего и проблемного назначения, доступе пользователя к удаленным базам данных и программам.
6.Технологическая платформа информационных технологий определяет:
1- модель вычислительного оборудования;
2 - операционную систему;
3 - компилятор языка программирования;
4 - совокупность стандартов, определяющих некоторую технологию разработки приложений.
A. Верно только 1, 2
B. Верно только 3, 4
C. Верно только 1, 2, 3
D. Все верно в различных сочетаниях
7.Поддерживает ли алгоритмический язык С преимущества столпов объектно-ориентированного программирования инкапсуляции, полиморфизма и наследования.
A. да
B. нет
C. частично
D.
8.Укажите правильное утверждение
A. Язык С является объектно-ориентированным языком.
B. Язык С++ является объектно-ориентированной надстройку над С.
C. Библиотек MFC (Microsoft Foundation Classes) используется совместно с языком Visual Basic
D. Платформа инструментальных средств Visual Basic не имеет интегрированных средств быстрой разработки программ
9.Платформа Microsoft .NET предоставляет:
1 - устойчивую общеязыковую среду выполнения CLR;
2 - средства разработки приложений на любом из многих языков программирования, поддерживаемых платформой .NET;
3 - библиотеку классов .NET Framework;
4 - модель безопасности и мощные инструментальные средства разработки.
A. Верно только 1, 2
B. Верно только 3, 4
C. Верно только 1, 2, 3
D. Все верно
10.Отличительными особенностями технологии .NET являются:
1 - возможность реализовать проверку типовой безопасности и проверку надежности;
2 - платформа целиком может быть реализована на многих различных типах компьютеров;
3 - имеется единая библиотека классов, используемая всеми языками, которые поддерживает платформа .NET;
4 - приложения, написанные на различных языках программирования платформы .NET, могут быть легко интегрированы друг с другом.
A. Верно только 1, 2

B. Верно только 3, 4
C. Верно только 1, 2. 3
D. Все верно
11.Платформа Grid предназначена для:
1 - обеспечения доступа к приложениям и совместное использование ресурсов распределенных глобальных сетей;
2 - для поддержки общей логики обеспечения безопасности, эффективного управления распределенными ресурсами;
3 - координированного восстановления информационной системы после сбоев;
4 - создания виртуальной сети.
A. Верно только 1, 2
B. Верно только 3, 4
C. Верно только 1, 2, 3
D. Все верно

Правильные ответы: 1-c, 2-a, 3-b, 4-d, 5-a, 6-a, 7-d, 8-b, 9-c, 10-d, 11-b

Тестирование
1.Основополагающими идеями распределенных автоматизированных информационных систем являются:
1) много организационно распределенных пользователей, одновременно работающих с общими данными;
2) много физически распределенных пользователей, одновременно работающих с общей базой данных;
3) логически распределенные данные образуют общую базу данных;
4) физически распределенные данные образуют единое взаимосогласованное целое общую базу данных
A. Верно только 1, 2, 3
B. Верно только 1, 3, 4
C. Верно только 2, 3, 4
D. Все верно
2.Основные принципы создания и функционирования распределенных баз данных:
1) прозрачность расположения данных для пользователя;
2) изолированность пользователей друг от друга ;
3) синхронизация и согласованность состояния данных в любой момент времени;
4) отсутствие центральной ЭВМ.
A. Верно только 1, 2, 3
B. Верно только 1, 3, 4
C. Верно только 2, 3, 4
D. Все верно
3.Дополнительные принципы создания и функционирования распределенных баз данных:
1) локальная автономия;
2) отсутствие центральной ЭВМ;
3) независимость от местоположения;
4) непрерывность функционирования.
A. Верно только 1, 2, 3
B. Верно только 1, 3, 4
C. Верно только 2, 3, 4
D. Все верно
4.Дополнительные принципы создания и функционирования распределенных баз данных:
1) распределенная обработка запросов;

