

Рекомендации по практикам и ВКР кафедры системного программирования СПбГУ

Я.А. Кириленко

Ю.В. Литвинов

С.Ю. Сартасов

Наиболее важное

- Выбираете научного руководителя и тему из списка предложенных. Список публикуется в соответствующей команде в Microsoft Teams или иным способом доводится руководителем практики.
- Выполняете работу, раз в неделю отчитываясь научному руководителю (и консультанту, если есть) о том, на что ушло время.
- В конце семестра сдаёте на кафедру отчёт, отзыв научного руководителя, и проходите защиту.
 - Шаблон отчёта: <https://github.com/spbu-se/matmex-diploma-template>
 - Шаблон презентации: https://github.com/spbu-se/report_presentation_template
 - Примеры работ: <https://se.math.spbu.ru/theses.html?supervisor=0&startdate=2021&enddate=2021>
 - Шаблоны отзывов:
 - * Научного руководителя: https://docs.google.com/document/d/1bjI10sBn47HhU_q9DRnrz-yjiWtiML85uDU_n50gQXc
 - * Консультанта: <https://docs.google.com/document/d/13jq0sNdH6lWY9qg0mdDIaxcU7eEK3iGJuveiaqzJxyk>
 - Шаблон акта о внедрении: <https://drive.google.com/file/d/1yd7DH2y94CDiBxHX59-I7BTwrtrwit6r>

1. Введение

Учебные (производственные, преддипломные и т.д.) практики — это ключевая форма обучения на кафедре системного программирования, позволяющая получить практический опыт и применить полученные знания в условиях, близких к реальным — как промышленных проектов, так и научно-исследовательской работы. Каждая практика уникальна¹ и состоит в решении достаточно сложной технической или научно-технической задачи. По сути, практика представляет собой мини-исследование (а иногда и не мини) и схожа по методологии и структуре с исследовательским проектом, хоть и может быть чисто технической по сути. Хорошие практики часто заканчиваются публикациями или внедрением в кодовую базу существующего проекта.

К практикам надо относиться как к промежуточной отчётности по некоторому проекту, которым вы занимаетесь в течение какого-то времени. Защищать практику надо в конце каждого семестра, поэтому минимальная продолжительность работы составляет один семестр, однако ничто не мешает (а на старших курсах даже ожидается) продолжать работу над одной темой и дальше. Например, можно в 3-м семестре позаниматься одной темой, в 4-м выбрать другую и работать над ней весь остаток второго и третий курс, на ВКР выбрать третью. Единственное, что на защите ВКР ожидается минимум годовая работа, поэтому менять тему зимой четвёртого курса может быть плохой идеей. В силу ориентированности на практические навыки кафедра СП также требует, чтобы хотя бы одна работа за год включала в себя содержательное программирование — например, осенью написать инструмент, весной поменять тему и сделать обзорную работу без строчки кода, это допустимо.

Работа ведётся под руководством научного руководителя, и, если тема практики предоставлена какой-либо компанией, консультанта, при тесном взаимодействии с ними. Заканчивается практика сдачей отчёта, отзывов научного руководителя и консультанта, и защитой результатов перед комиссией, состоящей из преподавателей и представителей работодателей. Допустимы и даже поощряются групповые проекты, но отчёт каждый участник пишет сам, и описывает как проект в целом, так и свой личный вклад. На защите каждый участник проекта выступает сам, и тоже должен явно обозначить, что конкретно он сделал.

Это позволяет обучающимся в практически полной мере почувствовать профессию, понять для себя, хотят ли они этим заниматься всю жизнь, выбрать специализацию, в которой будут дальше развиваться. Для преподавателей это возможность объективно оценить способности каждого обучающегося — списать практику невозможно, делать за кого-то слишком трудоёмко, поэтому обучающийся вынужден продемонстрировать именно свои навыки в профессиональном контексте. Это иногда становится нелёгким испытанием для обучающихся, потому что умение зубрить написанию практики никак не помогает, и

¹то есть у нас нет работ по методичкам

если вдруг курс по программированию был изучен недостаточно усердно, приходится навёрстывать упущенное. По статистике примерно треть обучающихся отправляется с практиками на пересдачу, поэтому стоит отнестись к практикам очень серьёзно.

Дальнейшее изложение будет устроено следующим образом: описание видов практик, типичного процесса работы, структуры отчёта, критериев оценивания, общих требований и рекомендаций — всё это применимо для всех курсов. Далее будет описана специфика третьего курса, работы над ВКР бакалавра, особенности практик в магистратуре. Таким образом, второкурсники могут прочитать только общую часть, чтобы на следующих курсах ознакомиться с остальными и не читать весь этот текст заново.

2. Виды работ

Практики защищаются в:

- для образовательных программ «Математическое обеспечение и администрирование информационных систем» и «Технологии программирования» — в комиссии той кафедры, с которой научный руководитель;
- для образовательной программы «Программная инженерия» — в комиссии кафедры системного программирования, со всеми вытекающими последствиями (поскольку по этой программе выпускает только кафедра СП и на защите диплома выбора всё равно не будет).

Темы, выдаваемые научными руководителями, также не произвольны, а определяются интересами кафедры — это связано с тем, что оценивать практики будет комиссия, и комиссия должна понимать, как оценивать работы. Кафедра системного программирования — это «программирование для программистов», то есть ей предлагаются темы, в основном связанные с компиляторами, отладчиками, профиляторами, другими инструментами разработки, операционными системами, библиотеками (возможно, прикладными, но обязательно для сторонних программистов), низкоуровневым программированием, методологиями и технологиями разработки. Например, машинное обучение, если это не машинное обучение для помощи программисту — не на СП (и даже если научный руководитель с ИАС, но студент с программной инженерии, защитить такую тему будет очень сложно). Вместе с тем, прикладные продуктовые задачи на кафедре системного программирования вполне можно защищать, даже если они с инструментами программирования не связаны. Но и оцениваться такие задачи будут как продуктовые.

Концептуально практики можно разделить на пять больших групп:

- Решение — обучающемуся поручили найти способ решения проблемы в области разработки программного обеспечения или теоретической информатики с учётом набора ограничений.
- Эксперимент — обучающемуся поручили изучить возможности, достоинства и недостатки новой технологии, платформы, языка и т. д. на примере какой-то задачи.
- Производственное задание — автору поручили реализовать потенциально полезное программное обеспечение.
- Сравнение — обучающемуся поручили сравнить несколько существующих продуктов и/или подходов.
- Теоретическое исследование — автору поручили доказать какое-то утверждение, исследовать свойства алгоритма и т.п., при этом не требуя написания кода.

Требования к практикам в целом одинаковые, но оцениваться они будут по-разному, то есть применяются одинаковые критерии, но с разными весами. Например, для производственного задания не важна научная новизна (но всё равно, практика не должна сводиться к решению задачи, которую давно и хорошо умеют решать), а для теоретического исследования не имеет значения качество кода (который вообще может отсутствовать), и поэтому не оценивается вовсе. Кафедра системного программирования ожидает, что любая работа имеет практическую реализацию, так что теоретические исследования лучше защищать на других кафедрах (либо придумать содержательную программистскую часть и перевести задачу в категорию «Решение»).

Для второго курса требования к теме сводятся к реализации приложения (или функциональности в уже существующем приложении) с использованием технологий, которых не было в обязательном курсе (типы работ в основном «Решение» или «Производственное задание»). Например, несложное мобильное приложение или игра вполне может быть защищено как односеместровая практика, если оно не было сделано по tutorialу и не совсем уж простое (надо уметь пояснить, на что потрачено примерно 60 часов, которые ожидаются на практику за семестр). Фактически от второго курса ожидается в основном умение писать хороший код и искать информацию. Дальше требования немного усложняются — на выходе из бакалавриата кафедра ожидает видеть грамотного и вдумчивого младшего разработчика, на выходе из магистратуры — разработчика с навыками исследователя и готовностью развиваться как в сторону высоких технических позиций, так и строить научную карьеру.

3. Процесс

Роли, участвующие в работе над практикой:

- *консультант* — ставит задачу, читает и рецензирует код, помогает с техническими проблемами, рассказывает про существующий проект (если практика делается в уже существующем проекте), кто в итоге принимает результат и говорит, доволен ли он тем, что получилось;
- *научный руководитель* — преподаватель отделения информатики матмеха (обязательно), который отвечает за адекватность постановки задачи и соответствие её профилю кафедры, следит за методологическими вопросами (например, что действительно выполнен обзор существующих решений, эксперименты соответствуют постановке задачи и т.д.), следит за ходом работы, помогает с текстом и подготовкой к защите; обратите внимание, научный руководитель и консультант могут быть одним лицом;
- *руководитель практики* — его часто путают с научным руководителем, но это совсем другая роль (ранее известная как *куратор*), его задачи: общая организация процесса, сбор и распределение тем, сбор отчётов и отзывов, организация защиты, решение организационных проблем; обычно он один на каждую кафедру и направление;
- *комиссия* — все защиты практик выполняются перед комиссией, даже если это первая попытка. Комиссия состоит из преподавателей кафедры, на которой защищается работа, и хотя бы одного (обычно больше) представителя индустрии, и назначается отдельным приказом на целый год.

Примерный календарный график работы над практикой таков.

- Конец августа, сентябрь для осенних практик; февраль, начало марта для весенних — выбор темы. Руководитель практики обычно посредством команды курса в MS Teams, почтовой рассылки или сайта кафедры СП (<https://se.math.spbu.ru/diplomas/index.html>) выкладывает предложенные преподавателями и компаниями темы, с указанием научных руководителей и консультантов, которые будут эти темы курировать. Также допустимо предлагать что-то своё, в этом случае всё равно потребуется научный руководитель, и нет гарантии, что вашу тему одобрят. Тему нужно обязательно согласовать с научным руководителем и получить подтверждение, что он готов взяться за руководство. Выбор темы — ответственность обучающегося, и отсутствие темы или научного руководителя не спасёт от получения незачёта. Выбранную тему и научника надо записать в таблицу, которую пришлёт руководитель практики — некоторые направления реализуются несколькими кафедрами, так что если обучающийся не записался, все будут думать, что он пишет работу где-то ещё.
 - Хорошие работы начинаются до начала семестра — например, в летней школе. Тогда, как правило, к концу семестра получается без проблем защищаемый результат.
 - Более того, в летней школе или на летней стажировке можно сделать большую часть работы, оставив на семестр только её оформление и защиту. Сдача полученного ранее результата вполне приветствуется.
- Сентябрь-начало декабря, и, соответственно, март-середина мая — работа над практикой. В это время обязательна как минимум еженедельная отчётность научному руководителю о ходе работы², того же обычно ожидает и консультант. Типичный план работы над практикой таков:
 - погружение в предметную область, которое заканчивается написанием примерно страницы текста с введением и постановкой задачи — это делается «по горячим следам» после первого общения с научником в течение нескольких дней, результирующий текст обязательно согласуется с научником;
 - обзор аналогов и используемых инструментов — требует времени (от пары недель до месяца), но очень важен, чтобы не изобрести велосипед; до начала обзора должна быть зафиксирована методика и критерии, результат — обычно от двух до 15 страниц текста, также согласуется с научником;
 - проектирование и реализация — тут в основном пишется код, но также и текст, где описываются основные архитектурные решения и то, над чем пришлось подумать в плане реализации;
 - апробация — выполнение экспериментов, показывающих (желательно, количественно), что результат соответствует постановке задачи;
 - написание текста, подготовка к защите — текст по большей части к этому моменту должен быть уже готов, надо лишь собрать всё воедино и оформить. Написание текста с нуля займёт неделю-две, сборка и оформление также займёт несколько дней и несколько попыток сдать текст научнику.

