

Практическое занятие 13-14

ДИСЦИПЛИНА	Технологии управления командами разработчиков программного обеспечения
ИНСТИТУТ	Институт перспективных технологий и индустриального программирования
КАФЕДРА	Кафедра индустриального программирования
ВИД УЧЕБНОГО МАТЕРИАЛА	Практические занятия
ПРЕПОДАВАТЕЛЬ	Зарипова Виктория Мадияровна
СЕМЕСТР	3 семестр, 2025-2026

Тема: Матрица Moscow

Цель занятий

Цель работы научиться управлять требованиями, научиться защищать проект от внесения изменений.

План занятий

- Ознакомиться с методом Moscow
- Заполнить матрицу Moscow требованиями по своему проекту
- Собрать дополнительные требования
- Подготовить проект изменений на основании новых требований

Итоги работы

При успешном выполнении работы студент должен получить следующие результаты:

- Заполненная матрица требований по шаблону Moscow
- Проект изменений на основании матрицы

Приложение 1: Метод MoSCoW

Метод MoSCoW — это техника приоритизации требований, широко используемая в управлении проектами, особенно в гибких методологиях разработки. Название метода является аббревиатурой, составленной из первых букв четырех категорий приоритетов:

Основные категории MoSCoW

- **M — Must have (Должно быть):** Критически важные требования, без которых продукт не будет работать или не будет иметь ценности. Эти требования обязательны к реализации в текущем релизе.
- **S — Should have (Следует иметь):** Важные требования, которые желательно реализовать, но без них продукт все равно может функционировать. Реализация этих требований значительно повышает ценность продукта.
- **C — Could have (Можно иметь):** Требования, которые было бы неплохо реализовать, если будут ресурсы и время. Эти требования повышают удовлетворенность пользователей, но их отсутствие не критично.
- **W — Won't have (Не будет в этот раз):** Требования, которые сознательно отложены на будущие версии продукта. Важно отметить, что это не отказ от требований, а перенос их на следующие итерации.

Преимущества использования MoSCoW

- Обеспечивает четкое понимание приоритетов для всех заинтересованных сторон
- Помогает фокусироваться на наиболее ценных функциях
- Облегчает процесс принятия решений при ограниченных ресурсах
- Упрощает коммуникацию между разработчиками и заказчиком
- Позволяет гибко управлять объемом работ при неизменном сроке

Процесс применения метода MoSCoW

1. **Сбор требований:** Соберите все требования от заинтересованных сторон.
2. **Анализ требований:** Проанализируйте каждое требование с точки зрения его ценности, сложности реализации и зависимостей.
3. **Распределение по категориям:** Распределите требования по четырем категориям MoSCoW.
4. **Согласование:** Обсудите и согласуйте распределение с заинтересованными сторонами.
5. **Регулярный пересмотр:** Периодически пересматривайте приоритеты, особенно при изменении условий проекта.

Практические рекомендации

При распределении требований важно соблюдать баланс. Рекомендуемое соотношение:

- "Must have" — не более 60% от общего объема требований
- "Should have" — около 20%
- "Could have" — около 20%
- "Won't have" — без ограничений, но важно их документировать

Избегайте ситуации, когда все требования попадают в категорию "Must have" — это обесценивает метод приоритизации.

Шаблон матрицы MoSCoW

ID	Требование	Категория	Обоснование	Проект выполнения
REQ-001	[Описание требования]	Must have	[Почему это критично]	[Границы реализации, ограничения, что не входит в объем]
REQ-002	[Описание требования]	Should have	[Почему это важно]	[Границы реализации, ограничения, что не входит в объем]
REQ-003	[Описание требования]	Could have	[Какую ценность добавляет]	[Границы реализации, ограничения, что не входит в объем]
REQ-004	[Описание требования]	Won't have	[Почему отложено]	[Предварительная оценка для будущих версий]