2) централизованное управление транзакциями;
3) независимость от аппаратуры;
4) независимость от типа операционной системы;
A. Верно только 1, 2, 3
B. Верно только 1, 3, 4
C. Верно только 2, 3, 4
D. Все верно
5. Дополнительные принципы создания и функционирования распределенных баз данных:
1) изолированность пользователей друг от друга;
2) распределенная обработка запросов;
3) независимость от коммуникационной сети;
4) независимость от СУБД.
A. Верно только 1, 2, 3
B. Верно только 1, 3, 4
C. Верно только 2, 3, 4
D. Все верно
6. Важнейшую роль в технологии создания и функционирования распределенных баз данных играет техника представлений (Views), которое определяется как . . .
A. сохраняемый в базе данных авторизованный запрос на выборку данных
B. сохраняемый в базе данных авторизованный глобальный запрос на выборку данных
C. сохраняемым в базе данных глобальный запрос на выборку данных
D. часть базы данных отображаемая на экране монитора.
7. НЕ ЯВЛЯЮТСЯ самостоятельными направлениями в технологиях распределенных систем:
A. Технологии реплицирования баз данных
B. Технологии Клиент-сервер
C. Технологии кластеризации
D. Технологии объектного связывания
8. Основными идеями, определяющими клиент-серверные технологии, являются:
1) общие для всех пользователей данные на одном или нескольких серверах;
2) общие для всех пользователей компьютеры;
3) много клиентов на различных компьютерах, параллельно и одновременно обрабатывают общие данные;
4) много клиентов на одном компьютере, последовательно обрабатывают общие данные.
A. Верно только 1, 2, 3
B. Верно только 1, 3, 4
C. Верно только 2, 3
D. Все верно
9. Сервером является
A. Любой компьютер с большим быстродействием и большой памятью.
B. Любая система, процесс, компьютер, владеющие каким-либо вычислительным ресурсом и предоставляющая этот ресурс другим компьютерам.
C. Любая система, процесс, компьютер, запрашивающие у другого компьютера какой-либо ресурс
D. Специализированный компьютер
10. Клиентом системы обработки информации является:
A. Любая система, процесс, компьютер, владеющие каким-либо вычислительным ресурсом и предоставляющая этот ресурс другим компьютерам.

B. Любой компьютер со средним быстродействием и небольшой памятью
C. Любая система, процесс, компьютер, пользователь, запрашивающие у сервера какой-либо ресурс
D. Неспециализированный компьютер
11. Существуют следующие модели технологий Клиент-сервер:
1) модель файлового сервера;
2) модель удаленного доступа к данным;
3) модель сервера базы данных;
4) модель сервера приложений.
A. Верно только 1, 2, 3
B. Верно только 1, 3, 4
C. Верно только 2, 3
D. Все верно
12. Укажите правильное утверждение
A. В модели файлового сервера основные компоненты размещаются на клиентском компьютере
B. В модели файлового сервера основные компоненты размещаются на сервере
C. Недостатком файлового сервера являются ее простота, отсутствие высоких требований к производительности сервера
D. В модели файлового сервера с помощью функций клиента в оперативную память клиентской машин полностью копируется файл базы данных.
13. Недостатками модели файлового сервера являются:
1) высокий сетевой трафик;
2) не высокий сетевой трафик;
3) отсутствие специальных механизмов безопасности файлов базы данных со стороны СУБД;
4) отсутствие высоких требований к производительности сервера.
A. Верно только 1, 2, 3
B. Верно только 1, 3
C. Верно только 2, 3
D. Все верно

Правильные ответы: 1-c, 2-a, 3-b, 4-d, 5-a, 6-a, 7-d, 8-b, 9-c, 10-d 11-c, 12-a 13-a

