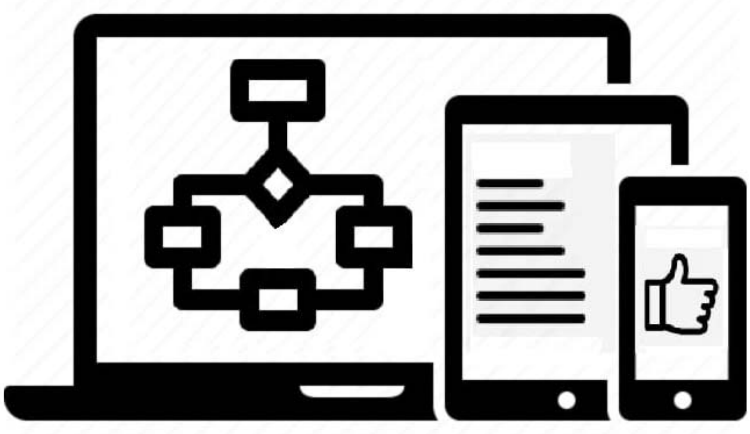
МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Программное обеспечение информационных технологий»

УПРАВЛЕНИЕ IT-ПРОЕКТАМИ

Методические рекомендации к лабораторным работам для студентов направлений подготовки 09.03.01 «Информатика и вычислительная техника», 09.03.04 «Программная инженерия» и 01.03.04 «Прикладная математика» очной формы обучения



Могилев 2026

УДК 004.4(07)
ББК 32.973я73
У67

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Программное обеспечение информационных технологий» «31» августа 2026 г., протокол № 1

Составитель канд. техн. наук Ю. В. Вайнилович

Рецензент канд. техн. наук, доц. В. М. Ковальчук

Методические рекомендации содержат требования к лабораторным работам по дисциплине «Управление IT-проектами» для студентов направлений подготовки 09.03.01 «Информатика и вычислительная техника», 09.03.04 «Программная инженерия» и 01.03.04 «Прикладная математика» очной формы обучения.

Учебное издание

УПРАВЛЕНИЕ IT-ПРОЕКТАМИ

Ответственный за выпуск	В. В. Кутузов
Корректор	А. Т. Червинская
Компьютерная верстка	М. М. Дударева

Подписано в печать 23.10.2026. Формат 60×84/16. Бумага офсетная. Гарнитура Таймс. Печать трафаретная. Усл. печ. л. 1,63. Уч.-изд. л. 1,63. Тираж 21 экз. Заказ № 771.

Издатель и полиграфическое исполнение:
Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 07.03.2019.
Пр-т Мира, 43, 212022, г. Могилев.

© Белорусско-Российский
университет, 2026

Содержание

Введение 4
Общее задание на выполнение лабораторных работ 5
Теоретические сведения 6
1 Лабораторная работа № 1. Описание компонентов IT-проекта как организационно-технической системы 17
2 Лабораторная работа № 2. Описание компонентов системы управления IT-проектом 18
3 Лабораторная работа № 3. Оценка трудоемкости IT-проекта 19
4 Лабораторная работа № 4. Формирование состава и структуры команды исполнителей IT-проекта 20
5 Лабораторная работа № 5. Декомпозиция задач IT-проекта на подзадачи 21
6 Лабораторная работа № 6. Распределение исполнителей IT-проекта на задачи 22
7 Лабораторная работа № 7. Оценка эффективности системы управления IT-проектом 24
Список литературы 25
Приложение А. Варианты заданий для выполнения лабораторных работ . 26

Введение

Цель учебной дисциплины состоит в формировании у студентов комплекса знаний, умений и навыков в области планирования, организации и контроля IT-проектов с использованием современных методологий и инструментов, необходимых для эффективного управления разработкой программного обеспечения и цифровых продуктов.

Методические рекомендации по дисциплине «Управление IT-проектами» к лабораторным работам предназначены для оказания помощи студентам при выполнении лабораторных работ по данной дисциплине. Они содержат задания для самостоятельного выполнения, инструкции, пояснения и рекомендации, которые помогают студентам освоить современные методологии и инструменты управления разработкой программного обеспечения и цифровых продуктов.

Общее задание на выполнение лабораторных работ

Лабораторные работы № 1–7 выполняются студентом последовательно и образуют единый сквозной практикум: каждая последующая работа опирается на результаты предыдущих. Перед началом выполнения лабораторных работ студент выбирает вариант индивидуального задания – предметную область IT-проекта – из перечня, приведенного в приложении А, либо получает вариант от преподавателя.

Общая цель практикума – на примере выбранной предметной области спроектировать и оформить комплексный документ «План управления проектом» для условного IT-проекта по разработке информационной системы. Начиная с лабораторной работы № 1 и до лабораторной работы № 7 студент поэтапно заполняет разделы этого документа: результаты каждой работы становятся исходными данными для следующей – так, WBS, построенная в лабораторной работе № 2, используется для оценки трудоемкости в работе № 3, оценки трудоемкости – для построения календарного плана в работе № 5, а структура команды, определенная в работе № 4, – для назначения ресурсов на задачи в работе № 6.

Соответствие лабораторных работ и разделов документа «План управления проектом»:

– лабораторная работа № 1 «Описание компонентов IT-проекта как организационно-технической системы» – разделы документа 1–3: устав проекта, реестр заинтересованных сторон, обоснование выбора методологии;

– лабораторная работа № 2 «Описание компонентов системы управления IT-проектом» – раздел документа 4: структурная декомпозиция работ (WBS);

– лабораторная работа № 3 «Оценка трудоемкости IT-проекта» – раздел документа 5: оценка трудоемкости;

– лабораторная работа № 4 «Формирование состава и структуры команды исполнителей IT-проекта» – раздел документа 6: команда проекта (структура, матрица навыков, матрица ответственности RACI);

– лабораторная работа № 5 «Декомпозиция задач IT-проекта на подзадачи» – раздел документа 7: календарный план и файл MS Project;

– лабораторная работа № 6 «Распределение исполнителей IT-проекта на задачи» – раздел документа 8: план управления ресурсами;

– лабораторная работа № 7 «Оценка эффективности системы управления IT-проектом» – разделы документа 9–12: реестр рисков, план управления качеством, план коммуникаций, показатели эффективности и мониторинг.