²для чего есть удобный сервис сайта кафедры: <https://se.math.spbu.ru/practice>.

- Конец ноября, начало мая — предзащиты. Это выступление-подготовка к защите, где не требуется текст и окончательные результаты, но слайды и рассказ должны быть приближены к финальным. На предзащитах можно получить обратную связь и поправить её к защите, существенно увеличив шансы на зачёт, однако оценки на предзащите принципиально не ставятся³. Предзащиты проводятся не во всех семестрах, но в конце осеннего семестра второго курса (первый опыт практики) и в конце весеннего семестра четвёртого курса бакалавриата (перед защитой ВКР) они всегда есть.
- Вторая неделя декабря и середина мая — начало выборочного рецензирования текстов работ. Тексты должны быть уже готовы и загружены в место, указанное руководителем практики (как правило, это команда Microsoft Teams либо сайт кафедры). Рецензенты из числа преподавателей, магистрантов и представителей индустрии читают работы (не все, а которые их по тем или иным причинам заинтересовали, почему рецензирование и выборочное) и пишут рецензию, оценивая текст по ряду критериев. В случае наличия существенных замечаний (а это в 80 процентах случаев так) потребуется поправить работу и сдать её на повторное рецензирование снова — загрузить снова и сообщить, что работа готова к повторной проверке. Рецензирование одинарное слепое, то есть рецензент знает, чья это работа, но обучающийся не знает, кто его рецензирует, и не может напрямую общаться с рецензентом. Если рецензия не получена за два дня до защиты, текст считается зачтённым без замечаний (хотя и может быть просмотрен комиссией во время защиты, и повлиять на оценку).
- Вторая половина декабря, вторая половина мая — защиты. Для допуска к защите требуется отзыв научного руководителя и консультанта (если консультант имеется) и текст. Отзыв пишет научный руководитель и обязательно заверяет его подписью. Допустимо отзыв научного руководителя и консультанта сдавать в виде одного документа с двумя подписями. На защите обучающийся выступает с докладом (со слайдами) не более чем на восемь минут, отвечает на вопросы, в это время комиссия оценивает доклад и представленные материалы по критериям, описанным ниже. Когда все защищающиеся выступают, комиссия совещается и выставляет оценки, которые публикуются в команде в Teams, после чего есть возможность получить от комиссии обратную связь, если есть потребность (обратите внимание, апелляции при защите учебных практик не проводятся, поэтому объявленные оценки окончательны).
 - Защиты проводятся даже если вы планируете продолжить работать над темой — на защиту выносятся текущие результаты, и они должны быть достаточны, чтобы было понятно, за что ставить зачёт сейчас, не принимая во внимание перспективы проекта. В случае многосеместровых проектов разбивайте работу так, чтобы к концу каждого семестра иметь законченный результат, будь то обзор, эксперименты или программная реализация.
 - Защиты обычно проводятся несколько дней, один из которых — день зачёта по расписанию. Защиты до дня официального зачёта не являются дополнительными попытками.
 - В случае получения оценки F или «незачёт» пересдача проходит по расписанию (либо также в несколько дней около даты пересдачи по расписанию, если желающих много).

4. Типичная структура отчёта по практикам

Отчёт по практике пишется по типовой схеме, состоящей из следующих разделов.

- Титульный лист, оформленный по шаблону из рабочей программы практики, которую можно найти на сайте СПбГУ (<https://spbu.ru/sveden/education>)⁴.
- Оглавление.
- Введение, с описанием предметной области, объяснением актуальности задачи и направления её решения.
- Формулировка цели работы. Одно предложение про то, зачем делается работа, далее нумерованный список задач, которые нужны для достижения цели (список из трёх-пяти пунктов, например, сделать обзор существующих решений, разработать архитектуру, реализовать, провести эксперименты).
- Обзор:
 - существующих решений;
 - используемых технологий;
 - существующих наработок исследовательской группы, научного руководителя, предыдущих обучающихся;
 - в общем всё, что сделано не вами, но необходимо для понимания работы.

³Да, возможно, что на предзащите скажут, что всё хорошо, а на защите поставят «неуд», такое бывает.

⁴Или воспользоваться готовым TeX-овским шаблоном, <https://github.com/spbu-se/matmex-diploma-template>.

- Описание реализации (если есть — для обзорных статей тут можно описать код, который пришлось написать для экспериментов, наборы данных и т.п.):
 - архитектура решения;
 - интересные детали реализации, нетривиальные алгоритмы;
- Описание экспериментов и/или апробации (если они готовы).
- Заключение с кратким перечислением текущих результатов (должно соответствовать постановке задачи) и ссылкой на реализацию (если есть).
 - Сюда же пишутся благодарности (не научному руководителю и консультанту, они и так упоминаются на титульном листе — но если с работой ещё кто-то помогал) и ссылки на гранты/стипендии, которыми поддержана работа (если есть).
 - Планы на будущее писать не стоит — только если они достаточно обширны для следующей практики, и вы готовы выступить консультантом.
- Список литературы.

5. Критерии оценивания

5.1. Выступление/работа (группа критериев В)

- *В1. Ясность изложения темы и задачи, их актуальности*
 - Отлично (5): обучающийся ясно и понятно доносит тему работы, ее актуальность, поставленные задачи и результаты.
 - Очень хорошо (4): требуется уточняющий вопрос для снятия сомнений по актуальности темы.
 - Хорошо (3): обучающемуся требуется задать несколько вопросов для полного понимания его темы, актуальности и поставленных задач.
 - Удовлетворительно (2): лучше, чем достаточно, но хуже, чем хорошо.
 - Достаточно (1): выступающему требуется задать серию вопросов, прежде чем станет понятна цель и задачи в его работе.
 - Плохо (0): выступающий не может внятно и понятно рассказать про актуальность выбранной темы, а также про цели и поставленные задачи.
- *В2. Степень полноты изложения*
 - Отлично (5): доклад содержит убедительные доказательства представленных утверждений, выводы аргументированы, изложение свободно от неточностей.
 - Очень хорошо (4): доклад в целом убедительно доказывает позицию автора, но требуются уточняющие вопросы, или автор допускает оценочные суждения, не влияющие при этом на полноту изложения основных результатов.
 - Хорошо (3): в изложении имеются упущения и неточности, не ставящие под сомнение справедливость результатов и выводов.
 - Удовлетворительно (2): лучше, чем достаточно, но хуже, чем хорошо.
 - Достаточно (1): в работе есть преодолимые неточности, незначительные ошибки, потребовавшие дополнительного обсуждения, или ошибки, найденные и корректно исправленные в ходе защиты.
 - Плохо (0): представленная работа содержит существенные ошибки или существенные упущения, заставляющие обоснованно сомневаться в корректности результата.
- *В3. Степень научной/инженерной новизны полученного результата*
 - Отлично (5): Работа содержит новые результаты, полученные лично автором.
 - Очень хорошо (4): лучше, чем хорошо, но хуже, чем отлично.
 - Хорошо (3): работа содержит результаты, повторяющие уже известные, но они получены применением новых подходов и методов.
 - Удовлетворительно (2): лучше, чем достаточно, но хуже, чем хорошо.

- Достаточно (1): результаты и методы их достижения, представленные в работе, являются известными, однако выбор и стиль их изложения демонстрирует базовые профессиональные навыки обучающегося.
 - Плохо (0): не продемонстрировано ничего из вышеизложенного.
- *B4. Способность к участию в научной дискуссии*
 - Отлично (5): в процессе защиты продемонстрирована способность к участию в научной дискуссии, даны аргументированные ответы на все вопросы, продемонстрирована культура рассуждения, высокая культура речи и общая эрудиция.
 - Очень хорошо (4): лучше, чем хорошо, но хуже, чем отлично.
 - Хорошо (3): в процессе защиты были даны обоснованные ответы на часть вопросов, продемонстрирована культура рассуждения, высокая культура речи и профессиональная эрудиция.
 - Удовлетворительно (2): лучше, чем достаточно, но хуже, чем хорошо.
 - Достаточно (1): в процессе защиты большинство ответов на вопросы были недостаточно аргументированы, либо имеются существенные проблемы с культурой речи или навыками ведения дискуссии.
 - Плохо (0): в процессе защиты не были даны ответы на большинство вопросов, заданных комиссией, либо комиссией были сделаны замечания о стиле ведения дискуссии, о грубых нарушениях культуры речи или было продемонстрировано неуважение к комиссии.
 - *B5. Качество подготовки презентационных материалов*
 - Отлично (5): презентационные материалы подготовлены с учетом всех рекомендаций, изложенных в РПП, ясно и наглядно поясняют работу, не требуют уточняющих вопросов.
 - Очень хорошо (4): лучше, чем хорошо, но хуже, чем отлично.
 - Хорошо (3): презентационные материалы подготовлены с учётом большинства рекомендаций, или имеется несколько незначительных ошибок. При этом найденные недочёты не мешают восприятию материала.
 - Удовлетворительно (2): лучше, чем достаточно, но хуже, чем хорошо.
 - Достаточно (1): презентационные материалы выполнены в плохом качестве. Докладчику требуется дополнительно объяснять презентационные материалы.
 - Плохо (0): презентационные материалы отсутствуют или выполнены в неудовлетворительном качестве.

5.2. Текст отчёта (группа критериев О)

- *O1. Соответствие содержания и оформления предъявленным требованиям*
 - Отлично (5): по своему содержанию и оформлению работа соответствует всем предъявленным требованиям.
 - Очень хорошо (4): лучше, чем хорошо, но хуже, чем отлично.
 - Хорошо (3): по своему содержанию и оформлению работа соответствует большинству предъявленных требований, или имеется ряд неточностей, которые не мешают общему восприятию работы.
 - Удовлетворительно (2): лучше, чем достаточно, но хуже, чем хорошо.
 - Достаточно (1): По своему содержанию и оформлению работа явно нарушает не более трёх предъявленных требований⁵.
 - Плохо (0): хуже, чем «достаточно».
- *O2. Умение работать с информацией, опубликованной в научных и иных источниках*
 - Отлично (5): в работе описан контекст решаемой задачи, присутствует обзор аналогов, предшествующих работ и используемых инструментов (если это уместно), существенные утверждения работы подтверждены ссылками на источники, составлена библиография по теме работы.
 - Очень хорошо (4): лучше, чем хорошо, но хуже, чем отлично.
 - Хорошо (3): продемонстрированы навыки работы с источниками, но упущены некоторые важные результаты, не все существенные утверждения работы подтверждены ссылками на источники. Составлена библиография по теме работы.
 - Удовлетворительно (2): лучше, чем достаточно, но хуже, чем хорошо.
 - Достаточно (1): в работе не описан контекст решаемой задачи, либо полностью отсутствует обзор аналогов, предшествующих работ либо используемых инструментов (если это уместно), присутствуют субъективные оценочные суждения или многочисленные неподтверждённые утверждения.
 - Плохо (0): отсутствует литературный обзор, библиография по теме работы.