Тестирование

1. В отношении модели удаленного доступа к данным укажите ПРАВИЛЬНЫЕ утверждения
A. Модель удаленного доступа к данным не учитывает специфики размещения и физического манипулирования данных во внешней памяти для реляционных СУБД.
B. Компонент доступа к данным реализуется в виде самостоятельной программной части СУБД и инсталлируется на компьютере клиента.
C. На клиентских машинах инсталлируются отделенные программные части СУБД, реализующие интерфейсные и прикладные функции.
D. Системный каталог базы данных находится на компьютере клиента.
2. Достоинствами модели удаленного доступа к данным являются:
1) общение клиента с сервером происходит через SQL-инструкции, а с сервера на клиентские компьютеры передаются только результаты обработки;
2) уменьшение загрузки сети по сравнению с моделью файлового сервера
3) Реализация функций ограничений целостности и безопасности данных при совместной работе нескольких пользователей
4) Унификация интерфейса взаимодействия прикладных компонентов информационных систем с общими данными
A. Верно только 1, 2, 3

B. Верно только 1, 3, 4
C. Верно только 2, 3
D. Все верно
3. Укажите верные утверждения в отношении модели сервера базы данных:
1) Основной идеей модели является механизм хранимых процедур.
2) Прикладной компонент полностью размещается на сервере системы.
3) Прикладной компонент полностью выполняется на сервере системы.
4) События, правила и процедуры, характеризующие предметную область АИС, описываются средствами языка SQL и хранятся вместе с данными на клиенте системы
A. Верно только 1, 2, 3
B. Верно только 1, 3, 4
C. Верно только 2, 3
D. Все верно
4. Достоинствами модели сервера базы данных являются:
1) Снижение нагрузки на сеть.
2) Надежность хранения и обработки данных.
3) Эффективно координируется коллективная работа пользователей с общими данными.
4) Снижает требований к вычислительной машине сервера.
A. Верно только 1, 2, 3
B. Верно только 1, 3, 4
C. Верно только 2, 3
D. Все верно
5. Укажите верные утверждения в отношении модели сервера приложений;
1) Прикладной компонент АИС переносится на специализированный сервер системы
2) На клиентских компьютерах располагается только интерфейсная часть системы
3) Выполнение низкоуровневых операций доступа и изменения данных реализуется на SQL-сервере.
4) Монитор транзакций - это программный компонент СУБД, инсталлируемый на сервере приложений.
A. Верно только 1, 2, 3
B. Верно только 1, 3, 4
C. Верно только 2, 3
D. Все верно
6.Укажите верные утверждения в отношении механизма транзакций:
A. Потерянные изменения возникают тогда, когда две транзакции одновременно читают один и тот же объект базы данных.
B. Потерянные изменения возникают тогда, когда две транзакции одновременно изменяют один и тот же объект базы данных.
C. Потерянные изменения возникают тогда, когда две транзакции последовательно читают один и тот же объект базы данных.
D. Потерянные изменения возникают тогда, когда две транзакции последовательно изменяют один и тот же объект базы данных.
7. Укажите верные утверждения в отношении механизма транзакций:
A. Грязные данные возникают тогда, когда одна транзакция изменяет какой-либо объект данных, а другая транзакция в этот момент читает данные из того же объекта.
B. Грязные данные возникают тогда, когда одна транзакция изменяет один объект данных, а другая транзакция в этот момент читает другой объект данных.
C. Грязные данные возникают тогда, когда две транзакции изменяют какой-либо объект данных.
D. Грязные данные возникают тогда, когда одна транзакция читает какой-либо объект данных, а другая