По результатам выполнения каждой лабораторной работы студент дополняет общий отчет соответствующим заполненным разделом, а по результатам лабораторных работ № 5 и 6 дополнительно оформляет и прикладывает файл MS Project (.mpp) с календарным планом. Итогом выполнения всего цикла лабораторных работ является полностью укомплектованный документ «План управления проектом», объединяющий разделы документа 1–12 для выбранного варианта задания.

Теоретические сведения

Назначение и структура документа «План управления проектом».

План управления проектом – это формальный, утвержденный документ, который определяет, как проект будет исполняться, контролироваться и завершаться. Он объединяет и согласовывает результаты всех процессов планирования (по содержанию, срокам, ресурсам, стоимости, качеству, рискам и коммуникациям) в единый непротиворечивый документ, которым руководствуются все участники проекта.

Важно различать устав проекта и план управления проектом. Устав проекта отвечает на вопрос «зачем и в каких границах выполняется проект» и авторизует его существование, тогда как план управления проектом отвечает на вопрос «как именно проект будет спланирован, выполнен, проконтролирован и закрыт», т. е. детализирует подходы и процессы по каждой предметной области управления. После утверждения план (в первую очередь его базовые показатели по содержанию, срокам и стоимости – так называемые baseline) становится точкой отсчета: фактический ход работ сравнивается именно с ним, а не с первоначальными ожиданиями заказчика или интуитивными представлениями команды.

На практике план управления проектом редко пишется полностью «за один присест». Как правило, применяется принцип последовательной разработки (progressive elaboration, «планирование методом набегающей волны»): ближайшие по времени работы планируются детально, а более отдаленные – укрупненно, с последующим уточнением по мере поступления информации. В рамках данного практикума этот принцип реализуется естественным образом: документ формируется постепенно, раздел за разделом, от лабораторной работы к лабораторной работе.

Рекомендации по выполнению:

– заведите единый файл документа «План управления проектом» сразу после выбора варианта задания и добавляйте в него новый раздел после каждой лабораторной работы – не собирайте документ из разрозненных файлов в самом конце;

– после заполнения очередного раздела возвращайтесь к уже написанным разделам и проверяйте их на непротиворечивость: например, состав ролей в разделе документа 6 «Команда проекта» должен совпадать с ресурсами, которые назначаются на задачи в разделе документа 8;

– формулируйте план управления проектом для конкретной предметной области своего варианта задания (приложение А), а не абстрактно – это облегчит защиту лабораторных работ и сделает документ более убедительным;

– ведите короткий журнал допущений и открытых вопросов по проекту – многие детали, которые кажутся очевидными в начале, потребуется уточнить при выполнении последующих лабораторных работ.

Устав проекта и заинтересованные стороны.

Устав проекта (Project Charter) – документ, формально авторизующий существование проекта и наделяющий руководителя проекта полномочиями использовать ресурсы организации для выполнения работ. Типовой устав включает бизнес-обоснование (почему проект нужен организации и какую проблему или возможность он закрывает), цели проекта, сформулированные по критериям SMART (конкретные, измеримые, достижимые, значимые, ограниченные по времени), основные результаты (deliverables), критерии успеха проекта, ограничения (бюджетные, временные, технологические), допущения, принятые при планировании, а также риски высокого уровня, которые уже видны на старте проекта.

Заинтересованные стороны (stakeholders) – лица, группы или организации, которые могут влиять на проект, подвергаться воздействию его результатов или считать себя таковыми. К ним относятся не только заказчик и конечные пользователи, но и спонсор проекта, служба поддержки, смежные подразделения, регуляторы, поставщики и даже конкуренты, если их действия способны повлиять на проект. Анализ заинтересованных сторон оформляется в виде реестра, где для каждой стороны фиксируются ее интересы, ожидания, уровень влияния на проект и уровень заинтересованности в его результатах.

Наиболее распространенный инструмент приоритизации stakeholders – матрица Power/Interest Grid (влияние × интерес), которая делит все стороны на четыре группы и предполагает для каждой из них свою стратегию вовлечения:

– высокое влияние и высокий интерес – управлять тесно (Manage Closely): регулярные встречи, вовлечение в ключевые решения;

– высокое влияние и низкий интерес – поддерживать удовлетворенность (Keep Satisfied): своевременно информировать о значимых изменениях, не перегружая деталями;

– низкое влияние и высокий интерес – держать в курсе (Keep Informed): регулярные, но менее формальные обновления;

– низкое влияние и низкий интерес – мониторить (Monitor): минимальные усилия, периодическая проверка изменений статуса.

Рекомендации по выполнению:

– формулируя цели по SMART, проверяйте каждую цель по всем пяти критериям явно – начинающие часто пропускают «измеримость» или «ограниченность по времени», ограничиваясь общей формулировкой;

– составляя реестр stakeholders (не менее 8 позиций), не ограничивайтесь очевидными ролями – подумайте про регуляторов, службу поддержки, смежные отделы, поставщиков технологий;

– для каждой позиции в матрице Power/Interest указывайте не только квадрант, но и короткое обоснование – почему сторона попала именно туда, а не в соседний квадрант;

– выбирайте стратегию вовлечения осознанно и привязывайте ее к квадранту матрицы; избегайте одинаковой стратегии для всех сторон.

Методологии управления IT-проектом.

Методология управления проектом определяет подход к планированию, организации и контролю работ. Каскадная модель (Waterfall) предполагает последовательное прохождение четко определенных фаз (анализ, проектирование, разработка, тестирование, внедрение), где переход к следующей фазе возможен только после завершения и приемки предыдущей. Такой подход хорошо подходит для проектов со стабильными, заранее известными требованиями и жесткими регуляторными или контрактными ограничениями.

Гибкие методологии (Agile) строятся на итеративной разработке короткими циклами, постоянной обратной связи с заказчиком и адаптации к изменяющимся требованиям. Наиболее распространены два фреймворка: Scrum, работающий фиксированными по длительности спринтами с четкими ролями (Product Owner, Scrum Master, команда разработки) и церемониями (планирование спринта, ежедневные встречи, обзор и ретроспектива), и Kanban, который делает акцент на визуализации потока работ, ограничении количества задач в работе (WIP-лимитах) и непрерывной, а не циклической поставке результата.

Гибридный подход (Hybrid) сочетает элементы обеих моделей – например, каскадное планирование верхнего уровня (укрупненные фазы, контрольные точки, бюджет) с гибкой разработкой внутри фаз. Такой подход часто применяется в крупных организациях, где отчетность перед заказчиком должна строиться вокруг фиксированных вех, а сама разработка ведется итеративно.