⁵Требования к тексту приводятся в следующем разделе

5.3. Теоретическая часть (группа критериев Т)

- *T1. Обоснование принятых решений/Теоретический анализ*
 - Отлично (5): в работе приведены исчерпывающие аргументы принятых решений.
 - Очень хорошо (4): лучше, чем хорошо, но хуже, чем отлично.
 - Хорошо (3): в работе приведены методологически верные аргументы принятых решений. Дополнительные аргументы могут улучшить работу.
 - Удовлетворительно (2): лучше, чем достаточно, но хуже, чем хорошо.
 - Достаточно (1): в работе есть слабая аргументация принятых решений.
 - Плохо (0): в работе отсутствует аргументация принятых решений.
- *T2. Сравнение с аналогами*
 - Отлично (5): в работе приведен полный сравнительный анализ с аналогами.
 - Очень хорошо (4): лучше, чем хорошо, но хуже, чем отлично.
 - Хорошо (3): в работе приведен подробный сравнительный анализ с аналогами. Однако не все значимые аналоги приведены или сравнение проводилось не по всем значимым критериям.
 - Удовлетворительно (2): лучше, чем достаточно, но хуже, чем хорошо.
 - Достаточно (1): в работе приведены аналоги, но их выбор необоснован, перечень недостаточен, или сравнение с ними ничего не демонстрирует.
 - Плохо (0): в работе отсутствует сравнительная часть с аналогами.

5.4. Практическая часть (группа критериев П)

- *П1. Качество практической части*
 - Отлично (5): качество кода на высоком уровне с соблюдением рекомендаций по архитектуре, стилю и тестированию ПО. Продемонстрировано владение современными технологиями и библиотеками. Также по открытому репозиторию исходного кода видно, что работа велась в течение всего года.
 - Очень хорошо (4): лучше, чем хорошо, но хуже, чем отлично.
 - Хорошо (3): хорошее качество кода, продемонстрировано владение современными технологиями и библиотеками. Или по открытому репозиторию исходного кода видно, что работа велась с существенными перерывами или только в течение одного семестра.
 - Удовлетворительно (2): лучше, чем достаточно, но хуже, чем хорошо.
 - Достаточно (1): практическая часть присутствует и работает исправно, однако архитектура не адекватна решаемой задаче и стиль исполнения низкого качества. Или по открытому репозиторию исходного кода видно, что работа велась только в последний месяц перед датой зачёта.
 - Плохо (0): практическая часть отсутствует или ее наличие не позволяет сделать выводы о техническом уровне автора. Открытый репозиторий исходного кода не предоставлен.
- *П2. Качество проводимых измерений и постановки экспериментов*
 - Отлично (5): экспериментальная методология полностью описана, соответствует принятым в данной области стандартам, и сама по себе не вызывает вопросов, непосредственные результаты задокументированы и доступны, обработка результатов методологически корректна.
 - Очень хорошо (4): лучше, чем хорошо, но хуже, чем отлично.
 - Хорошо (3): экспериментальная методология проработана недостаточно хорошо, однако общее направление экспериментов адекватно поставленной задаче. Обработка данных содержит ошибки, не ставящие, однако, под сомнение общий результат. Указаны только итоговые данные экспериментов.
 - Удовлетворительно (2): лучше, чем достаточно, но хуже, чем хорошо.
 - Достаточно (1): эксперименты выполнены, а результаты обработаны без опоры на качественную экспериментальную методологию, или в ходе работы были допущены существенные ошибки, ставящие результаты под сомнение. Итоговые данные предоставлены не полностью.
 - Плохо (0): экспериментов и/или измерений нет, или они содержат серьёзные ошибки или искажения.

5.5. Правила выставления оценок

По ряду критериев (таких, как О1, П1) комиссия в большой степени руководствуется отзывами научного руководителя, консультанта и рецензента, и ориентируется на выставленную научным руководителем оценку. Тем не менее оценка научного руководителя напрямую на оценивание работы не влияет.

Если работа имеет сугубо теоретический характер, то есть относится к классу «Теоретическое исследование», критерии Т2, П1 и П2 для неё не оцениваются. Предполагается, что «Сравнение с аналогами» (то есть близкими исследованиями) выполнено и является частью обзора (то есть оценивается по критерию О2).

Получение оценки «Плохо» (0) или двух оценок «Достаточно» (1) автоматически влечёт за собой оценку «незачёт» (F), кроме критериев, неприменимых к данному виду работы. Во всех остальных случаях итоговый процент освоения курса оценивается как отношение взвешенной суммы баллов по каждому из критериев всех групп к максимально возможной оценке. Веса определяются на основании вида практики:

Критерий	Решение	Эксперимент	Производственное задание	Сравнение	Теоретическое исследование
В1	3	3	3	3	3
В2	3	3	3	3	4
В3	2	4	1	1	3
В4	3	3	3	3	3
В5	2	2	2	2	3
О1	2	2	2	2	3
О2	3	4	2	3	4
Т1	3	4	2	2	3
Т2	3	3	3	5	0
П1	3	3	5	2	0
П2	3	4	2	4	0

Перевод в оценки делается по общей для СПбГУ шкале. Для годов поступления ранее 2020 оценка ECTS не выставляется, оценивание происходит только по шкале «зачёт/незачёт».

Итоговый процент выполнения	Оценка СПбГУ при проведении зачёта	Оценка ECTS
0-49	Не зачтено	F
50-60	Зачтено	E
61-69	Зачтено	D
70-79	Зачтено	C
80-89	Зачтено	B
90-100	Зачтено	A

Каждый из членов аттестационной комиссии оценивает документы независимо от остальных и принимает решение о выставлении зачета самостоятельно. Итоговая оценка определяется усреднением оценок членов комиссии.

6. Требования

6.1. Требования к тексту

1. Отчёт пишется индивидуально.

- (а) Попытка сдать чужой текст приводит согласно правилам обучения СПбГУ к отчислению без права восстановления.
- (б) Любой фрагмент текста длиннее шести слов, встречающийся в другом тексте (или очень близкий) и не оформленный как прямое цитирование, считается плагиатом и автоматически влечёт 0 баллов по критерию О1. Даже если проект групповой, общие куски текста недопустимы. Даже цитирование предыдущей своей опубликованной работы (включая практику прошлого семестра) недопустимо.
 - і. Исключение составляет работа над ВКР — комиссия ожидает увидеть целостный рассказ, так что переиспользовать фрагменты текста из зимней практики не только допустимо, но даже рекомендуется. Более ранние результаты придётся всё-таки изложить своими словами и сослаться на источник — иначе антиплагиат «зарезет» работу при сдаче.

2. Отчёт пишется в научном стиле:

- (а) не допускается использование местоимений «я», «мой» и т.д., при описании работы, отчёт пишется по возможности в пассивном залоге (не «я сделал», а «было сделано»);
 - (б) любое утверждение, сделанное автором, должно быть подтверждено или легко проверяемо: например, «как известно, Java — лучший язык программирования» недопустимо, тогда как «по данным рейтинга ТЮВЕ [1] на декабрь 2018 года Java является самым популярным языком программирования» допустимо;
 - (с) настоятельно рекомендуется использовать устоявшуюся терминологию языка текста работы, где она существует, не использовать сленг и жаргонизмы; переключаться на другой язык посреди предложения нельзя.
3. Используется выравнивание по ширине, для разделения абзацев — абзацный отступ.
 4. Точки в конце заголовков не ставятся.
 5. Разделы должны быть осмысленны и без заголовков. То есть нельзя писать в заголовке, например, «Haskell» и начинать раздел с «Это язык...».
 6. Списки и перечни в тексте оформляются по грамматическим правилам языка текста⁶
 7. Страницы отчета нумеруются, начиная с титульного листа. Цифру, обозначающую порядковый номер страницы, ставят в середине нижнего колонтитула страницы. На самом титульном листе номер не ставится.
 8. В заключении приводится ссылка на репозиторий с практической реализацией либо явно указывается, что код закрыт или отсутствует. Если проект групповой, указывается также имя аккаунта автора, позволяющее однозначно определить его вклад в проект.
 9. Список литературы оформляется по ГОСТ Р 7.0.5-2008⁷
 - (а) Для всех гиперссылок (и в списке литературы, и в подстраничных сносках) указывается дата последнего обращения (то есть дата, когда ссылка ещё точно вела туда, куда автор хотел сослаться).
 - (б) На все пункты списка литературы должна быть ссылка из текста.
 - (с) Ссылка в тексте является частью предложения: неправильно писать «инструмент такой-то. [1]», правильно «инструмент такой-то [1]». Перед открывающейся квадратной скобкой ставится неразрывный пробел.
 10. Рисунки, таблицы и листинги должны быть подписаны и пронумерованы, на них должна быть ссылка из текста. Используется сквозная нумерация. Заимствование иллюстративных материалов допустимо, но только со ссылкой на источник (прямо в подписи к рисунку, таблице и т.п.).
 11. Вставка листинга или таблицы в виде изображения недопустима.
 12. Орфографические и пунктуационные ошибки в тексте не должны существенно затруднять его чтение.

6.2. Требования к презентации

1. Выступление не должно длиться более восьми минут.
2. На титульном слайде должно быть название работы, ФИО защищающегося, учёная степень, учёное звание и ФИО научного руководителя. Учёное звание пишется после степени, должность — до. Например, «проф. каф. СП, д.ф.-м.н., проф. А.Н. Терехов».
3. Должен присутствовать слайд с целью работы и задачами.
 - (а) Если проект рассчитан больше, чем на один семестр, следует явно сформулировать задачи на проект в целом (даже уже выполненные) и задачи на конкретную практику.
4. Должен присутствовать слайд с результатами, выносимыми на защиту. Результаты должны соответствовать поставленным задачам.
 - (а) На защиту выносятся только результаты за один семестр.
5. Для групповых работ презентация должна быть своя у каждого участника, и рассказывать о вкладе участника в общий проект, а не про проект в целом (хотя описать весь проект во введении всё-таки нужно).
6. Презентация должна быть в формате .pdf.

⁶ для русского языка см. <https://infogra.ru/typography/11-pravil-oformleniya-spiskov-i-perechnej> (дата обращения: 19.07.2021г.)

⁷ГОСТ Р 7.0.5, URL: <https://docs.cntd.ru/document/1200063713> (дата обращения: 19.07.2021г.)

7. Должны быть указаны номера слайдов и общее их количество (например, «7/12»).
8. Нельзя читать со слайдов. Вообще, на слайдах не должно быть много текста.
9. Зачитывать текст выступления нельзя, ожидается живой рассказ о работе.
10. Знаки препинания в конце заголовков, пунктов списков и перечислений на слайдах не ставятся.
11. Шрифты с засечками на слайдах не используются.

6.3. Требования к репозиторию с реализацией

Если репозиторий ваш или вы активно в нём работаете:

1. В корне репозитория должна быть лицензия (любая, но лучше использовать какую-либо из разрешающих: Apache License 2.0, MIT, любую из BSD).
2. В корне репозитория должен быть файл README.md, где написано, что это за проект, как его собрать и запустить.
 - (a) Описания из README.md должно быть достаточно, чтобы воспроизвести эксперимент, описанный в тексте работы. Например, если для запуска требуются отдельные данные, в README.md должно быть написано, где их взять.
3. В случае, если используются компилируемые языки, должна быть настроена система непрерывной интеграции, проверяющая как минимум компилируемость проекта, либо в README.md должно быть написано, почему её нет⁸.