транзакция в этот момент читает другой объект
данных.
8. Укажите верные утверждения в отношении механизма транзакций:
A. Неповторяющиеся чтения возникают тогда, когда одна транзакция читает какой-либо объект базы данных, а другая до завершения первой не успевает изменить данные.
B. Неповторяющиеся чтения возникают тогда, когда одна транзакция читает какой-либо объект базы данных, а другая до завершения первой его изменяет и успешно фиксируется.
C. Неповторяющиеся чтения возникают тогда, когда одна транзакция изменяет один объект базы данных, а другая до завершения первой успешно читает другой объект базы данных.
D. Неповторяющиеся чтения возникают тогда, когда одна транзакция читает один объект базы данных, а другая до завершения первой изменяет и успешно фиксируется читает другой объект базы данных.
9. Механизм изоляции транзакций и преодоления ситуаций несогласованной обработки данных в общем виде основывается на:
A.Технике синхронизации транзакций
B. Технике удаления транзакций
C. Технике сериализации транзакций
D. Технике модификации транзакций
10. Сериализации транзакций может осуществляться в соответствии со следующими подходами:
1) синхронизационные захваты объектов базы данных;
2) асинхронизационные блокировки объектов базы данных;
3) временные метки объектов базы данных;
4) временные захваты объектов базы данных.
A. Верно только 1, 2, 3
B. Верно только 1, 3
C. Верно только 2, 3
D. Все верно
11. Технология объектного связывания данных решает задачу . .
A. обеспечения доступа из одной локальной базы к данным в другой локальной базе данных.
B. обеспечения доступа из серверной базы к данным в другой локальной базе данных.
C. обеспечения доступа из одной локальной базы к данным в серверной базе данных.
D. обеспечения доступа из одной серверной базы к данным в другой серверной базе данных.
12. В технологии реплицирования данных репликой называют . . .
A. копию базы данных используемую для восстановления поврежденной базы данных.
B. копию базы данных для размещения на другом компьютере сети с целью автономной работы пользователей с одинаковыми данными общего пользования.
C. особую копию базы данных в специальном формате
D. копию базы данных расположенную на сервере

Правильные ответы: 1-c, 2-a, 3-b, 4-d, 5-a, 6-a, 7-d, 8-b, 9-c, 10-d 11-c, 12-a

4.2. Оценочные средства промежуточной аттестации

По дисциплине предусмотрен зачет в 5 семестре и экзамен в 6 семестре. Зачет/Экзамен проходит по билетам. В каждом билете один теоретический вопрос и одно практическое задание. Зачет/Экзамен проводится в устной / письменной и компьютерной форме. Оценивается владение материалом, его системное освоение, способность применять нужные знания, навыки и умения при анализе проблемных ситуаций и решении практических заданий.

4.2.1. Устный или письменный ответ на вопрос

4.2.1.1. Порядок проведения.

Промежуточная аттестация нацелена на комплексную проверку освоения дисциплины. Обучающийся получает вопрос(ы)/задание(я) и время на подготовку. Промежуточная аттестация проводится в устной, письменной или компьютерной форме. Оценивается владение материалом, его системное освоение, способность применять нужные знания, навыки и умения при анализе проблемных ситуаций и решении практических заданий.

4.2.1.2. Критерии оценивания.

18-20 баллов ставится, если обучающийся:

В ответе качественно раскрыл содержание темы. Ответ хорошо структурирован. Прекрасно освоен понятийный аппарат. Продемонстрирован высокий уровень понимания материала. Превосходное умение формулировать свои мысли, обсуждать дискуссионные положения.

14-17 баллов ставится, если обучающийся:

Основные вопросы темы раскрыл. Структура ответа в целом адекватна теме. Хорошо освоен понятийный аппарат. Продемонстрирован хороший уровень понимания материала. Хорошее умение формулировать свои мысли, обсуждать дискуссионные положения.

11-13 баллов ставится, если обучающийся:

Тему частично раскрыл. Ответ слабо структурирован. Понятийный аппарат освоен частично. Понимание отдельных положений из материала по теме. Удовлетворительное умение формулировать свои мысли, обсуждать дискуссионные положения.

0--10 баллов ставится, если обучающийся:

Тему не раскрыл. Понятийный аппарат освоен неудовлетворительно. Понимание материала фрагментарное или отсутствует. Неумение формулировать свои мысли, обсуждать дискуссионные положения.

4.2.1.3. Оценочные средства.