Рекомендации по выполнению:

– обосновывайте выбор методологии через конкретные характеристики именно своего варианта задания – степень определенности требований, размер и распределенность команды, критичность сроков, – а не общими фразами вроде «Agile гибче»;

– если выбран гибридный подход, явно укажите, какие фазы или элементы проекта ведутся каскадно, а какие – итеративно, и почему именно так;

– сверяйте выбранную методологию с содержанием следующих разделов документа: если выбран Scrum, в разделе документа 5 «Оценка трудоемкости» уместно использовать story points и Planning Poker наравне с PERT.

Структурная декомпозиция работ (WBS).

WBS (Work Breakdown Structure, иерархическая структура работ) – это иерархическая декомпозиция общего содержания работ, которые должна выполнить команда проекта для достижения целей проекта и создания требуемых результатов. Декомпозиция может строиться разными способами: по фазам жизненного цикла проекта, по основным результатам (модулям информационной системы) или по видам деятельности (анализ, разработка, тестирование, документирование). Для IT-проектов чаще всего используется декомпозиция по результатам с последующей детализацией по видам работ внутри каждого результата.

Построение WBS необходимо для того, чтобы разбить сложный проект на управляемые компоненты, исключить дублирование и упущение работ (правило 100 % – WBS должна охватывать все содержание проекта, включая работы по управлению проектом, а не только «видимую» разработку), а также создать основу для последующей оценки трудоемкости, календарного планирования и распределения ответственности. Нижний, самый детальный уровень декомпозиции называется пакетом работ (work package) – это элемент, трудоемкость и длительность которого можно достоверно оценить, а результат – однозначно проверить.

Элементы WBS принято нумеровать иерархически (например, 1, 1.1, 1.1.1) – такая нумерация (WBS Code) используется впоследствии и в календарном плане, и в отчетности по проекту. Для описания содержания каждого элемента используется словарь WBS (WBS Dictionary), включающий код, наименование, описание работ, ожидаемый результат и критерии приемки; сама WBS может визуализироваться в виде иерархической таблицы или дерева. Распространенная ошибка начинающих – формулировать элементы WBS как процессы («тестирование», «разработка»), а не как результаты («протестированный модуль авторизации»); второй подход упрощает последующий контроль завершенности работ.

Рекомендации по выполнению:

– стройте декомпозицию «сверху вниз»: сначала крупные фазы или модули системы, затем – детализация до задач; так проще не упустить работы и сохранить целостную картину проекта;

– проверяйте себя правилом 100 % на каждом уровне: сумма дочерних элементов должна полностью покрывать содержание родительского элемента, без пробелов и без пересечений;

– не забывайте про «невидимые» работы – управление проектом, документирование, тестирование, обучение пользователей – а не только про разработку функционала;

– формулируйте элементы нижнего уровня как результаты, а не как процессы, и убедитесь, что для каждого пакета работ можно четко сформулировать критерий приемки.

Трудоемкость IT-проекта и методы ее оценки.

Трудоемкость работы – это количество труда (человеко-часов, человеко-дней), необходимое для выполнения работы одним исполнителем при заданной производительности. Трудоемкость не следует путать с длительностью работы: длительность зависит от количества привлеченных исполнителей и их доступности, тогда как трудоемкость – это объем труда как таковой. Так, работу трудоемкостью 80 человеко-часов один исполнитель, занятый полный день, выполнит за 10 рабочих дней, а два исполнителя, разделив работу поровну, – примерно за 5.

Оценка трудоемкости необходима для расчета стоимости и бюджета проекта (через ставки исполнителей), определения состава и численности

команды, построения реалистичного календарного плана и заблаговременного выявления перегрузки ресурсов – то есть служит фундаментом практически для всех последующих разделов плана управления проектом.

На практике применяется несколько методов оценки трудоемкости:

– экспертная оценка – на основе мнения одного или нескольких экспертов, обладающих опытом выполнения аналогичных работ;

– оценка по аналогам – используется трудоемкость похожих работ, выполненных ранее, с поправкой на отличия текущего проекта;

– параметрическая оценка – трудоемкость рассчитывается на основе статистической зависимости между историческими данными и параметрами проекта (например, количеством экранных форм, интеграций и т. п.);

– метод трехточечной оценки (PERT) – для каждой задачи определяются оптимистическая (O), наиболее вероятная (M) и пессимистическая (P) оценки, на основе которых рассчитываются ожидаемая трудоемкость и стандартное отклонение по формулам Expected (E) = (O + 4M + P) / 6 и Standard Deviation (σ) = (P − O) / 6; чем больше разброс между O и P, тем выше неопределенность оценки и тем больше значение σ;

– Planning Poker – групповой метод оценки в Agile-командах: каждый участник независимо выбирает карту с оценкой (обычно в story points по шкале Фибоначчи), после чего расхождения обсуждаются до консенсуса; такой подход снижает эффект «якорения» на чужом мнении;

– декомпозиция (bottom-up estimating) – трудоемкость крупной работы оценивается путем суммирования трудоемкости составляющих ее более мелких задач WBS; как правило, дает наиболее точный результат, но требует детально проработанной WBS.

Полученные по каждой задаче оценки суммируются, переводятся в человеко-часы и человеко-дни и используются для определения прогнозной длительности проекта с учетом доступности ресурсов. Важно понимать, что сумма стандартных отклонений отдельных задач позволяет оценить не только ожидаемую трудоемкость проекта в целом, но и степень уверенности в этой оценке.

Рекомендации по выполнению:

– оценивайте методом PERT только ключевые или наиболее неопределенные задачи из WBS, а не все подряд – это соответствует формулировке задания и экономит время;

– сверяйте итоговую оценку в человеко-часах со здравым смыслом: для учебного проекта суммарная трудоемкость в тысячи человеко-часов, скорее всего, свидетельствует об ошибке масштаба или избыточной детализации WBS;

– если выбран Agile-подход, используйте Planning Poker и story points как основной метод, но при этом отдельно рассчитайте PERT для 3–5 самых рискованных задач – так вы покажете владение обоими подходами;

– не забывайте указывать не только Expected, но и стандартное отклонение – это ожидается заданием и показывает понимание неопределенности оценки.

Команда проекта: роли, FTE, матрица навыков, RACI-матрица.