7. Рекомендации

7.1. Рекомендации по планированию работы

Методология работы критически важна для успеха, причём методологические ошибки могут сделать работу несостоятельной и их невозможно быстро исправить. Самый частый и самый критичный методологический провал — когда работа делается без оглядки на уже существующие результаты, а на защите оказывается, что задача уже была давно решена и проделанная работа проделана впустую. Также часто случается, что работа проделана, но решает не поставленную задачу, либо даже решает то, что нужно, но экспериментальная часть доказывает что-то другое. Обычно за такими вещами должен следить научный руководитель, но у него не всегда хватает времени и сил, поэтому кое-какие соображения приводятся здесь.

- Типичный план работы приводится здесь в разделе 3, стоит его придерживаться. Перед тем, как браться за задачу, стоит выполнить быстрый поиск аналогов и сообщить предполагаемому научному руководителю о своих находках — возможно, научный руководитель сам не знал, что то, что он хочет, существует.
- Также перед тем, как браться за работу, стоит понять актуальность работы, и изложить это в письменном виде как черновик введения, обсудив с научным руководителем. Отвечать за актуальность и новизну на защите придётся защищающемуся, так что полностью довериться научному руководителю в таких вопросах нельзя — *обучающийся сам должен понимать и уметь убеждать других в важности и полезности того, что он делает.*
- Ещё следует оценить объём задач — научный руководитель, как правило, представляет себе объём работы, но оценить его самостоятельно также очень важно. Задачи могут быть слишком простыми — это не повод радоваться, поскольку на защите придётся объяснять, куда было потрачено время. Задачи могут быть слишком сложными, и тогда на защите будет очень тяжело объяснить, почему такие плохие эксперименты и работа не закончена.
- Начать собственно работу всегда следует с изучения литературы и аналогов, и делать это не для галочки, а в поисках ответов на вопросы, вытекающие из постановки задачи. На это может уйти почти весь осенний семестр, и хороший обзор может быть самостоятельным публикабельным материалом.
 - Самый сложный вопрос при этом — что читать. В силу принципа «Publish or perish» в любой сколько-нибудь значимой предметной области публикуются десятки тысяч статей в год, большую часть из которых читать нет никакого смысла. Следует ориентироваться на число цитирований источника (с поправкой на год публикации — более свежие работы имеют меньше цитирований, но могут быть лучше), на индекс Хирша авторов, предпочитать хорошие обзорные работы, но также не забывать и про релевантные «tool papers» и «case studies».
 - Хорошим подспорьем в этом может быть работа B. Kitchenham et al, Guidelines for performing Systematic Literature Reviews in Software Engineering.

⁸Для некоторых технологий настроить CI может быть отдельной учебной практикой, тогда можно обойтись и без CI.

- Также важно быстро определиться с методикой эксперимента (ещё до того, как что-то будет написано) и проверить, вместе с научным руководителем, что эксперимент действительно проверяет то, что декларировалось в постановке задачи.
- Важно хотя бы раз в неделю синхронизироваться с научным руководителем и консультантом, они могут подсказать, если работа пошла не в том направлении. При этом не надо бояться признать свои ошибки и неудачи — даже сообщение руководителю, что прогресса нет, лучше, чем полное отсутствие коммуникации.
- При написании кода стоит следовать традиционным практикам программной инженерии: инкрементальная разработка с регулярными демонстрациями научному руководителю, компилируемость кода и проходящие юнит-тесты на всех этапах разработки, наличие CI и хотя бы некоторой документации.
- Помогает также выписать все задачи, составить себе календарный план их выполнения и в ходе работы следить, что всё успевается. Могут помочь инструменты типа GitHub Projects, Trello или Pivotal Tracker. Реализационную часть нужно планировать так, чтобы осталось время на апробацию и написание текста, с учётом технических рисков. На текст следует заложить как минимум неделю, апробация сильно зависит от работы и может быть очень трудоёмкой — нельзя, чтобы это оказалось сюрпризом в конце.
- Стоит просить у научного руководителя или консультанта код-ревью. Хотя бы раз за всё время работы, но лучше почаще — после реализации крупной части функциональности, например.
- При планировании стоит учесть сроки подачи работ на конференции, на которых могут быть опубликованы промежуточные результаты. Обычно студенческие конференции имеют сроки подачи в начале февраля и до конца марта — это значит, что в идеале большой и публикательный кусок работы со всеми экспериментами должен быть закончен в январе. Если работа началась в сентябре, это сложно. К концу марта обычно сроки подачи конференций послабее, там стоит представить свою работу в любом случае.

7.2. Рекомендации по написанию текста

7.2.1. Общие рекомендации

- Отчёт пишется не только для того, чтобы получить зачёт, но и, в том числе, как подробное описание работы для других обучающихся, которые, возможно, будут её продолжать и развивать. Следите за тем, чтобы работа была воспроизводима: должно быть можно взять текст, быстро найти там ссылку на исходники, там, в свою очередь, найти README.md с описанием процесса сборки и запуска, который был бы актуален хотя бы пару лет (то есть лучше всего предоставлять Docker-образ для сборки, настроить CI, указывать конкретные версии инструментов и библиотек). Следите за тем, чтобы работа была понятна тем, кто в вашей области не разбирается.
- Название работы должно быть понятно. Например, нельзя «DSL для MIRF», можно «Предметно-ориентированный язык для библиотеки MIRF».
- Соблюдайте общие правила вёрстки научных текстов.
 - Вместо чисел до 10 используются числительные. Например, неверно «массив делится на 2 части», верно «массив делится на две части».
- Используйте LaTeX для вёрстки текстов. Это именно рекомендация, а не требование, но рекомендация настоятельная.

7.2.2. Введение

Введение должно занимать где-то одну-две страницы и пишется по стандартной схеме, подводящей читателя к постановке задачи так, чтобы ему всё было понятно. Цели введения — обрисовать контекст работы, пояснить её актуальность (кому это надо, вообще в мире и конкретно откуда задача), обозначить вклад работы (пояснить научную или техническую новизну, спозиционировавшись относительно аналогов), и сделать это так, чтобы постановка задачи была понятна без дополнительных пояснений. Начинаются все введения с общеизвестной информации (в духе «программировать сложно»), затем переходим к специфике предметной области, которая уже может быть не знакома читателю (в духе «поэтому придумали визуальные языки»). Затем поясняем проблему, которую мы хотим решить (то, что в англоязычной литературе называется *gap* — то, чего не хватает человечеству, и что должна заполнить ваша работа, например, «но создавать визуальные языки тоже сложно, несмотря на работы [1, 2, 3]»). Затем обозначаем предлагаемый нами подход к решению и вклад работы (в духе «поэтому мы разработали передовую методологию метамоделирования такую-то»). На этом введение заканчивается и начинается следующий раздел — постановка задачи.

- Введение должно быть понятно широкому кругу специалистов, нельзя предполагать, что все читатели разбираются в предметной области работы.

- Введение часто единственная часть работы, которую читают — чтобы оценить, релевантна ли работа, содержит ли значимые результаты, достаточно ли грамотно написана, чтобы ей вообще можно было доверять. Поэтому введение должно быть написано максимально хорошо.

7.2.3. Постановка задачи

Постановка задачи должна одним предложением формулировать цель работы (то, чего в итоге хочется достигнуть) и дальше списком задачи, которые нужно решить, чтобы достичь цели. Задач обычно 3-5, и они обычно включают в себя обзор, проектирование и реализацию, апробацию и эксперименты. На что надо обращать внимание:

- тут уже не надо объяснять актуальность и важность работы, это должно быть сделано во введении;
- при формулировке цели и задач не должно появляться новых терминов или каких-то других новых знаний, как бы ни был велик соблазн использовать их для более точной формулировки задач;
- задачи должны быть специфичны для работы — просто «сделать обзор», «реализовать», «апробировать» не подойдёт (это легко проверяется мысленным экспериментом — достаточно просто представить себе, что в работе хочется починить микроволновку — если задачи всё ещё подходят, это плохо сформулированные задачи);
- список задач формулируется после того, как исследование выполнено, поэтому не нужно писать сюда то, что хотелось, но не было сделано — также, если была задача «посмотреть, что бывает, и реализовать выбранное», можно писать уже конкретно, что было реализовано;
- список задач должен быть достаточен для достижения цели;
- если отчёт — это промежуточный отчёт по годовой работе, следует явно разделить список задач на два: то, что было сделано в осеннем семестре, и то, что надо будет сделать в весеннем.

7.2.4. Обзор

Обзор можно разделить на три части: обзор аналогов, обзор используемого, обзор предыдущих результатов (в принципе, любая из этих частей может отсутствовать, но хоть какие-то близкие работы стоит упомянуть). Любой из этих обзоров пишется с определённой целью, например, чтобы показать, что могут аналоги, а что они не могут, или чтобы сравнить по какому-то критерию качество алгоритмов в какой-то области на данном этапе развития науки, или чтобы продемонстрировать, что конкретно вашу задачу никто не решал, но есть подобные решённые задачи. Определиться с целью обзора необходимо, поскольку цель определит ваши дальнейшие шаги — какие работы вы будете искать, какую информацию из них извлекать и как её в дальнейшем обрабатывать. В конечном итоге обзор должен подводить читателя к тому, что поставленная задача не была решена до вас, и что ваше решение или подход — это в широком смысле что-то новое.

Составление обзоров аналогов следует определённой методологии, и хорошим примером такой методологии является перенос систематических обзоров литературы из медицины в программную инженерию, сделанный Б. Китченэм с коллегами⁹. Систематический обзор — это трудозатратное занятие, которое при этом даёт методологически качественный результат. Если кратко, шаги систематического обзора следующие:

- Определяется цель обзора, предметная область, в которой он будет производиться, а также исследовательские вопросы, ответив на которые, вы достигнете цели обзора.
- Составляется стратегия поиска литературных источников: в каких источниках искать информацию, по каким ключевым словам, насколько ранние работы считаются релевантными (например, во многих областях работы старше 5 лет считаются устаревшими, но в некоторых узких областях это не так). На этом же этапе планируются меры по недопущению публикационного перекоса (publication bias), например, включение в перечень источников не только русскоязычных, но и зарубежных, в том числе и на сравнительно экзотичных языках¹⁰. С практической точки зрения хорошим единым индексом научных работ является Google Scholar¹¹.
- На основе стратегии поиска составляется первоначальный набор источников. Их число может быть очень большим — вплоть до тысяч. У каждого исследователя, принимающего участие в обзоре, должен быть к ним полный доступ.
- Определяются критерии качества работ, например, наличие экспериментов в тексте работы, ссылок на репозиторий, активная поддержка кода в репозитории, наличие сравнительного анализа в работе и т.д.

⁹Guidelines for performing Systematic Literature Reviews in Software Engineering, URL: https://www.elsevier.com/_data/promis_misc/525444systematicreviewsguide.pdf (дата обращения: 13.03.2023).

¹⁰Случаи, когда обзор литературы делался только на основе русскоязычных источников, бывали, но ничем хорошим защиты таких обзоров не заканчивались

¹¹Google Scholar, URL: <https://scholar.google.com/> (дата обращения: 13.03.2023).