Пример заполнения матрицы для веб-приложения

ID	Требование	Категория	Обоснование	Проект выполнения
REQ-001	Авторизация пользователей	Must have	Базовая безопасность системы	Включает: авторизацию по email/паролю, восстановление пароля. Не включает: авторизацию через соцсети, двухфакторную аутентификацию. Макс. время ожидания ответа системы: 3 сек.
REQ-002	Личный кабинет пользователя	Must have	Основная функциональность	Включает: просмотр профиля, редактирование основных данных. Не включает: загрузку аватара, настройку уведомлений. Лимит полей профиля: не более 10.
REQ-003	Фильтрация результатов поиска	Should have	Улучшает пользовательский опыт	Включает: фильтрацию по 3-5 основным параметрам. Не включает: сохранение фильтров, комбинированные фильтры. Лимит одновременно применяемых фильтров: 3.
REQ-004	Темная тема интерфейса	Could have	Улучшает визуальный комфорт	Включает: переключение между светлой и темной темой. Не включает: настройку отдельных элементов темы, автоматическое переключение по времени суток.

REQ-005	Интеграция с соцсетями	Won't have	Отложено на следующую версию	Предварительная оценка: авторизация через 2-3 основные соцсети. Не будет включать публикацию контента в соцсети, импорт контактов.
---------	------------------------	------------	------------------------------	--

Приложение 2: Процесс управления изменениями в проекте

Процесс управления изменениями — это структурированный подход к переходу от текущего состояния проекта к желаемому будущему состоянию. В контексте проектной деятельности, это формальная процедура, определяющая как предложения по изменениям регистрируются, оцениваются, утверждаются, отклоняются и внедряются.

Типичные действия менеджера проекта в процессе управления изменениями

1. **Регистрация запроса на изменение:** Менеджер проекта документирует все поступающие запросы на изменение в специальном журнале или системе управления изменениями, присваивая им уникальные идентификаторы.
2. **Анализ влияния изменения:** Проводится оценка влияния предлагаемого изменения на объем работ, расписание, бюджет, ресурсы, качество и риски проекта. Часто используется матрица MoSCoW для оценки приоритета изменения.
3. **Разработка вариантов реагирования:** Менеджер разрабатывает альтернативные варианты реагирования на запрос об изменении (полное внедрение, частичное внедрение, отклонение, отложение).
4. **Представление на рассмотрение совету по контролю изменений:** Подготовка материалов и представление анализа влияния изменения и вариантов реагирования органу принятия решений (CCB — Change Control Board).
5. **Коммуникация решения:** После принятия решения, менеджер проекта информирует всех заинтересованных сторон о результате рассмотрения запроса и дальнейших действиях.
6. **Обновление проектной документации:** При одобрении изменения вносятся корректировки в план проекта, WBS, расписание, бюджет и другие проектные документы.
7. **Реализация изменения:** Координация действий команды по внедрению утвержденного изменения согласно плану.
8. **Мониторинг и контроль:** Отслеживание процесса внедрения изменения и его влияния на проект, внесение корректирующих действий при необходимости.
9. **Документирование уроков:** Фиксация полученного опыта и уроков для будущих проектов.

Риски и последствия неэффективного управления изменениями

Ползучесть объема работ (Scope Creep)

- Суть риска: неконтролируемое расширение требований без соразмерного увеличения сроков, бюджета и ресурсов.
- Признаки: растущий список «маленьких» изменений, частые срочные правки, отсутствие фиксированных границ реализации.
- Причины: нечеткие границы требований, слабая дисциплина согласования, желание «угодить всем».
- Последствия: снижение предсказуемости, ухудшение качества из-за распыления усилий, переработки команды.
- Профилактика: строгое описание «входит/не входит», формальный CR-процесс, лимиты на WIP, таймбоксы, MoSCoW для новых запросов.

- Метрики: доля Must в релизе, число CR в спринте, процент отклоненных CR, факт/план по объему работ.

Перерасход бюджета

- Суть риска: фактические затраты превышают утвержденный бюджет из-за недооценки изменений.
- Признаки: частые перерасчеты сметы, рост незапланированных статей расходов, «скрытые» часы.
- Причины: внедрение изменений без оценки ТСО, слабый контроль трудозатрат, недооценка интеграций.
- Последствия: урезание функциональности, заморозка проекта, потеря доверия заказчика.
- Профилактика: обязательная сметная оценка CR, буферы на изменения, CAPEX/OPEX-разделение, поэтапное финансирование.
- Метрики: CPI (cost performance index), отклонение факта от бюджета, доля незапланированных расходов.