Формирование команды проекта начинается с определения необходимых ролей и требований к ним (например, аналитик, backend-разработчик, frontend-разработчик, тестировщик, руководитель проекта), после чего рассчитывается размер команды в единицах FTE (Full-Time Equivalent) – эквиваленте полной занятости одного сотрудника. Единица FTE позволяет сопоставлять вклад сотрудников с разной степенью занятости в проекте: например, два сотрудника, занятых в проекте на 50 %, в сумме дают 1 FTE. На основе ролей строится организационная структура команды (org chart), фиксирующая подчиненность и взаимодействие участников.

Матрица навыков (Skills Matrix) отражает уровень владения (обычно по шкале от 0 – «нет навыка» – до 4 – «эксперт») каждым необходимым навыком у каждого члена команды. Сопоставление требуемых и имеющихся навыков (gap-анализ) позволяет выявить дефицит компетенций и спланировать его устранение – обучение действующих сотрудников или найм новых. Стоит учитывать, что команды на практике проходят через стадии развития (формирование, конфликты, нормализация, результативная работа – модель Такмена), и на ранних стадиях производительность команды может быть ниже расчетной.

Распределение ответственности за работы проекта оформляется в виде RACI-матрицы, где для каждой работы указываются: R (Responsible) – исполнитель, непосредственно выполняющий работу; A (Accountable) – ответственный за конечный результат (на одну работу допускается только один Accountable, иначе размывается ответственность); C (Consulted) – эксперт, с которым консультируются перед принятием решения (двусторонняя коммуникация); I (Informed) – участник, которого информируют о ходе или результате работы (односторонняя коммуникация). Иногда используется расширенный вариант матрицы – RASCI, где отдельно выделяется роль Support (поддерживающий исполнитель).

Рекомендации по выполнению:

– начинайте заполнение RACI-матрицы с самых крупных и рискованных работ – так быстрее наберется минимум 15 работ, требуемых заданием, и матрица получится содержательной, а не формальной;

– проверяйте матрицу на «пустые» и «задвоенные» ячейки: у каждой работы должен быть ровно один Accountable, и не должно быть работ без ответственного вовсе;

– сопоставляйте состав ролей команды (раздел документа 6) с элементами WBS (раздел документа 4) – на каждую значимую группу работ должна найтись роль, способная ее выполнить;

– при расчете FTE не забывайте, что руководитель проекта и аналитик часто заняты не на 100 %, а разделяют время между несколькими задачами или проектами.

Календарное планирование и критический путь.

Календарный план проекта устанавливает даты начала и окончания работ с учетом их длительности и зависимостей друг от друга. Между задачами выделяют четыре типа зависимостей: FS (finish-to-start – окончание предыдущей задачи перед началом следующей, наиболее распространенный тип), SS (start-to-start – одновременное начало), FF (finish-to-finish – одновременное окончание) и SF (start-to-finish – начало предыдущей задачи перед окончанием следующей, применяется редко). Отдельные значимые точки плана, не имеющие длительности (например, «завершение этапа тестирования» или «релиз версии 1.0»), выделяются как вехи (milestones).

Критический путь проекта – это последовательность взаимосвязанных задач, определяющая минимально возможную длительность проекта: любая задержка задачи на критическом пути напрямую сдвигает срок окончания проекта. Метод расчета критического пути (Critical Path Method, CPM) основан на вычислении раннего и позднего начала/окончания каждой задачи; разница между ранним и поздним сроком задачи и есть ее резерв времени (float). Для задач вне критического пути резерв положителен – их можно задержать без влияния на общий срок проекта, тогда как для задач на критическом пути резерв равен нулю.

Утвержденный календарный план фиксируется как базовый план (baseline) и в дальнейшем используется для сравнения с фактическим ходом работ. Для построения календарного плана, определения критического пути и визуализации диаграммы Ганта в рамках практикума применяется MS Project, где для каждой задачи можно задать тип планирования (фиксированная длительность, фиксированный объем работ или фиксированное количество исполнителей) – это влияет на то, как программа пересчитывает даты при изменении назначений ресурсов. Если после первичного расчета срок проекта не устраивает заказчика, применяют методы сжатия расписания: fast tracking (распараллеливание задач, ранее выполнявшихся последовательно) или crashing (добавление ресурсов на задачи критического пути, как правило, с увеличением стоимости).

Рекомендации по выполнению:

– прежде чем расставлять зависимости в MS Project, набросайте логику последовательности задач на бумаге или прямо в структуре WBS – это ускорит дальнейшую работу в самом инструменте;

– используйте зависимости FS как основной тип и прибегайте к SS/FF только там, где это оправдано содержанием работ – избыточное количество нестандартных зависимостей затрудняет чтение плана;

– обязательно сохраните baseline сразу после согласования плана – без него не с чем будет сравнивать фактический прогресс в последующих лабораторных работах;

– проверяйте длительность каждой задачи на согласованность с трудоемкостью, рассчитанной в разделе документа 5: длительность задачи должна логично следовать из трудоемкости и количества назначенных исполнителей.

Управление ресурсами проекта.

Управление ресурсами включает назначение исполнителей на задачи календарного плана и планирование их загрузки. В MS Project принято различать трудовые ресурсы (work resources – люди), материальные ресурсы (material resources – расходуемые в ходе проекта материалы) и затратные ресурсы (cost resources – разовые статьи расходов, не привязанные к объему работ, например, стоимость лицензии на программное обеспечение).

Доступная емкость (capacity) команды по периодам рассчитывается исходя из числа рабочих дней в периоде, длительности рабочего дня и накладных расходов (например, времени на совещания и административные задачи, выражаемого в процентах от рабочего времени). Сопоставление доступных и запланированных часов по каждому ресурсу представляется в виде ресурсных гистограмм (Resource Histograms), которые наглядно показывают периоды перегрузки ресурса (Resource Overallocation) или его недозагрузки (Underutilized). Перегрузка чаще всего возникает из-за того, что один и тот же исполнитель одновременно назначен на несколько параллельных задач с суммарной загрузкой более 100 %.

Устранение выявленной перегрузки выполняется путем выравнивания ресурсов (Resource Leveling) – перераспределения задач во времени (сдвиг задачи в пределах имеющегося резерва без изменения срока проекта), изменения назначений (замена или добавление исполнителя) или, в крайнем случае, привлечения дополнительных ресурсов. Выравнивание может выполняться автоматически средствами MS Project либо вручную – второй вариант дает больше контроля над тем, какие именно задачи будут сдвинуты.