Вопросы для устного или письменного ответа на зачет

1. Что такое промышленный программный продукт. Дать определения пакета прикладных программ, программной системы.
2. Жизненный цикл программного обеспечения. Дать краткую характеристику каждого этапа.
3. Почему программные системы сложны. Привести пять признаков сложной системы.
4. Техническое задание. Перечислить и охарактеризовать разделы, входящие в техническое задание.
5. Унифицированный процесс разработки программного обеспечения. Жизненный цикл унифицированного процесса.
6. Работа с кадрами. Перечислить роли разработчиков и дать характеристику каждой из них.
7. Дать определения проекта, процесса, продукта с точки зрения унифицированного процесса разработки программного обеспечения.
8. Что такое артефакт. В чем преимущества организованного процесса разработки программного обеспечения.
9. Использование языка UML при проектировании сложных программных систем. Какие диаграммы используются в UML для создания моделей программной системы.
10. Диаграмма вариантов использования, ее назначение. Рассказать о варианте использования и действующем лице. Правила построения диаграммы вариантов использования.
11. Понятие класса и объекта. Что может быть объектом. Что такое атрибут и операция.
12. Пять критериев проверки правильности построения класса.
13. Что такое классификация с точки зрения объектно-ориентированного проектирования программных систем. Теории классификации.
14. Методы классификации.

Вопросы для устного или письменного ответа на экзамен

1. Микропроцесс проектирования. Перечислить этапы и основные виды деятельности выполняемые на каждом из них.
2. Микропроцесс проектирования первый этап.
3. Микропроцесс проектирования второй этап.
4. Микропроцесс проектирования третий этап.
5. Микропроцесс проектирования четвертый этап.
6. Диаграммы взаимодействия. Основное назначение.
7. Диаграмма классов. Ее назначение. Что она включает. Рассказать об основных видах связей между классами.
8. Дать определение тестированию и отладке. Особенности и объекты тестирования. Автономное и комплексное тестирование.
9. Дать определение тестированию и отладке. Направления тестирования. Стратегия тестирования. Контрольный лист тестирования модуля.
10. Дать определение тестированию и отладке. Локализация ошибок. Классификация ошибок. Безопасное программирование.
11. Оценки ошибок.
12. Документирование. Состав и содержание документов прилагаемых к программной системе.

13.Внедрение программного комплекса. Планирование испытаний.

14.Внедрение программного комплекса. Подготовка тестовых данных. Анализ результатов испытаний.

15.Что такое качество с точки зрения квалиметрии. Дать определение свойству и показателю качества ПО. Основные задачи решаемые при оценке качества.

16.Оценка качества программного обеспечения. Методы оценки свойств программного обеспечения.

4.2.2. Практическое задание

4.2.2.1. Порядок проведения.

Практические навыки проверяются путём выполнения обучающимися практических заданий в условиях, полностью или частично приближенных к условиям профессиональной деятельности. Проверяется знание теоретического материала, необходимое для правильного совершения необходимых действий, умение выстроить последовательность действий, практическое владение приёмами и методами решения профессиональных задач.

4.2.2.2. Критерии оценивания

27-30 баллов ставится, если обучающимся:

Задание выполнено полностью и правильно.

22-26 баллов ставится, если обучающимся:

Задание выполнено полностью, но нет достаточного обоснования. Или при верном решении допущена ошибка или недочет, не влияющий на правильную последовательность рассуждений.

18-21 баллов ставится, если обучающимся:

Задание выполнено частично или с фактическими ошибками.

0-17 баллов ставится, если обучающимся:

Задание не выполнено или выполнено с большим количеством фактических ошибок.

4.2.2.3. Содержание оценочного средства

1. Описать требования к программному обеспечению информационной системы магазина розничной торговли
2. Описать требования к программному обеспечению информационной системы сети магазинов.
3. Описать требования к программному обеспечению интернет магазина.
4. Описать требования к программному обеспечению информационной системы туристического агентства.
5. Описать требования к программному обеспечению информационной системы агентства недвижимости.
6. Описать требования к программному обеспечению информационного сайта медицинского учреждения.
7. Описать требования к программному обеспечению информационного сайта учебного заведения.
8. Описать модель предметной области для информационной системы магазина розничной торговли.
9. Описать качество программного обеспечения для информационной системы сети магазинов.
10. Описать качество программного обеспечения для сайта интернет магазина.
11. Описать качество программного обеспечения для туристического агентства.
12. Организация деятельности администрации гостиницы.
13. Организация работы службы автоинспекции.
14. Деятельность налоговой службы.
15. Организация работы службы социальной помощи.
16. Деятельность абонентской службы АТС.
17. Организация работы рекламного агентства.
18. Деятельность службы трудоустройства.
19. Организация работы службы общественного питания.
20. Организация работы службы скорой помощи.
21. Деятельность фирмы бартерного обмена.)