Срыв сроков

- Суть риска: сдвиг ключевых вех из-за внедрения poorly-scoped изменений.
- Признаки: хронические переносы демо, увеличение незавершенных задач в конце спринта, рост очередей на код-ревью и тестирование.
- Причины: оптимистичные оценки, отсутствие перекрестной синхронизации зависимостей, параллельная работа над многими CR.
- Последствия: сдвиг рыночного окна, штрафные санкции, каскадные задержки интеграций.
- Профилактика: буферы по срокам, критический путь и управление зависимостями, WIP-лимиты, канбан-метрики потока.
- Метрики: SPI (schedule performance index), среднее время цикла CR, процент выполненных задач в спринте.

Снижение качества

- Суть риска: компромиссы по тестам и архитектуре ради «быстрого» внедрения изменений.
- Признаки: рост дефектов, повторные открытия багов, деградация производительности.
- Причины: отсутствие NFR-критериев в CR, пропуск регресса, «горячие» фиксы напрямую в прод.
- Последствия: технический долг, частые инциденты, падение удовлетворенности пользователей.
- Профилактика: Definition of Done с NFR и тест-критериями, автоматизация регресса, обязательный архитектурный просмотр CR.
- Метрики: дефекты на единицу функциональности, MTBF, тестовое покрытие, производительность до/после.

Конфликты с заинтересованными сторонами

- Суть риска: ожидания сторон расходятся с фактом из-за непрозрачности процесса изменений.

- Признаки: эскалации, параллельные «серые» каналы запросов, частые переквалификации приоритетов.
- Причины: отсутствие единого журнала CR, слабая коммуникация, непоследовательные решения ССВ.
- Последствия: затягивание согласований, блокировки, репутационные потери.
- Профилактика: единый реестр CR, SLA на оценку и решение, публичные протоколы ССВ, визуализация статусов.
- Метрики: время от регистрации до решения CR, доля CR, изменивших приоритет, NPS стейкхолдеров.

Демотивация команды

- Суть риска: постоянные неплановые изменения ломают фокус, вызывают переработки.
- Признаки: рост переработок, текучесть, снижение скорости (velocity), выгорание.
- Причины: незащищенный бэклог, «вечный срочный режим», частая переприоритизация.
- Последствия: потеря экспертизы, ухудшение качества, падение инновационности.
- Профилактика: стабильные окна изменений, четкие критерии срочности, защита спринта, ретроспективы с экшн-планами.
- Метрики: удержание, eNPS, средняя скорость, доля овертаймов.

Нарушение договорных обязательств

- Суть риска: изменения вносятся без корректировок контрактных приложений и SLA.
- Признаки: расхождения версий ТЗ, споры по трактовкам, претензии по SLA.
- Причины: неактуализируемая документация, отсутствие контроля версий, «устные договоренности».
- Последствия: штрафы, юридические споры, блокировка платежей.
- Профилактика: контрактные Change Orders, контроль версий артефактов, обязательная юр-проверка существенных CR.
- Метрики: число конфликтов по контракту, доля CR с оформленными допсоглашениями, соблюдение SLA.

Потеря контроля над проектом

- Суть риска: управление становится реактивным, решения принимаются «по факту» инцидентов.
- Признаки: нет актуального плана, метрики не собираются, ССВ собирается ад-хок.
- Причины: отсутствие системы управления изменениями, нехватка ролей/процесс-оунеров, «ручное» управление.
- Последствия: срыв целей, хаотичные релизы, рост операционных рисков.
- Профилактика: внедрение легковесного, но формального процесса СМ, роли владельца процесса, регулярные обзоры метрик.
- Метрики: частота ССВ, полнота журнала CR, доля изменений, прошедших полный цикл СМ.

Связанные риски безопасности и соответствия

- Суть риска: изменения обходят контроли безопасности и комплаенса.

- Признаки: деплой без security review, отсутствуют записи об оценке рисков, отклонения по аудитам.
- Причины: ускоренные «горячие» релизы, неинтегрированные проверки, недостаток SAST/DAST.
- Последствия: уязвимости, утечки данных, штрафы регуляторов.
- Профилактика: DevSecOps-практики, обязательные чек-листы комплаенса в CR, автоматизированные проверки.
- Метрики: количество security findings, время закрытия критичных уязвимостей, результаты аудитов.