Рекомендации по выполнению:

– сначала назначьте ресурсы на все задачи, и только потом анализируйте гистограммы – иначе трудно понять, откуда взялась перегрузка и по какому именно ресурсу;

– если перегрузка возникла у нескольких ролей одновременно, начинайте выравнивание с самой дефицитной или дорогой роли – это обычно дает наибольший эффект при наименьших затратах;

– указывайте в анализе загрузки не только сам факт перегрузки, но и конкретные предложенные меры (перераспределение, изменение сроков, добавление ресурсов) применительно к своему проекту, а не в общем виде;

– проверяйте, что накладные расходы (%), заложенные при расчете доступных часов, реалистичны для вашей команды – типовое значение 10 %...20 % на совещания и административные задачи.

Управление рисками.

Риск проекта – это неопределенное событие или условие, которое в случае наступления оказывает положительное или отрицательное воздействие на цели проекта. Управление рисками включает их идентификацию, качественный анализ (оценку вероятности и влияния каждого риска, как правило, по шкале от 1 до 5) и построение матрицы рисков «вероятность × влияние», которая позволяет ранжировать риски и выделить наиболее критичные из них. Для

более зрелых проектов дополнительно может проводиться количественный анализ рисков, при котором степень их влияния выражается в конкретных денежных или временных величинах, однако для целей данного практикума достаточно качественной оценки.

Для приоритетных рисков разрабатываются стратегии реагирования: для угроз – уклонение (изменение плана так, чтобы исключить риск), передача (страхование, аутсорсинг), снижение (меры, уменьшающие вероятность или влияние) или принятие (осознанное решение ничего не предпринимать, обычно с формированием резерва на непредвиденные обстоятельства); для благоприятных возможностей – использование, усиление, разделение с партнером или принятие. Реестр рисков фиксирует все выявленные риски, их оценку, назначенного владельца риска, триггеры (признаки приближающегося наступления риска) и планируемые меры реагирования; реестр регулярно актуализируется на протяжении всего проекта, а не составляется один раз в начале.

Рекомендации по выполнению:

– формулируйте риски по схеме «событие + причина + возможное последствие» (например, «из-за нехватки backend-разработчиков модуль оплаты может быть сдан с задержкой»), а не общими словами вроде «риск срыва сроков» – так проще оценивать риск и работать с ним;

– не подбирайте одинаковую стратегию реагирования для всех рисков – обоснуйте выбор именно этой стратегии для каждого из топ-5 рисков применительно к своему проекту;

– ищите риски не только в технической области (сложность интеграции, незнакомый стек), но и в организационной (доступность экспертов, зависимость от внешних поставщиков, изменение требований заказчика);

– назначайте владельца для каждого приоритетного риска – того, кто отвечает за отслеживание триггеров и выполнение мер реагирования.

Управление качеством.

Управление качеством в IT-проекте направлено на обеспечение соответствия результатов проекта установленным требованиям и ожиданиям заказчика. Принято различать обеспечение качества (quality assurance) – совокупность плановых действий, направленных на предотвращение дефектов (стандарты, регламенты, обучение), и контроль качества (quality control) – проверку уже созданных результатов на соответствие требованиям (тестирование, code review, приемочные испытания). Дешевле предотвратить дефект, чем найти и исправить его на поздних этапах – эта идея лежит в основе понятия стоимости качества (cost of quality), которая складывается из затрат на предупреждение дефектов, затрат на их выявление (проверки, тестирование) и затрат на исправление уже обнаруженных дефектов.

Для проекта определяются стандарты качества и измеримые метрики (например, показатели по качеству кода, количеству дефектов на релиз, доле покрытия автоматическими тестами) с целевыми значениями. Критерий

Definition of Done (DoD) формально определяет условия, при которых результат работы считается завершенным, и может формулироваться на разных уровнях – для отдельной задачи, пользовательской истории (user story), спринта и релиза; чем выше уровень, тем более комплексные критерии он включает (например, DoD релиза дополнительно требует прохождения регрессионного тестирования и обновления документации). Дополнительным инструментом контроля качества служит чек-лист code review, применяемый при проверке исходного кода перед его включением в основную ветку разработки; для полноты картины полезно также упомянуть уровни тестирования – модульное, интеграционное, системное и приемочное, на которых проверяется качество продукта.

Рекомендации по выполнению:

– делайте Definition of Done проверяемым (условие «да/нет»), а не декларативным (формулировка вроде «код должен быть хорошим» не поддается однозначной проверке);

– привязывайте метрики качества к специфике своей предметной области – например, для платежной системы уместны метрики по безопасности и точности расчетов, а для CRM – метрики по производительности выборок;

– указывайте не только сами метрики, но и конкретные целевые значения (например, «покрытие тестами не менее 70 %»), иначе метрика остается декларативной;

– формируя чек-лист code review, включайте не только стилистические, но и содержательные пункты – обработку ошибок, покрытие тестами, отсутствие дублирования кода.

Управление коммуникациями.

План управления коммуникациями определяет, какая информация о проекте, в каком формате, с какой периодичностью и через какие каналы должна передаваться заинтересованным сторонам. В основе процесса коммуникации лежит простая модель: отправитель кодирует сообщение и передает его получателю по определенному каналу, а получатель декодирует сообщение и, как правило, предоставляет обратную связь; на каждом из этих этапов возможны искажения («шум») – например, из-за различий в терминологии между технической командой и бизнес-заказчиком.

Выделяют три основных метода коммуникации: push-коммуникация (информация рассылается получателям без запроса с их стороны – письма, отчеты, уведомления), pull-коммуникация (получатель самостоятельно обращается за информацией – база знаний, дашборд проекта, документация в общем доступе) и интерактивная коммуникация (двусторонний обмен информацией в реальном времени – совещания, звонки, переписка в чате). Основой плана коммуникаций служит матрица коммуникаций со stakeholders (Stakeholder Communication Matrix), которая для каждой заинтересованной стороны или группы фиксирует потребность в информации, ответственного за коммуникацию, канал связи и частоту отчетности.

Рекомендации по выполнению:

– соотносите канал и частоту коммуникации с типом stakeholder из раздела документа 2 – топ-менеджменту, как правило, не нужны еженедельные технические статус-отчеты, а команде разработки не всегда нужны формальные письма там, где достаточно короткого сообщения в чате;

– используйте push-коммуникацию для регулярных статус-отчетов, pull-коммуникацию – для документации и базы знаний, а интерактивную – для обсуждения решений и разбора проблем;

– не перегружайте план коммуникаций избыточным количеством каналов – трех-четырех основных (например, статус-отчет, совещание, чат, дашборд) обычно достаточно для учебного проекта.