Приложение 2
к рабочей программе дисциплины (модуля)
Б1.О.05.03 Программная инженерия

Перечень литературы, необходимой для освоения дисциплины (модуля)

Направление подготовки: 09.03.03 - Прикладная информатика

Профиль подготовки: Прикладная информатика в экономике

Квалификация выпускника: бакалавр

Форма обучения: очное

Язык обучения: русский

Год начала обучения по образовательной программе: 2025

Основная литература:

1. Введение в программную инженерию : учебник / В.А. Антипов, А.А. Бубнов, А.Н. Пылькин, В.К. Столчнев. — Москва : КУРС : ИНФРА-М, 2025. — 336 с. - ISBN 978-5-906923-22-6. - Текст : электронный. - URL: https://znanium.ru/catalog/product/2173919 (дата обращения: 07.02.2025). – Режим доступа: по подписке.
2. Волк, В. К. Введение в программную инженерию : учебное пособие / В. К. Волк. — Курган : КГУ, 2018. — 156 с. — ISBN 978-5-4217-0452-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/177902 (дата обращения: 07.02.2025). — Режим доступа: для авториз. пользователей.
3. Волк, В. К. Практическое введение в программную инженерию : учебное пособие для вузов / В. К. Волк. — 2-е изд., стер. — Санкт-Петербург : Лань, 2022. — 100 с. — ISBN 978-5-507-44920-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/249848 (дата обращения: 07.02.2025). — Режим доступа: для авториз. пользователей.
4. Маран, М. М. Программная инженерия : учебное пособие для вузов / М. М. Маран. — 3-е изд., стер. — Санкт-Петербург : Лань, 2022. — 196 с. — ISBN 978-5-8114-9323-4. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/189470 (дата обращения: 07.02.2025). — Режим доступа: для авториз. пользователей.

Дополнительная литература:

1. Птицына, Л. К. Методология программной инженерии : учебное пособие / Л. К. Птицына, И. Б. Бондаренко. — Санкт-Петербург : СПбГУТ им. М.А. Бонч-Бруевича, 2021. — 55 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/279605 (дата обращения: 07.02.2025). — Режим доступа: для авториз. пользователей.
2. Антамошкин, О. А. Программная инженерия. Теория и практика [Электронный ресурс] : учебник / О. А. Антамошкин. - Красноярск: Сиб. Федер. ун-т, 2012. - 247 с. - ISBN 978-5-7638-2511-4. - Текст : электронный. - URL: https://znanium.com/catalog/product/492527 (дата обращения: 07.02.2025). – Режим доступа: по подписке.
3. Доррер, Г. А. Методология программной инженерии : учебное пособие / Г. А. Доррер. — Красноярск : СибГУ им. академика М. Ф. Решетнёва, 2021. — 190 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/195097 (дата обращения: 07.02.2025). — Режим доступа: для авториз. пользователей.

Приложение 3
к рабочей программе дисциплины (модуля)
Б1.О.05.03 Программная инженерия

Перечень информационных технологий, используемых для освоения дисциплины (модуля), включая перечень программного обеспечения и информационных справочных систем

Направление подготовки: 09.03.03 - Прикладная информатика

Профиль подготовки: Прикладная информатика в экономике

Квалификация выпускника: бакалавр

Форма обучения: очное

Язык обучения: русский

Год начала обучения по образовательной программе: 2025

Освоение дисциплины (модуля) предполагает использование следующего программного обеспечения и информационно-справочных систем:

Office Professional Plus 2010,
GIMP,
Inkscape,
Notepad ++,
Python,
Lazarus
Электронная библиотечная система «ZNANIUM.COM»
Электронная библиотечная система Издательства «Лань»
Электронная библиотечная система «Консультант студента»