- Определяются критерии включения работ в обзор и исключения из него. Например, если ваша цель — сделать обзор аналогов разрабатываемого решения, то вы будете искать все работы, в которых решается ваша или похожая на вашу задача. Если же ваша цель — сделать обзор аналогов на JVM, вы будете отбрасывать аналоги, разработанные, скажем, для .NET. Критерии качества работы также могут повлиять критерии включения и исключения — например, вы решили отбрасывать все работы без экспериментов. Критерии включения и исключения полезно включить в текст отчёта по практике.
- На основе критериев включения и исключения первоначальный набор фильтруется до набора релевантных работ, а также делается оценка качества. Их уже может быть сравнительно немного — несколько десятков или сотен. Рекомендуется проводить фильтрацию в несколько этапов: сперва просмотреть работы вскользь или только введение и заключение, чтобы понять, относятся они к вашей теме или нет, а затем оставшиеся читать более вдумчиво. Сколько бы этапов фильтрации ни было, каждый из них исследователям рекомендуется делать независимо друг от друга, а потом совместно решать обнаруженные разногласия по поводу включения или исключения конкретных работ.
- Формируется перечень информации, которая извлекается из каждой релевантной статьи (описание алгоритма, сравнение с аналогами, особенности экспериментов и т.д.), а затем производится её сбор из статей — в идеале, каждым исследователем независимо, а затем разногласия решаются совместно.
- На основе полученной из статей информации составляются ответы на исследовательские вопросы, и делается вывод по всему обзору.

Так как вы, скорее всего, будете свою работу делать в одиночку, в ряде пунктов не удастся получить независимую оценку несколькими участниками. Это нормально, но тем не менее методологию обзора стоит показать своему научному руководителю. Хотя описанная выше методология изначально относится к обзору литературных источников, она может быть применена без существенных изменений и в случае обзора аналогичных технологий, инструментов, моделей и т.д.

Обзор используемых инструментов, библиотек или технологий тоже должен пояснять, почему именно такие инструменты были выбраны, и не просто перечислять инструменты, но и приводить про них достаточно сведений, чтобы дальнейший текст не требовал отсылки к первоисточникам.

Обзор предыдущих результатов должен описывать результаты научной группы, в которой выполнялась работа, проекта, быть может результаты прошлого года автора же — чтобы читающему было понятно, что было до работы, и что было сделано в её рамках. На что стоит обратить внимание:

- обзор пишется не для галочки, а для чего-то — подтвердить актуальность темы, понять, откуда можно позаимствовать требования, решения, с кем можно сравниться;
- обзор должен быть согласован с постановкой задачи: если задачей было реализовать конкретный алгоритм в конкретной библиотеке, не надо делать обзор библиотек-аналогов, надо делать обзор реализаций конкретного алгоритма (например);
- должны быть какие-то выводы из обзора: если проводилось какое-то количественное или качественное сравнение, его стоит свести в таблицу, и в любом случае явно описать, что читатель должен был понять из обзора (например, что все аналоги не очень, а в этой работе сейчас покажут, как надо, или что обычно делают вот так, и в этой работе тоже будет так делаться);
- наличие прямых аналогов не означает провал на защите из-за неактуальности работы, как раз наоборот — если этой темой занимаются, значит, она актуальна;
- если пишется теоретическая работа, не стоит забывать про практические реализации, и наоборот, если работа чисто техническая, не надо игнорировать академические статьи и ссылаться только на техническую документацию.

В обзор должно быть вынесено всё, что сделано не автором. Или даже автором, но не в рамках данной работы — например, в обзоре вполне можно сослаться на работу прошлого года. В других разделах внезапных рассказов об используемых технологиях или чужих результатах быть не должно, как бы ни был велик соблазн, например, прямо в реализации рассказать об используемой библиотеке.

Прямых текстовых заимствований даже из своих работ быть ни в коем случае не должно. Должна быть ссылка и краткое описание содержимого работы. Иногда краткого описания не приводят, если источник не важен для понимания работы (например, «в работах [1-5] используется подход А, в работах [6-8] подход Б»). Однако следует помнить, что работа должна быть самодостаточна и нельзя рассчитывать, что читатель пройдёт по ссылкам. Поэтому важные для понимания работы следует подробно описать.

Обзор является одной из самых объёмных частей любой работы, на него обычно отводится где-то от трети до половины общего объёма текста. А если работа обзорная, то и девять десятых.

7.2.5. Описание реализации

В первом семестре работы над многосеместровым проектом описание реализации обычно состоит из одного раздела, описывающего архитектуру и планируемую реализацию. В последующих отчётах разделов, как правило, несколько; каждый раздел описывает решение одной из подзадач из постановки задачи. Например, если отдельно выносилось создание архитектуры, в работе должен быть раздел, который разработанную архитектуру описывает. Обычно также описывают детали реализации, может, в отдельных разделах отдельных крупных подзадач. Основной целью описания реализации является воспроизводимость работы — прочитав описание реализации должно быть понятно, как примерно можно повторить реализацию, на какие проблемы можно наткнуться и как их решить. По описанию реализации же комиссия оценивает техническую сложность работы и грамотность её описания.

На что следует обратить внимание:

- на следование стандартам — не следует рисовать неформальные картинки, если есть диаграммы UML, не надо описывать алгоритм текстом, если можно представить листинг с псевдокодом;
- на обоснованность описываемых решений; следует также по возможности описывать альтернативные решения и почему от них отказались (чтобы при дальнейшей работе над представленной системой следующие поколения обучающихся не пытались лишний раз всё переделать);
- на то, что описан используемый инструментарий — если из работы непонятно даже, какой язык программирования использовался, она не получит высокого балла;

7.2.6. Апробация/Экспериментальное исследование

Самая интересная часть работы, где автор пытается убедительно доказать, что получены содержательные и полезные результаты. В осеннем семестре реализация ещё не закончена, но этот раздел, тем не менее, тоже обязательно должен быть — следует описать план и методику экспериментов¹². Что именно будет в этом разделе, сильно зависит от постановки задачи. Если была поставлена задача разработать более быструю/точную/требующую меньше памяти реализацию, чем у аналогов, достаточно численно показать, что действительно стало лучше (или хотя бы не сильно хуже). Если задачей было сделать что-то более удобное, чем аналоги, надо провести эксперимент по сравнению эргономичности по какой-либо методике, подразумевающей количественную оценку удобства использования приложения (например, System Usability Scale¹³). Если, однако, задачей было сделать что-то, реализующее новую функциональность, численные оценки и сравнение может быть невозможно провести — в таком случае также может помочь юзабилити исследование (например, по методологии System Usability Scale), или менее структурированный опрос пользователей, описание тестовых сценариев. Однако всегда следует стараться получить независимые отзывы — если в работе написано, например, что приложение на Google Play имеет среднюю оценку 4.9, вопросов не возникнет, а если автор описывает, как сам же проверил основные сценарии и «вроде работает», у комиссии неизбежно будут сомнения в состоятельности работы.

Про методику выполнения эксперимента пишутся целые книги и научные статьи (и рекомендуется ознакомиться хотя бы с некоторыми), но вот некоторые базовые рекомендации.

- Эксперимент должен быть согласован с постановкой задачи — проверяться должно то, что цель работы достигнута. Если не было задачи сделать быстрее, замерять производительность нет особого смысла. Если была задача сделать такую-то категорию пользователей счастливее, докажите, что они стали счастливее (а вовсе не то, что программа работает как задумано — иначе получится подмена понятий).
- Необходима статистическая обработка результатов. При планировании эксперимента надо понять, какие из измеряемых параметров являются случайными величинами и какие распределения имеют. Например, время работы программы — в большинстве случаев случайная величина, имеющая нормальное распределение, которое характеризуется двумя параметрами: математическим ожиданием и дисперсией. Целью эксперимента на самом деле является не «замер времени», а достаточно точная оценка параметров распределения наблюдаемой случайной величины. Как это делать «настоящему», рассказывается в курсе математической статистики, но если вкратце, надо минимизировать случайные факторы (например, постараться, чтобы измеряемая программа работала одна на компьютере), выполнить несколько (не меньше 10) замеров, посчитать среднее и среднеквадратичное отклонение. Если среднеквадратичное отклонение сопоставимо со средним (например, всего вдвое меньше), подумать, почему так, и, возможно, переделать эксперимент (возможно, наблюдаемая величина распределена не так, как ожидалось, например, мультимодально — тогда надо разбить эксперимент на два). Если всё выглядит прилично, в тексте привести таблицу именно с матожиданием и среднеквадратичным отклонением (среднеквадратичное отклонение — это корень из дисперсии, его проще интерпретировать).

¹²Это соответствует индустриальным практикам, где приёмочные испытания описываются в техническом задании ещё даже до подписания контракта.

¹³System Usability Scale, URL: <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html> (дата обращения: 03.08.2021).

- Обращайте внимание на точность числовых значений — как минимум, число значащих цифр должно быть одинаковым для всех ячеек таблицы (не стесняйтесь использовать нули). Также надо понимать, за сколько значащих цифр можно отвечать: если среднеквадратичное отклонение получилось 10 секунд, приводить среднее время работы 121.56 секунды — это продемонстрировать свою безграмотность.
- Следует опасаться математических глупостей — например, при вычитании случайных величин дисперсии, конечно, не вычитаются друг из друга¹⁴.
- При описании эксперимента в области машинного обучения следует соблюдать особую осторожность, поскольку там сложилось много специфичных практик, которые надо знать и понимать. Например, использовать стандартные метрики оценки качества моделей — точность, полноту, F-меру, ROC-кривую. Понимать, какие метрики наиболее важны для решаемой задачи. Использовать стандартные методики постановки эксперимента — разделение датасета на обучающее, тестовое и валидирующее множества, кросс-валидацию.
- Очень важный раздел описания эксперимента — это *Угрозы валидности* (threats to validity в англоязычной литературе). В нём описывается, почему, по мнению автора, результаты эксперимента могут быть недостаточно корректны, и какие меры принял автор для того, чтобы минимизировать влияние угроз валидности эксперименту. Например, если юзабилити-исследование выполнялось только среди друзей автора, неизбежен «перекос» (англ. «bias») — друзья из желания порадовать автора будут иметь тенденцию завышать оценки. Наличие неустранённых угроз валидности не является недостатком работы, потому что все понимают, что абсолютно корректные эксперименты в программной инженерии требуют в буквальном смысле эпических ресурсов. Поэтому следует правдиво и не приукрашивая описать все потенциальные проблемы валидности эксперимента. Это не уменьшит ценность работы, а наоборот, покажет культуру автора.
- *Эксперимент должен быть воспроизводим*. Причём с минимальным количеством усилий со стороны того, кто будет пытаться это сделать. Для этого надо предоставить код в удобном для сборки и запуска виде, использованные данные, подробное описание методики. При оценке работы рецензент должен попытаться воспроизвести эксперимент, и если у него не получится это сделать без обращения к автору, это приведёт к снижению баллов.
- Следует описать параметры тестового стенда — процессор, оперативную память, используемую операционную систему и т.д., всё, что может повлиять на результаты. Это улучшит воспроизводимость (точнее, снимет вопросы, если у других людей получатся не такие цифры, как в работе). Если эксперименты по каким-то причинам проводятся на разных стендах, то указать, почему результаты экспериментов корректно сравнивать друг с другом.
- Если задача прикладная (типа написания приложения для конечного пользователя), это не отменяет воспроизводимости — приложение должно как минимум собираться из главной ветки репозитория и запускаться. Стоит также в описании эксперимента в работе привести несколько ключевых скриншотов, чтобы даже без запуска было понятно, о чём идёт речь.
- Воспроизводимость отменяется только закрытым характером работы. Если код или данные под договором о неразглашении, нарушать его нельзя. В этом случае требуется доказательство того, что кто-то, кроме автора, всё-таки работу видел и запускал — отзыв о внедрении, например. Раздел с экспериментами, конечно, всё равно должен быть, и описывать, как сам автор проводил эксперимент.