Показатели эффективности и мониторинг проекта.

Мониторинг проекта строится на системе ключевых показателей эффективности (KPI), которые принято группировать по категориям: Schedule (сроки – например, отклонение от планового графика, доля выполненных в срок задач), Quality (качество – количество дефектов, покрытие тестами), Team (команда – загрузка, текучесть, соблюдение сроков спринтов) и Stakeholder Satisfaction (удовлетворенность заинтересованных сторон – по результатам опросов или обратной связи). Важно различать опережающие индикаторы (leading indicators), которые сигнализируют о проблеме до того, как она повлияла на результат (например, рост числа открытых дефектов), и запаздывающие индикаторы (lagging indicators), фиксирующие уже свершившийся факт (например, просроченная веха).

Для более комплексной оценки состояния проекта по срокам и стоимости в проектном управлении применяется метод освоенного объема (Earned Value Management, EVM), оперирующий тремя базовыми величинами: плановый объем (Planned Value, PV) – стоимость работ, которые должны были быть выполнены к текущей дате по плану; освоенный объем (Earned Value, EV) – стоимость фактически выполненных работ; фактические затраты (Actual Cost, AC) – реально понесенные затраты. На их основе рассчитываются индекс выполнения сроков (Schedule Performance Index, SPI = EV / PV) и индекс выполнения стоимости (Cost Performance Index, CPI = EV / AC): значение индекса меньше единицы говорит об отставании от плана или перерасходе бюджета соответственно. Использование EVM выходит за рамки обязательного минимума данного практикума, но может быть предложено как дополнительный, более продвинутый способ представления показателей эффективности проекта.

Для наглядного представления текущего состояния проекта используется информационная панель проекта (Project Dashboard), включающая общий статус проекта по RAG-шкале (Red/Amber/Green – критическое состояние/есть отклонения/все в норме), ключевые метрики, список основных рисков и проблем (Top Risks/Issues), а также статус ключевых вех проекта (Milestones Status).

Регулярный мониторинг этих показателей позволяет руководителю проекта своевременно принимать управленческие решения и корректировать план.

Рекомендации по выполнению:

– задавайте не только сами KPI, но и целевые значения или пороги RAG-статуса (например, «зеленый статус – отклонение по срокам не более 5 %») – без порогов дашборд остается бессмысленным;

– проверяйте, что KPI покрывают все четыре категории (Schedule, Quality, Team, Stakeholder Satisfaction), а не только сроки;

– делайте макет Project Dashboard компактным и читаемым «с одного взгляда»: 4–6 ключевых метрик и статусов обычно достаточно для учебного проекта;

– если хотите усилить работу, добавьте расчет SPI и CPI для одного контрольного периода проекта – это хорошо демонстрирует понимание количественных методов мониторинга.

1 Лабораторная работа № 1. Описание компонентов IT-проекта как организационно-технической системы

Цель работы: формализовать IT-проект через создание устава проекта, идентификацию заинтересованных сторон и обоснованный выбор методологии управления.

Порядок выполнения работы

1 Разработать устав проекта (Project Charter).
2 Идентифицировать и проанализировать заинтересованные стороны.
3 Построить матрицу Power/Interest Grid.
4 Выбрать и обосновать методологию управления проектом.

Результаты

Заполненный раздел документа «1. Устав проекта»:
– Бизнес-обоснование;
– Цели проекта (SMART);
– Основные результаты;
– Критерии успеха;
– Ограничения и допущения;
– Риски высокого уровня.
Заполненный раздел документа «2. Реестр заинтересованных сторон»:
– Таблица stakeholders (минимум 8);
– Анализ влияния и интереса;
– Power/Interest Grid;
– Стратегии вовлечения.

Заполненный раздел документа «3. Обоснование выбора методологии»:
- Анализ характеристик проекта;
- Выбор методологии (Waterfall/Agile/Hybrid);
- Обоснование выбора.

Контрольные вопросы

1 Что такое устав проекта (Project Charter) и какие сведения он должен содержать?

2 Чем цели проекта, сформулированные по критериям SMART, отличаются от произвольно сформулированных целей?

3 Кто такие заинтересованные стороны проекта (stakeholders) и зачем составляется их реестр?

4 Как строится и для чего применяется матрица Power/Interest Grid?

5 По каким критериям осуществляется выбор между Waterfall, Agile и гибридной методологией управления проектом?

6 Какие риски высокого уровня целесообразно указывать уже на этапе устава проекта?

2 Лабораторная работа № 2. Описание компонентов системы управления IT-проектом

Цель работы: декомпозировать IT-проект на управляемые компоненты и создать иерархическую структуру работ (WBS).

Порядок выполнения работы

1 Определить основные фазы проекта.
2 Декомпозировать фазы на работы (или компоненты).
3 Детализировать работы (компоненты) до уровня задач (минимум 3 уровня).
4 Создать WBS Dictionary (словарь WBS) с описанием каждого элемента.
5 Визуализировать WBS в виде таблицы или дерева.

Результаты

Заполненный раздел документа «4. Структурная декомпозиция работ (WBS)»:
- WBS минимум 3-х уровней детализации;
- WBS Dictionary с описанием каждого элемента:
  - WBS Code;
  - Название элемента;
  - Описание;
  - Результат;
  - Критерии приемки.
- Визуализация WBS (иерархическая таблица или диаграмма).

Контрольные вопросы

1 Что такое WBS и какую задачу она решает при планировании проекта?

2 В чем заключается «правило 100 %» при построении WBS?

3 Чем пакет работ (work package) отличается от фазы и работы верхнего уровня декомпозиции?

4 Какие сведения включаются в словарь WBS (WBS Dictionary) для каждого элемента?

5 Какими способами можно визуализировать WBS?

6 Как WBS используется на последующих этапах планирования проекта (оценка трудоемкости, календарное планирование)?

3 Лабораторная работа № 3. Оценка трудоемкости IT-проекта

Цель работы: оценить трудоемкость задач IT-проекта с использованием различных методов оценки и рассчитать общую длительность проекта.

Порядок выполнения работы

1 Применить метод трехточечной оценки (PERT) к ключевым задачам из WBS.