Итог: каждое изменение должно проходить предсказуемый цикл регистрации, оценки влияния, принятия решения, реализации и пост-анализа с метриками эффективности.

Связка рисков с категориями MoSCoW

- Scope Creep → чаще возникает при некорректной маркировке Should/Could как Must. Рекомендация: лимит Must $\leq 60\%$, документировать Won't для прозрачного отсечения.
- Перерасход бюджета → повышенный риск при повышении категории с Could/Should до Must без пересмотра сметы. Вводить «стоимость приоритета» для апгрейда категории.
- Срыв сроков → рост вероятности при расширении пула Must в середине итерации. Защита: защита спринта, перенос на Should/Could следующего релиза.
- Снижение качества → характерно для срочных Must без NFR. Требовать DoD с NFR даже для Must.
- Конфликты стейкхолдеров → разночтения, когда ожидания Won't не зафиксированы. Явно вести список Won't и пересматривать по CCB.
- Демотивация команды → эффект от частых апгрейдов Could → Must. Ограничивать частоту пересмотра категорий и защищать фокус команды.
- Нарушение договорных обязательств → риск при эскалации категории до Must без обновления приложений к договору. Требовать Change Order.
- Потеря контроля → отсутствие правил перехода между категориями. Завести политику промоутов: критерии для Must, окна пересмотра, роли, метрики.
- Безопасность и комплаенс → попытка пометить как Must «горячие» изменения в обход security review. Политика: ни одно Must не минует DevSecOps чек-листы.
- Политика переходов между категориями MoSCoW
 - Must → Should: если риски и стоимость превышают ценность текущего релиза
 - Should → Must: только при влиянии на ключевые метрики или обязательства по SLA/регуляторике
 - Could → Should: при низкой стоимости и высокой синергии со спринтом
 - Любой → Won't: при неблагоприятном соотношении ценности к стоимости или конфликте со сроками критического пути

Чек-листы для оценки Change Request (CR)

- Чек-лист предварительной регистрации CR
 - ⇒ [] Ясная формулировка цели изменения
 - ⇒ [] Связь с бизнес-метриками или OKR
 - ⇒ [] Указан инициатор и владелец изменения
 - ⇒ [] Описание области влияния: компоненты, подсистемы, интеграции

- ⇒ ☐ Указаны зависимости и предположения
- ⇒ ☐ Первичная категория MoSCoW предложена инициатором
- Чек-лист оценки влияния
 - ⇒ ☐ Оценка на объем работ и сроки с вариантами исполнения
 - ⇒ ☐ Оценка затрат: трудозатраты, лицензии, инфраструктура (CAPEX/OPEX)
 - ⇒ ☐ Оценка рисков и план смягчения
 - ⇒ ☐ Оценка качества и NFR: производительность, безопасность, доступность
 - ⇒ ☐ Оценка влияния на UX и поддержку пользователей
 - ⇒ ☐ Необходимые изменения в документации и контрактах (SLA, приложения)
 - ⇒ ☐ Выбор категории MoSCoW на основе оценки (пересмотр приоритетов)
- Чек-лист решения CCB
 - ⇒ ☐ Принято решение: утвердить / частично / отложить / отклонить
 - ⇒ ☐ Обоснование решения с привязкой к метрикам
 - ⇒ ☐ Назначены ответственные, сроки и критерии приемки
 - ⇒ ☐ Обновлено артефакты: бэклог, план релизов, договорные документы
 - ⇒ ☐ Определены коммуникации: кто, что, когда сообщает
- Чек-лист реализации и пост-анализа
 - ⇒ ☐ Обновлено тест-кейсы и автотесты, пройден регресс
 - ⇒ ☐ Обновлено мониторинг и алерты, SLO проверены
 - ⇒ ☐ Проведено UAT и приемка по DoD
 - ⇒ ☐ Выпущены релиз-ноты и инструкции поддержки
 - ⇒ ☐ Пост-аналитика: фактические сроки, затраты, влияние на метрики
 - ⇒ ☐ Уроки и улучшения процесса внесены в ретроспективу