7.2.7. Заключение

Заключение должно кратко подводить итог работе и перечислять результаты, выносимые на защиту. Обычно заключение практически дословно повторяет постановку задачи, но в совершенной форме — например, если в постановке было «разработать», то в заключении «разработано», и с подробностями (поскольку результат уже был получен — если в задаче было «выбрать», то результат может быть «выбрано то-то и то-то»). Оформляется это так же, как и в постановке задачи — нумерованным списком. Допустимо и приветствуется писать про каждую задачу несколько более подробно, описывая, что в итоге получилось. Всё, что упоминается в заключении, должно было уже где-то встречаться в тексте, то есть результаты из ниоткуда в заключение писать нельзя.

В заключении должна быть приведена ссылка на репозиторий с исходным кодом проекта, или должно быть явно написано, что код закрыт. Даже если ссылка приводилась где-то ещё, приведите её и в заключении, чтобы не искать по всему тексту.

В заключении также можно привести направления развития работы, но только в том случае, если кажется, что эти направления могут стать темой отдельной практики или ВКР. Задачи в духе «доработать UI» писать не стоит, потому что могут возникнуть вопросы в духе «а что ж вы сразу не доработали?».

Здесь же можно написать благодарности, если хочется особо отметить людей, помогавших в работе. Научного руководителя благодарить не принято (это его работа), а вот консультанта, может быть, стоит (хотя консультант упоминается на титульнике, это уже в каком-то смысле благодарность — важнее поблагодарить кого-то, кто сильно помогал, но вообще нигде не упоминается).

Заключение обычно читают сразу после введения, так что это тоже очень важная часть работы.

¹⁴В это трудно поверить, но на математико-механическом факультете СПбГУ такое однажды принесли на защиту.

7.2.8. Список литературы

Список литературы — самая формализованная часть работы, оформляется по ГОСТ Р 7.0.5-2008 (он простой и короткий, состоит на 80% из примеров, так что очень стоит его просмотреть). Вот несколько рекомендаций:

- источников считается приличным иметь примерно столько, сколько страниц в работе;
- ссылаться следует в первую очередь на рецензируемые работы (статьи в научных журналах, книги), хотя на нерецензируемые (документацию, посты) тоже можно;
- при оформлении ссылки стоит давать возможно больше библиографических сведений, как минимум название и авторов статьи, год публикации, журнал/сборник, страницы, URL. Если есть место издания, издательство, редактор и т.п. — тоже нелишне указать. Для нерецензируемых источников должно быть как минимум название и URL (авторов может быть сложно установить, тогда их можно не указывать, либо указать название компании в качестве автора);
- для каждой электронной ссылки должна быть указана дата последнего обращения (это когда в последний раз видели, что ссылка ещё валидна и ведёт туда, куда предполагается);
- ссылаться на свои работы и работы своей исследовательской группы не только можно, но и нужно;
- все источники стоит реально прочитать — понятно, что проверять никто не будет, но практика показывает, что люди в статьях часто пишут интересные и полезные вещи, которые могут реально помочь в работе;
- управление ссылками может быть сложным, поэтому верстать работу рекомендуется в TeXе, используя для списка литературы biblatex со стилем ugost2008. Либо пользоваться библиографическими возможностями любимого редактора, но не делать ссылки вручную.

7.3. Рекомендации касательно презентации и доклада

Доклад должен занимать не более восьми минут, поэтому вся презентация должна занимать не более пяти-семи содержательных слайдов. За это время надо как можно лучше покрыть все пункты из критериев оценивания — комиссия обычно не читает тексты работ, а ставит оценку, опираясь на доклад. В восемь минут надо «упаковать» всё, чем занимались в течение полугода/года.

Приветствуется короткое демо на одну-две минуты (в дополнение к восьми на рассказ) с видеодемонстрацией работы программы. Живое демо устраивать не надо (хотя и возможно), лучше записать GIF или видеоролик, где показать, как программой пользоваться. Даже если это библиотека, можно продемонстрировать её работу, написав консольное приложение, вызывающее её методы. Однако важно учесть требования к неразглашению, если работа пишется в компании — демонстрацию интерфейса ещё не выпущенного продукта вполне могут запретить.

7.3.1. Типичная структура презентации

Типичная структура презентации следующая:

- титульный слайд;
- введение;
- постановка задачи;
- обзор;
- описание реализации;
- эксперименты/апробация;
- заключение.

7.3.2. Рекомендации по основным разделам презентации

Краткие рекомендации по основным разделам презентации следующие.

- Во введении за одну-две минуты надо ввести в предметную область работы слушателей, которые, скорее всего, вообще без идей, и заодно пояснить актуальность работы. Не надо тратить время на «воду» и определять общеизвестные понятия. Стоит явно сказать, откуда взялась тема, и не стесняться указать конкретную компанию, которая её предложила, если таковая имеется.

- Если работа является продолжением работы прошлых лет или на кафедре защищались похожие работы, надо спозиционироваться относительно них. Архив с работами на сайте кафедры в этом может сильно помочь.
- Слайд с постановкой задачи должен быть как в тексте — цель работы, дальше по пунктам список задач. Для осеннего семестра должно быть явно указано, что делалось осенью, что планируется весной.
- В описании реализации сложно соблюсти баланс между донесением до слушателей содержательности работы и уютом их в непонятных деталях. Хорошей практикой является строить рассказ вокруг слайда с архитектурой, где в виде диаграммы изображена структура системы, либо слайда с алгоритмом — словами рассказываются детали, при этом основная информация перед глазами у слушателей. Можно сделать ещё один-два слайда про детали, требующие отдельного пояснения, но сильно увлекаться рассказом про реализацию не стоит.
 - Особое внимание надо уделить формулам, если работа сильно математическая. Помогает сначала их со слайдов вообще убрать, посмотреть, понятна ли работа, и если да, то так и оставить. Дело в том, что много страшных формул, конечно, покажет, что докладчик умный, но больше неподготовленному слушателю не покажет ничего. А баллы за качество презентации за непонятный рассказ вполне могут снять.
 - Не надо писать код на слайдах — по тем же причинам, либо он будет слишком тривиальным, чтобы иметь смысл, либо слишком большим и сложным, чтобы его можно было прочитать и понять за те 30 секунд, что будет показываться слайд.
- Слайдов с апробацией/экспериментами может быть много (если надо продемонстрировать разные таблицы, графики и т.п.), но должен быть хотя бы один. К визуализации результатов экспериментов тоже надо относиться очень серьёзно, с учётом того, что слушателям надо будет за очень короткое время разобраться в результатах и сделать те же выводы, что и автор.
 - То есть, например, вывалить на слайд огромную таблицу — плохая идея.
 - Не подписывать оси, рисовать несколько линий на одном графике очень похожим цветом и т.д. — тоже плохая идея.
 - Хорошая идея — привести конфигурацию тестового стенда. Обязательно привести все статистические подробности (матожидание, среднее квадратичное отклонение, число запусков и т.п.).
 - Если работа продуктовая, приведите хотя бы один скриншот.
- Заключение должно быть таким же, как в тексте — по пунктам перечисляется всё, что выносится на защиту. При этом комиссия будет проверять соответствие результатов работы постановке задачи (чисто формально, какой результат какой задаче соответствует, и если биекцию установить невозможно — беда), и то, что про каждый результат есть слайд или хотя бы говорилось в докладе.
- Слайд с заключением должен быть последним, потому что он должен висеть на экране, пока задают вопросы к докладу. Раз уж на нём написано, про что был доклад.

7.3.3. Рекомендации по докладу в целом

Комментарии к оформлению презентации и доклада в целом таковы.

- Презентации на предзащитах (а они предполагаются в весеннем семестре) должны быть в целом такими же, как и презентации на защитах. Но поскольку возможно, что ещё не все результаты получены, можно оставлять некоторые таблицы/графики/места на слайдах пустыми, с устным комментарием, что как только всё доделается, тут будет то-то и то-то.
- Презентации на защиту лучше приносить в формате pdf, так он гарантированно откроется и будет выглядеть так, как предполагалось. .pptx использовать особенно не рекомендуется, потому что Microsoft Office есть не у всех, а в других офисных пакетах он может открыться в совсем неожиданном виде. Даже если Microsoft Office, нужных презентации шрифтов может не оказаться.
- Используйте векторные шрифты, а не растровые. На проекторе растр может быть не заметен, но если презентация будет в удалённом формате, слайды с растровым шрифтами выглядят плохо.
- На слайдах не должно быть много текста. Презентация — это иллюстрирование доклада, а не его дублирование. На слайдах необходимо размещать тезисы (то есть очень кратко основные мысли), а также картинки, графики, схемы, диаграммы, таблицы и т.п. Следует относиться к слайдам как к плакату, а не как к тексту работы.
- По этим же соображениям точки в заголовках и в конце пунктов списков не ставятся.
- Нельзя читать текст со слайдов.

- Если есть шанс, что защита будет в аудитории, следует учитывать особенности отображения на проекторе, т.е. стараться избегать слишком хитрых цветовых гамм своих картинок или графиков — лучше всего использовать спокойные, но контрастные цвета, либо какие-то сильно различающиеся стили заливки.
- Не используйте шрифты с засечками (например, Times New Roman).
- Размер шрифта должен быть таким, чтобы текст был читаем даже с задних рядов большой аудитории. Даже при дистанционных защитах (люди часто сидят с телефонов, им будет ничего не видно, если шрифт маленький).
- Так же, как в тексте, пользуйтесь стандартными графическими нотациями (UML, ER, DFD и т.д.).
- Не нужно сосредотачивать внимание на общеизвестных вещах, в то же время нельзя делать предположения о наличии у аудитории фундаментальных знаний в предметной области работы.
- Следует избегать фраз «не нужно пояснять», «всем понятно» и т.п. Если всем действительно должно быть понятно, то не надо об этом говорить. Но надо помнить, что то, что кажется очевидным докладчику, может быть неизвестно или непонятно слушателю (и комиссии). Причём, если докладчик знает чего-то, что комиссия не знает, баллы снимаются не комиссии. Сделать доклад понятным — важная задача докладчика.
- Все элементы презентации должны облегчать понимание работы, а не усложнять его, все должны служить главной цели — донесению сути работы до слушающих.
- Следует избегать жаргонизмов и англоязычных терминов, особенно если есть общепринятый русскоязычный аналог. Активное использование таких вещей может быть расценено как невладение терминологией и недостаточная погружённость в предметную область.
- Нужно стараться избегать ссылок вперед, т.е. фраз типа «это вы увидите на следующем слайде», «в конце презентации я вам покажу» и т.п.
- Даже если презентация делается удалённо, зачитывать подготовленный текст рассказа *нельзя* — это прекрасно слышно и это очень тяжело слушать.
- Если презентация делается в аудитории, во время выступления не стоит смотреть в экран. Смотрите на аудиторию.

Настоятельно рекомендуется в явном виде написать текст своего рассказа, и потренироваться на каких-нибудь сторонних слушателях (в идеале) или хотя бы рассказать себе, замеряя при этом время. Должно получиться примерно пять-семь минут (максимум в восемь минут на защите почти наверняка будет превышен, если восемь было уже на репетиции). Помимо всего прочего, написанный текст проще анализировать на предмет описанных выше рекомендаций.