2 Провести оценку методом Planning Poker (симуляция или в команде).

3 Рассчитать ожидаемую трудоемкость (Expected) и стандартное отклонение (σ) для каждой задачи.

4 Рассчитать общую трудоемкость проекта.

5 Определить длительность проекта с учетом доступности ресурсов.

Результаты

Заполненный раздел документа «5. Оценка трудоемкости»:

- Таблица оценок трудоемкости всех задач из WBS:
  - Task ID, Task Name;
  - Optimistic (O), Most Likely (M), Pessimistic (P);
  - Expected (E) = (O + 4M + P) / 6;
  - Standard Deviation (σ) = (P - O) / 6;
- Story points для функциональных элементов (если Agile);
- Итоговая оценка проекта:
  - Общая трудоемкость в человеко-часах;
  - Общая трудоемкость в человеко-днях;
  - Прогнозная длительность проекта.

Контрольные вопросы

1 Что такое трудоемкость и чем она отличается от длительности работы?

2 В чем заключается метод трехточечной оценки (PERT) и по каким формулам рассчитываются ожидаемая трудоемкость и стандартное отклонение?

3 Как проводится оценка методом Planning Poker и какие преимущества у этого метода?

4 Какие еще методы оценки трудоемкости, помимо PERT и Planning Poker, применяются на практике?

5 Как рассчитывается общая трудоемкость проекта и его прогнозная длительность с учетом доступности ресурсов?

6 Зачем помимо трудоемкости в человеко-часах рассчитывать ее в человеко-днях?

4 Лабораторная работа № 4. Формирование состава и структуры команды исполнителей IT-проекта

Цель работы: определить требования к команде IT-проекта, создать организационную структуру, матрицу навыков и распределить ответственность с использованием RACI-матрицы.

Порядок выполнения работы

1 Определить необходимые роли для проекта.
2 Определить размер команды и FTE (Full-Time Equivalent).
3 Создать организационную структуру команды.
4 Создать матрицу требуемых навыков (Skills Matrix).
5 Провести gap-анализ навыков.
6 Разработать RACI-матрицу для ключевых работ (минимум 15 работ).

Результаты

Заполненный раздел документа «6. Команда проекта».
6.1 Структура команды:
- Список ролей с обязанностями;
- Организационная структура (Org chart);
- Расчет FTE по ролям;
- Общая потребность в человеко-месяцах.

6.2 Матрица навыков (Skills Matrix):
- Таблица навыков по членам команды;
- Легенда уровней (0-4);
- Gap-анализ (требуемые vs имеющиеся навыки);
- План устранения дефицита навыков (обучение/найм).

6.3 Матрица ответственности (RACI):

- RACI-матрица для минимум 15 ключевых работ.
- Для каждой работы определены:
- R (Responsible) – исполнитель;
- A (Accountable) – ответственный (только один!);
- C (Consulted) – консультируемый;
- I (Informed) – информируемый.

Контрольные вопросы

1 Как определяется необходимый состав ролей команды IT-проекта?

2 Что такое FTE (Full-Time Equivalent) и как он рассчитывается?

3 Для чего строится матрица навыков (Skills Matrix) и что такое gap-анализ навыков?

4 Что означают буквы R, A, C и I в RACI-матрице и почему для одной работы допускается только один ответственный (Accountable)?

5 Какие меры устранения дефицита навыков команды можно предусмотреть по результатам gap-анализа?

6 Как организационная структура команды проекта влияет на распределение ответственности?

5 Лабораторная работа № 5. Декомпозиция задач IT-проекта на подзадачи

Цель работы: создать детальный календарный план IT-проекта с установлением зависимостей между задачами и определением критического пути с использованием MS Project.

Порядок выполнения работы

1 Создать новый проект в MS Project.

2 Настроить календарь проекта (рабочие дни, праздники).

3 Импортировать WBS и задачи из Google Docs.

4 Установить длительность задач (из раздела документа 5 «Оценка трудоемкости»).

5 Установить зависимости между задачами (FS, SS, FF, SF).

6 Определить вехи (Milestones) проекта.

7 Определить критический путь проекта.

8 Сохранить baseline.

9 Экспортировать диаграмму Ганта в Google Docs.

Результаты

1 Файл MS Project (.mpp) с календарным планом проекта.

2 Заполненный раздел документа «7. Календарный план (Диаграмма Ганта)»:

- Диаграмма Ганта (скриншот или экспорт из MS Project);
- Таблица задач с датами начала и окончания;
- Список зависимостей между задачами;
- Анализ критического пути:
  - Задачи на критическом пути;
  - Общая длительность проекта;
  - Резервы времени (float) для некритических задач;
  - Ключевые вехи проекта с датами.

Контрольные вопросы

1 Какие типы зависимостей между задачами (FS, SS, FF, SF) применяются при построении календарного плана?

2 Что такое критический путь проекта и как он определяется?

3 Чем веха (milestone) отличается от обычной задачи календарного плана?

4 Что такое резерв времени (float) и в чем разница между свободным и полным резервом?

5 Для чего в MS Project сохраняется базовый план (baseline) проекта?

6 Как оценка трудоемкости, полученная в предыдущей лабораторной работе, используется при установлении длительности задач календарного плана?

6 Лабораторная работа № 6. Распределение исполнителей IT-проекта на задачи

Цель работы: распределить задачи между членами команды, назначить ресурсы на задачи в MS Project, спланировать загрузку ресурсов и выполнить балансировку.

Порядок выполнения работы

1 В MS Project создать ресурсы на основе раздела 6.1 «Структура команды».

2 Назначить ресурсы на задачи из календарного плана.

3 Настроить стоимость ресурсов (если применимо).

4 Рассчитать capacity (емкость) команды по периодам.

5 Построить ресурсные гистограммы (Resource Histograms).

6 Выявить перегрузку ресурсов (Resource Overallocation).

7 Выполнить выравнивание ресурсов (Resource Leveling).

8 Экспортировать данные о распределении ресурсов в Google Docs.

Результаты

1 Обновленный файл MS Project (.mpp) с назначенными ресурсами.

2 Заполненный раздел документа «8. План управления ресурсами»:

8.1 Распределение задач:

- Таблица распределения задач:
  - ID задачи, Название задачи;
  - Длительность;
  - Ресурс(ы);
  - Назначение %;
  - Трудоемкость (часы);
  - Дата начала, Дата окончания;
  - Матрица назначений (задачи × ресурсы).