Защита проекта от внесения изменений

Матрица MoSCoW с детальным проектом выполнения является мощным инструментом для защиты проекта от необоснованных изменений:

- Четко определенные границы реализации помогают предотвратить расширение трактовки требований
- Документированные ограничения и "out of scope" элементы обеспечивают защиту от "ползучести объема" (scope creep)
- При появлении новых требований можно сравнить их с зафиксированными границами реализации
- Любые изменения в проекте выполнения должны проходить через формальный процесс управления изменениями
- Детализация проекта выполнения должна быть достаточной для однозначной трактовки требования всеми заинтересованными сторонами

Рекомендации по составлению проекта выполнения:

- Всегда указывайте, что **входит** и что **не входит** в объем работ по каждому требованию

- Определяйте конкретные количественные ограничения (количество записей, время отклика, число функций и т.д.)
- Описывайте взаимодействие с другими системами и компонентами
- Фиксируйте предположения и зависимости, влияющие на реализацию
- Согласовывайте проект выполнения с заказчиком до начала работ

При правильном использовании метод MoSCoW с детальным проектом выполнения позволяет эффективно управлять объемом работ, защищать проект от необоснованных изменений и обеспечивать четкое понимание границ реализации всеми участниками проекта.

Приложение 3: Примеры

Пример матрицы MoSCoW для модуля назначения встреч с врачом в личном кабинете пациента

ID	Требование	Категория	Обоснование	Проект выполнения
REQ-101	Просмотр доступных слотов для записи к врачу	Must have	Базовая функциональность модуля записи	Включает: отображение свободных слотов по дням и врачам, фильтрацию по специальности. Не включает: фильтрацию по рейтингу врачей, просмотр слотов более чем на 30 дней вперед. Макс. время загрузки данных: 5 сек.
REQ-102	Выбор формата встречи (онлайн/офлайн)	Must have	Необходимо для корректной организации приема	Включает: выбор между двумя форматами при записи, индикацию доступности формата для конкретного врача. Не включает: гибридный формат, детальное описание условий онлайн-консультаций. Доступность выбора формата только для врачей, поддерживающих оба варианта.
REQ-103	Подтверждение записи с уведомлением	Must have	Гарантия информирования пациента	Включает: email-уведомление, SMS-уведомление, отображение информации в ЛК. Не включает: push-уведомления, интеграцию с календарями. Время отправки уведомления: не более 5 минут после подтверждения записи.
REQ-104	Перенос времени встречи пациентом	Must have	Базовая гибкость в управлении своим расписанием	Включает: возможность однократного переноса времени не позднее чем за 24 часа до приема, выбор из доступных слотов. Не включает: многократные переносы, перенос менее чем за 24 часа, приоритетное бронирование слотов. Ограничение: не более 1 переноса на 1 запись.
REQ-105	Отмена записи пациентом	Must have	Необходимо для эффективного управления расписанием	Включает: возможность отмены не позднее чем за 12 часов до приема, автоматическое освобождение слота в расписании. Не включает: отмену менее чем за 12 часов (требуется звонок), автоматический возврат средств при платном приеме. Обязательное поле причины отмены.
REQ-106	История посещений врачей	Should have	Улучшает пользовательский опыт	Включает: список прошедших визитов с датой, временем и специализацией врача. Не включает: оценку качества приема,