Обязательно надо показать слайды научному руководителю, а лучше порепетировать с ним.

7.3.4. Чеклист по презентации

На защиту следует идти, только если в *каждой* строке таблицы можно честно поставить «да».

Титульник	Фамилия	
	Название работы на русском языке	
	ФИО и группа студента в формате ХХ.БХХ-мм	
	Научный руководитель с должностью, степенью и званием	
Введение	Краткий обзор тематики работы	
	Не нужно определять общеизвестные понятия	
	Применимость/полезность данной работы	
	Обоснование выбора именно этой темы	
	Если тема похожа на темы других работ (в том числе прошлых лет), явно описана разница	
Постановка задачи	Четкая формулировка цели данной работы и постановка задачи (отдельным слайдом)	
Обзор	Обзор аналогичных разработок/исследований/результатов	
Описание решения	Описание выносимых на защиту результатов, процесса или особенностей их достижения и т.д.	
	Не должно быть много технических деталей и формул	
Эксперименты	Проверяется именно то, что надо было сделать	
	Матстатистика там, где она нужна	
Итоги	Практически то же, что и на слайде с постановкой задачи, но в совершенной форме	
	Только то, что делал лично автор	
	Все результаты были в докладе до этого	
	Нет слайдов типа «Всё», «Вопросы?», «Спасибо за внимание»	
Техническое	Презентация в формате .pdf	
	Векторные шрифты	
	Шрифт достаточно крупный, чтобы все слайды были читаемы с телефона	
	Нет точек в заголовках и буллет-листах	
	Если на слайде несколько формул, они пронумерованы	
	Нет "Рис.:" в подписи к рисунку	
	Показали слайды научнику и он одобрил	
Текст	Нет абзацев текста на слайдах	
	Нет кода, если нет крайней нужды его разместить	
	Формул не очень много и каждый знак каждой формулы будет реально объяснён в докладе	
	Используются стандартные нотации	
Картинки	Облегчают понимание работы, а не усложняют его	
	Адекватное разрешение, без «мыла»	
	Чужие рисунки — со ссылкой на источник в подписи или в сноске	
	Свои рисунки — на языке доклада	
	Подписи к осям	
Рассказ	Не читаете со слайдов	
	Не предполагаете знаний в предметной области работы	
	Не говорите «не нужно пояснять», «совершенно очевидно, что» и т.д.	
	Не используете жаргонизмы и иноязычные термины не по делу	
	Отрепетировали выступление с научником	
	Укладываетесь в 7-8 минут	

В осеннем семестре в презентации должно быть всё, кроме, быть может, экспериментов. Вместо экспериментов приводится план экспериментов, все остальные требования в осеннем семестре тоже должны выполняться.

8. Особенности третьего курса

От третьего курса ожидается участие в работе исследовательской группы либо промышленного коллектива (например, на стажировке или работе), которое заканчивается созданием чего-то осмысленного и либо в научном, либо в техническом смысле нового. Тут уже надо уметь объяснять, чем созданный продукт лучше аналогов (в том числе и для исследовательских проектов — чем он раздвигает границы человеческого знания, хотя бы на чуть-чуть). Научная новизна или публикабельность не требуются, но провести сравнение с аналогами, понимать цель и смысл работы в более широком контексте, чем «так научник сказал» или «просто прикольно» уже необходимо. От третьего курса ожидается не только умение что-то сделать, но и понимание того, зачем и почему это делалось.

В остальном процесс работы над практикой не отличается от второго курса, разве что на третьем курсе нет выделенных часов в расписании на представление кафедр и предзащиты (поэтому предзащиты либо не проводятся вовсе, либо проводятся в весеннем семестре — в зависимости от загрузки преподавателей).

9. Особенности написания выпускной квалификационной работы бакалавра или магистра

Процесс написания выпускной квалификационной работы более бюрократизирован и более тщательно контролируется СПбГУ, поэтому отличий от третьего курса довольно много. Во-первых, защиты будут проходить перед Государственными Экзаменационными Комиссиями (ГЭК), собранными из представителей работодателей, причём даты защит назначаются ещё в марте и не переносятся, пересдать проваленную защиту нельзя, а несоблюдение сроков подачи текста влечёт автоматическую «неудовлетворительно» на защите и отчисление. Во-вторых, текст ВКР проходит обязательное рецензирование назначаемым формально рецензентом, и рецензия зачитывается на защите. В-третьих, от самих обучающихся требуется выполнение ряда формальных действий, причём строго в срок.

Итак, обо всём поподробнее.

9.1. План работы

ГЭК будет ожидать на защите годовую работу, поэтому хороший диплом начинают писать в августе-сентябре, причём если он основывается на предыдущих результатах, шанс успешной защиты выше. Поэтому и план работы над ВКР более конкретный, чем в случае с практиками:

- До 20-го сентября — выбор научного руководителя и темы работы. Если в случае с учебными практиками вопрос формулировки темы можно было отложить хоть до конца ноября, то в случае с ВКР в конце сентября формируют пул тем и научных руководителей, из которых потом надо будет официально выбрать свою тему, в это же примерно время готовятся письма о предложении темы работодателем.
 - Эти письма — первое бюрократическое препятствие на пути будущего выпускника. Каждая тема ВКР обязательно должна быть либо предложена компанией-работодателем, либо одобрена ею (это чтобы нельзя было защищать высосанные из пальца или сделанные по методичке дипломы). Для того, чтобы в дипломе было написано, что тема ВКР была предложена какой-то уважаемой компанией, обучающийся, если он там работает, сам должен договориться с начальством, чтобы подписали такое письмо. Шаблон письма можно найти тут: <https://disk.yandex.ru/i/GKPlfoXTD61xPg>, при этом оно, если возможно, должно быть зарегистрировано как исходящий документ в системе документооборота компании, то есть написано на бланке компании, подписано и с печатью. Если это слишком сложно, достаточно просто подписи (в идеале — с печатью), но как можно более высокого начальства (хотя бы потому, что это официальное письмо, направляемое в адрес другой организации, так что без должных полномочий подписавшего такое могут сильно наказать за самозванство). Формулировки возможно менять (например, фраза «В рамках сотрудничества» может не понравиться юридическому отделу).
 - Письмо о предложении темы можно и не делать, тогда тема будет предложена СПбГУ и подтверждена кем-то из работодателей.
 - Подтверждение актуальности темы может инициировать обучающийся (вот шаблон: https://disk.yandex.ru/i/4Lhy_sujiExxoA), по тем же правилам, что и письмо о предложении темы, и это полезно, если по тем или иным причинам письмо о предложении темы сделать не успели. Подтверждение темы уместно до конца октября.
 - Можно не делать никакие письма, тогда актуальность темы будет подтверждена кем-то из дружественных кафедре компаний, и именно они будут указаны в дипломе.

- Октябрь-ноябрь — представление темы кафедре. Это неформальное мероприятие, где надо за две минуты по двум слайдам рассказать, что, для кого и зачем делается, и что должно получиться в итоге. Тут могут попросить переформулировать или даже сменить тему.
 - Смена темы на любом этапе работы влечёт переподтверждение её актуальности у работодателя, поэтому без крайней нужды тему менять не стоит, а если пришлось сменить, заранее позаботиться о письме.
- Конец декабря — зачёт по осенней практике. Тут всё как обычно, но отчёты по практике публиковаться не будут, чтобы текст можно было переиспользовать в тексте ВКР и это не спровоцировало бы неадекватную реакцию системы антиплагиата.
- Конец января – начало февраля — утверждение рецензента. Рецензент формально назначается, но его вполне может предложить и сам обучающийся, либо научный руководитель/консультант (если есть консультант, обычно именно он предлагает рецензента, поскольку лучше всех ориентируется в области и имеет связи). Рецензента стоит выбрать заранее и начать с ним работать, например, отправив отчёт по зимней практике.
- Март-апрель — непредзащита. Слово «предзащита» в СПбГУ официально запрещено использовать (о чём даже приказ есть), после скандала, когда на защите кому-то поставили оценку существенно ниже, чем на предзащите, о чём он решил пожаловаться. Мы не ставим оценок на предзащитах, но приказ есть приказ, поэтому у нас «непредзащиты» и «непредпредзащиты». Непредпредзащита — это выступление по ВКР, насколько возможно приближенное к выступлению на защите. Структура презентации должна быть такой же, и если каких-то результатов пока нет, должны быть просто пустые слайды, или, лучше, незаполненные таблицы и графики. Текста ВКР для непредпредзащиты не нужно. По результатам непредпредзащиты выставляется (или не выставляется) зачёт за преддипломную практику.
- Середина-конец апреля — зачёты по весенним практикам. На выпускном курсе (будь то четвёртый курс бакалавриата или второй курс магистратуры) весенний семестр короткий, сессия начинается уже в середине апреля, чтобы до 15 мая её завершить, даже с пересдачами (напомним, защита не переносится, а с долгами до защиты не допускают). Поэтому и зачёт по преддипломной практике так рано. Для зачёта требуется выступление на непредпредзащите (его можно сделать хоть в начале марта, сильно заранее), отчёт и отзыв научного руководителя на преддипломную практику. Отчёт может быть уже почти настоящим текстом ВКР, но на титульном листе должно быть написано, что это отчёт именно по преддипломной практике.
- Где-то в начале-середине мая выходит приказ о допуске к защите, где окончательно фиксируется дата заседания и тема работы. После этого ничего менять уже нельзя.
- За две недели до защиты — загрузить текст в систему «Blackboard». Если этого не сделать за 14 календарных дней до даты защиты по приказу, на защите автоматически выставляется «неудовлетворительно», поэтому крайне не рекомендуется тянуть до последнего. После загрузки текст менять будет уже нельзя¹⁵.
- Примерно за две недели до защиты — непредзащита. Либо близко до, либо уже после крайнего срока загрузки текста — все результаты уже точно должны быть получены и работа должна быть полностью готова к защите. На непредзащите отшлифовывается выступление, исправляются последние огрехи в презентации работы. Если хотя бы один слайд по результатам непредзащиты надо серьёзно переделать, непредзащиту предложат пройти снова¹⁶.
- С 15 мая по 15 июня — собственно, защиты.
- Начало июля — выдача диплома. Диплом дают только в руки либо по нотариально заверенной доверенности, поэтому сразу после защиты далеко уезжать нельзя.

9.2. Выбор научного руководителя и темы

- *Научным руководителем ВКР может быть только сотрудник СПбГУ с учёной степенью* (то есть только кандидат или доктор наук — «бакалавр» и «магистр» в России учёной степенью не являются).
- Защита, как правило, проходит в ГЭК той кафедры, с которой научный руководитель, поэтому выбор темы и научного руководителя определяет, с кем и когда будет защита.

¹⁵Частая ошибка — загрузить текст с «Преддипломная практика» на титульном листе. Это можно поправить, написав заявление в студотдел, с объяснением, что произошла техническая ошибка, и попросив загрузить текст снова.

¹⁶Непредзащита не обязательна, потому что к моменту её проведения защищающиеся уже обучающимися не являются, нет никаких формальных механизмов не допустить их до защиты. Но за последние 15 лет никто от непредзащиты (даже повторной) не отказывался, потому что если «вынесут» на ГЭК — это отчисление.