8.2 Планирование загрузки:

- Расчет доступной емкости команды:
  - Рабочие дни/месяц;
  - Накладные расходы, %;
  - Доступные часы;
  - Запланированные часы;
- Ресурсные гистограммы по ключевым ролям (скриншоты из MS Project);
- Анализ загрузки по периодам (недели/месяцы):
  - Общая емкость команды (ч);
  - Запланированная работа (ч);
  - Резерв (ч);
  - Загрузка %;
  - Статус (OK / Overallocated / Underutilized);
- План балансировки:
  - Выявленные перегрузки;
  - Меры по устранению (перераспределение, изменение сроков, добавление ресурсов).

Контрольные вопросы

1 Чем распределение ресурсов по задачам отличается от простого назначения исполнителей?

2 Как рассчитывается доступная емкость (capacity) команды по периодам?

3 Что показывает ресурсная гистограмма (Resource Histogram) и как по ней выявить перегрузку ресурса?

4 В чем заключается выравнивание ресурсов (Resource Leveling) и как оно влияет на сроки проекта?

5 Какими способами можно устранить перегрузку ресурса, помимо выравнивания средствами MS Project?

6 Как накладные расходы (%) учитываются при расчете доступных часов ресурса?

7 Лабораторная работа № 7. Оценка эффективности системы управления IT-проектом

Цель работы: разработать комплексную систему оценки и управления эффективностью IT-проекта, включающую управление рисками, качеством, коммуникациями и KPI.

Порядок выполнения работы

1 Часть 1. Управление рисками:
- Идентифицировать минимум 10 рисков проекта;
- Оценить вероятность (1-5) и влияние (1-5) каждого риска;
- Построить матрицу рисков (вероятность × влияние);
- Разработать стратегии реагирования на топ-5 рисков.

2 Часть 2. Управление качеством:
- Определить стандарты качества для проекта;
- Создать Definition of Done по уровням;
- Определить метрики качества с целевыми значениями;
- Создать чек-лист code review.

Часть 3. Управление коммуникациями:
- Создать план коммуникаций со stakeholders;
- Определить каналы коммуникации и частоту отчетности.

Часть 4. Показатели эффективности и мониторинг:
- Определить KPI проекта по категориям:
  - Schedule (сроки);
  - Quality (качество);
  - Team (команда);
  - Stakeholder Satisfaction (удовлетворенность);
- Спроектировать структуру Project Dashboard с RAG-статусами.

Результаты

1 Заполненный раздел документа «9. Реестр рисков»:
- Таблица рисков (минимум 10);
- Матрица вероятность × влияние;
- План реагирования на топ-5 рисков.

2 Заполненный раздел документа «10. План управления качеством»:
- Quality Standards;
- Definition of Done (4 уровня: задача, user story, спринт, релиз);
- Метрики качества (код, дефекты, тестирование) с целевыми значениями;
- Чек-лист code review.

3 Заполненный раздел документа «11. План коммуникаций»:

- Stakeholder Communication Matrix;
- Каналы коммуникации;
- График отчетности.

4 Заполненный раздел документа «12. Показатели эффективности и мониторинг»:

- Таблица KPI по категориям (Schedule, Quality, Team, Stakeholder Satisfaction);
- Макет Project Dashboard:
  - Overall Status (RAG);
  - Ключевые метрики;
  - Top Risks/Issues;
  - Milestones Status.

Контрольные вопросы

1 Из каких этапов состоит процесс управления рисками проекта и как строится матрица рисков «вероятность × влияние»?

2 Какие стратегии реагирования на риски применяются для угроз и для благоприятных возможностей?

3 Что такое Definition of Done и для каких уровней (задача, пользовательская история, спринт, релиз) он определяется?

4 Какие метрики качества обычно используются для оценки кода, дефектов и тестирования?

5 Как формируется план коммуникаций проекта и что должна включать матрица коммуникаций со stakeholders?

6 Какие категории KPI используются для оценки эффективности IT-проекта и что показывает RAG-статус на Project Dashboard?

Список литературы

1 Управление программными проектами : учебник для вузов / под ред. Р. Ф. Маликова. – М. : Юрайт, 2026. – 167 с.

2 **Чекмарев, А. В.** Управление цифровыми проектами и процессами : учебник для вузов / А. В. Чекмарев. – 2-е изд., перераб. и доп. – М. : Юрайт, 2026. – 424 с.

3 Управление проектами : учебник и практикум для вузов / А. И. Балашов, Е. М. Рогова, М. В. Тихонова, Е. А. Ткаченко ; под ред. Е. М. Роговой. – М. : Юрайт, 2026. – 302 с.

4 **Зуб, А. Т.** Управление проектами : учебник и практикум для вузов / А. Т. Зуб. – 2-е изд., перераб. и доп. – М. : Юрайт, 2026. – 397 с.

5 **Горбашко, Е. А.** Управление проектами : учебник для вузов / Е. А. Горбашко ; под ред. Е. А. Горбашко. – М. : Юрайт, 2026. – 358 с.

Приложение А
(справочное)

Варианты заданий для выполнения лабораторных работ

1 Автострахование.
2 Агентство по сдаче автомобилей в аренду.
3 Аренда коньков, роликов, велосипедов, лыж.
4 Аэропорт – пассажирское расписание и перевозки.
5 Банковская система вкладов (физических и юридических лиц) .
6 Банковская система кредитования (физических и юридических лиц).
7 Биллинг сотовой компании.
8 Ветеринарная лечебница.
9 Клуб обучения танцам.
10 Магазин косметики.
11 Машиностроительное предприятие: система по разработке и модификации изделий (ведение архива, стандартов и пр.).
12 Нефтеперерабатывающая компания.
13 Парикмахерская.
14 Поставка вин.
15 Приемная комиссия вуза.
16 Производство мебели (прием индивидуальных и типовых заказов и изготовление).
17 Рекламное агентство.
18 Риелторская компания: аренда, продажа первичного и вторичного жилья.
19 Санаторий.
20 Система управления проектом для IT-компании.
21 Складская логистика.
22 Спа-салон (услуги, обслуживающий персонал и пр.).
23 Страховая компания.
24 Такси.
25 Транспортная логистика.
26 Туристическое агентство (путешествия за рубеж).
27 Туристическое агентство (путешествия по Беларуси).
28 Учет оборудования на крупном промышленном предприятии.
29 Филармония.
30 Электронный проездной.