				просмотр медицинских заключений. Глубина истории: до 12 месяцев.
REQ-107	Напоминания о предстоящем визите	Should have	Снижает количество пропущенных приемов	Включает: email-напоминание за 24 часа, SMS-напоминание за 2 часа. Не включает: настройку времени напоминаний, выбор каналов коммуникации, push-уведомления. Обязательное содержание: дата, время, врач, формат, адрес/ссылка.
REQ-108	Предварительное заполнение анкеты перед приемом	Should have	Экономит время на приеме	Включает: стандартную анкету с симптомами и жалобами, возможность приложить фото. Не включает: загрузку медицинских документов, интеграцию с медкартой. Объем анкеты: не более 10 вопросов.
REQ-109	Интеграция с календарем смартфона	Could have	Улучшает удобство планирования	Включает: экспорт данных о записи в календари Google и Apple. Не включает: двустороннюю синхронизацию, автоматическое обновление при переносе. Формат: стандартный .ics файл.
REQ-110	Оценка качества приема	Could have	Позволяет улучшать качество обслуживания	Включает: простую анкету обратной связи после визита. Не включает: развернутые комментарии, публичные отзывы о враче. Шкала оценки: от 1 до 5 по трем параметрам.
REQ-111	Выбор предпочтительного врача	Could have	Повышает лояльность пациентов	Включает: добавление до 3-х врачей в "избранное". Не включает: автоматическое уведомление о появлении слотов у избранных врачей, приоритетную запись. Функция доступна только авторизованным пользователям.
REQ-112	Видео-консультация непосредственно в ЛК	Won't have	Отложено до следующей версии	Предварительная оценка: интеграция с сервисом видеоконференций, предварительная проверка оборудования, возможность обмена сообщениями во время консультации. Не будет включать запись консультации, распознавание речи.
REQ-113	Автоматическая запись на повторный прием	Won't have	Отложено до следующей версии	Предварительная оценка: запись на повторный прием по рекомендации врача с напоминанием, интеграция с системой назначений. Не будет включать автоматическую оплату, серийную запись на регулярные процедуры.

Новые требования для модуля (для примера проекта изменений)

ID	Требование	Описание	Источник
NEW-001	Запись на прием для членов семьи	Возможность записывать на прием не только себя, но и членов семьи (детей, пожилых родителей и т.д.) через свой личный кабинет	Опрос пользователей
NEW-002	Система лояльности за регулярные посещения	Накопление баллов за посещения и возможность их обмена на скидки или бесплатные услуги	Маркетинговый отдел
NEW-003	Интеграция с лабораторными исследованиями	Возможность просмотра результатов анализов и возможность записи на повторный прием при отклонениях от нормы	Заведующий клиникой
NEW-004	Умная рекомендация врачей	Система должна рекомендовать подходящих врачей на основании симптомов и истории болезни пациента	Совет директоров
NEW-005	Автоматическая запись к профильным специалистам	После первичного приема у терапевта система должна предлагать записаться к узким специалистам на основе рекомендаций врача	Главврач

Проект изменений на основании новых требований

ID	Требование	Рекомендуемая категория	Влияние на текущие требования	Обоснование
NEW-001	Запись на прием для членов семьи	Should have	Требует модификации REQ-101, REQ-103, REQ-104, REQ-105	Функционал актуален для значительной части пользователей (особенно для родителей детей и взрослых с пожилыми родителями). Усложняет реализацию базовых требований, но значительно расширяет применимость системы.
NEW-002	Система лояльности за регулярные посещения	Won't have	Не влияет напрямую на текущие требования	Маркетинговая функция, не влияющая на основной процесс записи на прием. Может быть реализована в следующих версиях после успешного запуска основной функциональности.
NEW-003	Интеграция с лабораторными исследованиями	Could have	Возможно расширение REQ-106 (История посещений)	Функционал полезен, но требует существенной интеграции с лабораторной системой. Может быть реализован как дополнительный модуль без изменения ядра системы записи.

NEW-004	Умная рекомендация врачей	Won't have	Потребуется значительной модификации REQ-101	Функционал требует сложных алгоритмов и доступа к медицинским данным, что выходит за рамки текущего проекта. Целесообразно отложить до следующей версии.
NEW-005	Автоматическая запись к профильным специалистам	Could have	Пересекается с отложенным требованием REQ-113	Функционал ценен с точки зрения медицинского процесса. Может быть реализован в упрощенном виде (без автоматизации, но с удобным интерфейсом для ручной записи на основе рекомендаций).

В результате анализа новых требований рекомендуется включить в текущую версию проекта только требование NEW-001 (Запись на прием для членов семьи) с категорией Should have, а также рассмотреть возможность реализации упрощенных версий требований NEW-003 и NEW-005 с категорией Could have. Требования NEW-002 и NEW-004 рекомендуется отложить до следующей версии системы.

Необходимо учесть, что включение NEW-001 потребует модификации нескольких Must have требований, что может повлиять на сроки реализации. Поэтому важно обсудить с заказчиком, готов ли он к возможному увеличению сроков или уменьшению функциональности в других аспектах системы.