- Есть традиционные исключения, когда руководитель числится на одной кафедре, но сфера его научных интересов близка к другой. Например, темы про формальные языки и трансляции, как правило, защищаются в комиссиях кафедры СП.
 - Однако защита возможна только в ГЭК кафедр, выпускающих по данному направлению подготовки. Если для «математического обеспечения» и «технологий программирования» есть выбор из четырёх кафедр (кафедры информатики и параллельных алгоритмов обычно защищаются в одном ГЭК, так что выбор из трёх комиссий), то для «программной инженерии» выбора нет, по этому направлению выпускает только кафедра системного программирования (и научный руководитель должен быть с этой кафедры).
 - Если формальности позволяют и научный руководитель не против, можно попросить руководителя практики защищаться на «чужой» для научного руководителя кафедре. Если формальности не позволяют (например, для обучающихся «программной инженерии»), защищаться с другим направлением *невозможно*.
- Брать темы можно из списка предложенных — для ВКР работодателя активно предлагают темы в августе. Можно продолжить свою работу предыдущего курса (третьего для бакалавриата и первого для магистратуры — для магистров это даже желательно, потому что хорошая магистерская пишется два года, а её план и ожидаемые результаты должны бы быть описаны ещё в мотивационном письме при поступлении).
 - Ещё хорошей идеей может быть взять тему на работе/стажировке, потому что совмещать работу с учёбой тяжело, но если на работе вы занимаетесь учёбой (написанием ВКР) и вам за это ещё и платят деньги — это прекрасно. Однако, как обычно, нужен научный руководитель из СПбГУ, который бы одобрил тему, а значит, как минимум, ориентировался бы в ней.
 - Выбранную тему надо будет вписать в таблицу, присланную руководителем практики, и заранее подготовить письмо о предложении/подтверждении актуальности темы, описанное выше.
 - Менять тему после того, как она попала в приказ о назначении темы и научного руководителя (как правило, начало ноября) можно только по заявлению в студотдел за подписью научного руководителя и проинформировав руководителя практики (вот шаблон заявления: <https://disk.yandex.ru/i/r-gPcPixqrPYRQ>). Смена темы делает необходимым новое подтверждение актуальности темы работодателем (даже если формулировка поменялась совсем чуть-чуть), этим лучше озаботиться заранее.

9.3. Выбор рецензента

Выбором рецензента придётся озаботиться в конце января, и это, как правило, вызывает непонимание как со стороны обучающихся, так и со стороны рецензентов (поэтому стоит часть текста ниже отправить рецензенту сразу же, вот она отдельным документом: <https://disk.yandex.ru/i/nLIJiWZUlwYc7A>). Обычно рецензента рекомендует консультант, иногда научник (для этого их надо попросить подыскать вам кого-нибудь), либо согласовать с консультантом и научником кандидатуру, которую вы считаете подходящей. Если не подать рецензента в срок, рецензент будет назначен учебно-методической комиссией, и скорее всего, он будет не очень лоялен. Рецензенту уже имеет смысл показать текст работы (зимней практики, например), хороший рецензент поможет работу улучшить. Задачей рецензента будет прочитать текст, посмотреть код, проверить, что всё работает — и написать рецензию на 2-3 страницы к защите, которая потом будет зачитана перед ГЭК. В частности, надо, чтобы рецензент имел доступ к коду (так что если код закрыт, выбирайте рецензентом того, у кого есть доступ или можно быстро предоставить).

Формальных ограничений на рецензента только два: рецензент не должен быть безработным (на момент написания рецензии) и не должен работать в организации, признанной экстремистской и запрещенной (например, Meta). Дополнительно, для магистерских ВКР, рецензент не должен работать в СПбГУ. Кроме этого, есть некоторые правила приличия:

- рецензент должен иметь профильное образование и опыт работы;
- рецензент не должен быть аффилирован со студентом, то есть:
 - не быть консультантом ВКР;
 - не работать со студентом в одной исследовательской группе/команде (в одной компании — вполне может);
 - * это нежёсткое правило, так как часто бывает, что экспертов в предметной области ВКР больше просто нигде не найти, однако если есть возможность получить рецензию «извне», это надо стараться делать;
 - состоять со студентом в родственных или слишком хороших личных отношениях.

Проверять это, понятно, никто не будет, но если вдруг государственной экзаменационной комиссии покажется, что рецензия может быть необъективной, ей будут меньше доверять.

9.3.1. Требования к рецензии

Ожидаемый объём — от одной до трёх страниц, где надо обязательно привести в виде списка выявленные недостатки в работе и указать рекомендуемую оценку по пятибалльной шкале (в т.ч. «неудовлетворительно»). Очень желательно в рецензии как-то продемонстрировать, что рецензент работу читали — кратко описать, про что каждая глава, например. Очень-очень желательно отразить качество практической части — качество кода, воспроизводимость экспериментов, оформление репозитория. В целом же рецензия пишется в свободной форме. Обратите внимание, что рецензент оценивает результат, а не ход работы.

Вот шаблон рецензии: <https://disk.yandex.ru/i/kd1MDCEthRIH9w>. Вот пример рецензии: <https://disk.yandex.ru/i/Cdm0CycdiBxBdg>.

Подпись рецензента и печать на рецензии не обязательны. Рецензия в электронном виде загружается перед защитой самим рецензентом в систему поддержки обучения Blackboard по логину/паролю, который пришлют где-то за пару недель до защиты. Рецензию можно вообще ввести в текстовое поле — факт работы в Blackboard по логину/паролю рецензента считается достаточным подтверждением её подлинности.

Однако кафедра системного программирования имеет традицию выкладывания материалов с защит своих выпускников на сайт (см. (<https://se.math.spbu.ru/theses.html?worktype=3>), так что очень желательно, чтобы рецензия была отдельным файлом, с подписью (можно .pdf со вставленным изображением подписи, это не критично). Печатей организации не надо.

На Blackboard лучше загружать рецензию и как файл, и как текст (просто скопировать текст рецензии в соответствующее поле) — так с ней будет проще работать на защите.

«Живая копия» рецензии не нужна — в СПбГУ электронный документооборот, живые документы не требуются.

Касательно сроков подготовки рецензии — формально рецензирование начинается с момента загрузки текста ВКР обучающимся в систему Blackboard, а это он обязан сделать не позднее 14 календарных дней до даты защиты. Крайний срок по рецензированию — не позднее пяти календарных дней до даты защиты. Поскольку рецензент никому ничего не должен, в этот срок можно и не уложиться (более того, студент имеет право защищаться без рецензии), однако это даёт студенту право оспорить результаты защиты и защититься ещё раз (что потребует нового созыва ГЭК и очень тяжело для всех участников, поэтому это крайне нежелательно). Хороший рецензент, однако, начинает работать с дипломником (если он тоже хороший и прислал текст заранее) задолго до этого срока, выдавая неформальные комментарии и давая возможность студенту их исправить, тем самым улучшив работу.

9.4. Документы для защиты

Вот что потребуется сдать в начале мая, чтобы защита прошла успешно.

- Текст ВКР, с русским и английским титульными листами, оформленными строго по шаблону <https://github.com/spbu-se/matmex-diploma-template> с опцией type, равной bachelor или master — строго за 14 дней до даты защиты. На титульном листе не должно быть написано «Преддипломная практика», и тема должна быть строго той, что указана в приказе (иначе могут снять с защиты). Максимальный объём текста — 40 страниц для бакалавров и 60 страниц для магистров. Однако это нежёсткое ограничение, в случае обзорной работы или крайней нужды может быть превышено. Однако чем короче текст, тем быстрее его читать рецензенту.
- Аннотация — это по одному абзацу текста на русском и английском языках, про что работа. Что-то вроде аннотации к научным статьям. Её надо будет вставить в Blackboard в момент загрузки текста, стоит согласовать её заранее с научным руководителем.
- Отзыв научного руководителя — за пять календарных дней до защиты. Научный руководитель, разумеется, пишет и загружает его сам, но хороший обучающийся должен проконтролировать, что всё в порядке, научный руководитель не забыл, отзыв написан и готов к загрузке в срок.
- Рецензия — так же за пять календарных дней до защиты, и так же это проходит даже без ведома обучающегося, рецензент грузит рецензию сам. Как и в случае с отзывом, следует проконтролировать, что всё хорошо. Если у рецензента что-то не получилось с Blackboard, он может направить рецензию на электронную почту reviews@spbu.ru, там её вручную выложит соответствующий сотрудник. Но в этом случае подпись уже обязательна.
- Презентация и дополнительные материалы (например, видео с демонстрацией работы системы) — за 1-2 дня до защиты. Это всё так же грузится в Blackboard, и это лучше не забыть сделать, поскольку минимизирует риски на защите (и делает для кафедры более удобным сбор материалов для выкладывания на сайте).
 - В тексте, отзыве, рецензии и презентации тема работы должна быть строго одинакова, иначе это может быть воспринято как подлог.

9.5. Как проходят защиты

- За одно заседание защищается максимум 8 человек, максимум 2 заседания в день.
- Порядок защиты фиксируется (когда вас распределяют по датам), где-то в начале-середине мая, приказом о допуске к защите. При необходимости поменять порядок выступления следует перед защитой согласовать это с председателем ГЭК.
- Выступление защищаемого, вопросы от членов ГЭК и аудитории, отзыв научного руководителя, рецензия (зачитывается научным руководителем или секретарём, если рецензента нет), вопросы по отзыву/рецензии, ответное слово (если надо).
- По окончании заседания ГЭК совещается, обычно от получаса до часа.
- Члены ГЭК выставляют оценки по критериям, каждый независимо. Критерии можно найти здесь: <https://edu.spbu.ru/gia.html>.
- Каждый член ГЭК ставит итоговую оценку (при этом критерии — это только рекомендации), оценки всех членов ГЭК усредняются и ставится итоговая. Обратите внимание, оценки научного руководителя и рецензента непосредственно не учитываются! Они имеют рекомендательное значение для ГЭК. Бывало так, что научный руководитель и рецензент ставят «отлично», итоговая оценка после защиты «удовлетворительно». Бывало и наоборот.
- Итоговая оценка заносится в протокол защиты и выставляется на Blackboard.
- Защищаемых приглашают, оглашают результаты, поздравляют с присвоением квалификации (или нет).

9.6. Общие рекомендации

- По особенностям написания текстов есть замечательное методическое пособие Д.В. Кознова и электив «Практика разработки документации» в последнем семестре обучения, который предназначен именно для помощи в структурировании текста и аккуратном описании результатов работы.
- ГЭК смотрит на код, поэтому репозиторий должен быть правильно оформлен:
 - аккуратные исходники с соблюдением стиля кодирования и комментариями;
 - README:
 - * плашки системы CI, анализаторов качества кода и т.д. и т.п. — чем их больше, тем профессиональнее кажется работа в глазах ГЭК,
 - * общее описание проекта,
 - * описание процесса сборки,
 - * описание воспроизведения эксперимента — у рецензента должно получиться воспроизвести эксперимент без посторонней помощи;
 - настроенная и проходящая сборка в системе непрерывной интеграции;
 - лицензия (какая хотите, мы рекомендуем Apache License 2.0 или MIT);
 - желательно, всё по-английски;
 - если работа является вкладом в чужой проект, то правильно оформленный запрос на включение изменений: с подробным описанием изменений, их внутреннего устройства, и, если уместно, с GIF-файлом, демонстрирующим работу;