Кафедра «Техника и технологии»

ЭКЗАМЕНАЦИОННАЯ РАБОТА

по дисциплине:

«Архитектура информационных систем»

Направление подготовки/специальность 09.03.02 Информационные системы и технологии

(код, наименование)

Обучающийся Запьянцев Анатолий Николаевич

(ФИО полностью)

Группа И-107А Подпись\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(номер группы)

Форма обучения Очная

Проверил Ефимов Матвей Александрович

(Фамилия И.О. преподавателя)

Должность \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка\_\_\_\_\_\_\_\_\_\_

Подпись \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Челябинск, 2025 г.

Оглавление

[Связь моделей с требованиями: обеспечение согласованности и полноты 3](#_Toc205984679)

[Раздел 1: Трассировка требований через модели 3](#_Toc205984680)

[Раздел 2: Создание матрицы трассировки требований (RTM) 4](#_Toc205984681)

[Раздел 3: Проверка полноты покрытия требований в моделях 6](#_Toc205984682)

[Раздел 4: Выявление противоречий в требованиях через диаграммы 7](#_Toc205984683)

[Раздел 5: Управление изменениями и их влияние на трассировку 8](#_Toc205984684)

[Список литературы 8](#_Toc205984685)

### ****Связь моделей с требованиями: обеспечение согласованности и полноты****

### ****Раздел 1: Трассировка требований через модели****

1. **Понятие и цели трассировки требований**

Матрица трассировки требований – это инструмент, который помогает связать разные этапы разработки системы. С ее помощью можно проследить путь от бизнес-требований до кода и тестов. Связь моделей с требованиями является ключевым аспектом этого процесса, где визуальные модели (например, диаграммы UML) выступают в качестве промежуточного звена между абстрактными требованиями и конкретной реализацией.

Основные цели трассировки требований через модели:

* 1. *Анализ влияния изменений (Impact Analysis):* Позволяет быстро оценить, на какие компоненты системы повлияет изменение конкретного требования, и наоборот, как модификация элемента модели отразится на реализованных требованиях.
  2. *Обеспечение полноты покрытия:* Гарантирует, что каждое требование было учтено на этапе проектирования и для него существует соответствующий элемент в архитектуре системы.
  3. *Повышение качества продукта:* Понимание взаимосвязей помогает избежать упущений и логических ошибок на ранних стадиях, что напрямую влияет на качество конечного продукта.
  4. *Упрощение тестирования и верификации:* Облегчает создание тестовых сценариев, напрямую связанных с конкретными требованиями и элементами моделей.
  5. *Подтверждение необходимости функционала:* Помогает гарантировать, что каждый элемент дизайна и каждая функция в коде существуют для реализации конкретного, утвержденного требования, предотвращая "раздувание" проекта (scope creep). [[1](#_Список_литературы)] [[2](#_Список_литературы)]

1. **Типы связей в трассировке**

Трассировка может осуществляться в нескольких направлениях, что обеспечивает всесторонний контроль над проектом.

1. Прямая трассировка. Эта матрица показывает, как требования идут по стадиям разработки, например, к функциям и тестам. Здесь смотрим на связи так: бизнес-требования → функциональные требования → тест-кейсы.
2. Обратная трассировка. Помогает проверить, какие требования учли при разработке и тестировании. Например, чтобы не получилось, что функция есть, а бизнес-требованию не отвечает. Тут работаем со связями в обратном порядке: тест-кейсы → функциональные требования → бизнес-требования.
3. Двусторонняя трассировка. Тут оба подхода вместе. Можно смотреть связи и туда, и обратно. Можно проверить, какие тесты покрывают каждое требование. И ещё понять, зачем нужен каждый тест-кейс, связав его с требованиями. Для двусторонней трассировки обычно используют ту же матрицу, что и для прямой, только при прямой трассировке её читают по строкам. Бывает, что делают две таблицы: для прямой и для обратной трассировки, но это редкость. [[1](#_Список_литературы)] [[2](#_Список_литературы)]

### ****Раздел 2: Создание матрицы трассировки требований (RTM)****

1. **Структура и назначение матрицы**

Матрица трассировки требований (Requirements Traceability Matrix, RTM) — это инструмент, который помогает связать разные этапы разработки системы. С ее помощью можно проследить путь от бизнес-требований до кода и тестов. В зависимости от сложности проекта матрица может содержать различное количество атрибутов. Типичная расширенная матрица включает:

1. УникальныйID требования Уникальный код для каждого требования (например, BR-001, FR-01.1).
2. Описание требования: Краткое описание, в чем суть требования.
3. Источник требования: Ссылка на документ или человека, который выдал требование.
4. Приоритет: Насколько важно требование (например, Высокий, Средний, Низкий).
5. ID элемента модели: Ссылка на компонент, класс или другой элемент, который реализует требование.
6. Тип модели/диаграммы: Какой тип диаграммы (например, диаграмма классов).
7. ID тест-кейса: Ссылка на сценарий тестирования, который проверяет реализацию.
8. Статус: Как сейчас обстоят дела с реализацией и тестированием (например, Спроектировано, В разработке, Протестировано). [[2](#_Список_литературы)]
9. **Пример Матрицы трассировки требований (RTM)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID требования | Описание | Источник | Use Case | Класс | Состояние системы | Тестовый сценарий | Статус |
| REQ-001 | Регистрация пользователя | GTYFD | UC-1 | UserService | Неактивен | TEST-LOGIN-01 | Реализовано |
| REQ-002 | Шифрование паролей | - | C-4 | CryptoModule | - | TEST-SEC-05 | В процессе |

1. **Процесс создания и инструменты**

Создание RTM — это итеративный процесс, который начинается на ранних этапах проекта и продолжается на протяжении всего его жизненного цикла.

* 1. *Шаг 1: Идентификация требований: изучение* требования заказчика, выясняете их суть и выставление приоритетов.
  2. *Шаг 2: Создание трассировочной матрицы:* Разработка таблицы, в которой указываются связи между требованиями и другими элементами проекта, такими как дизайн, разработка и тестирование.
  3. *Шаг 3: Отслеживание требований:* на этом идёт котроль за выполнением требований на протяжении всего проекта, обновлявление трассировочной матрицы и регистрация любых изменений или обновлений.
  4. *Шаг 4: Верификация и валидация:* в конце проекта проверяется, были ли все требования успешно реализованы и соответствуют ли они ожиданиям заказчика.
  5. *Шаг 5: Анализ и обновление.* Анализ матрицы для проверки полноты покрытия. Матрица должна обновляться каждый раз при добавлении или изменении требований или элементов дизайна.

Хоть RTM можно и вести вручную в Excel, для крупных проектов это неэффективно и ведёт к ошибкам. Рекомендуется использовать специализированные системы (например, Siemens Polarion, IBM DOORS, Jama Connect), которые автоматизируют процесс создания и поддержания связей.

Для малых проектов: Google Sheets или Excel.

Для Agile-команд: Jama Connect.

Для крупных корпораций: IBM DOORS или Polarion [[2](#_Список_литературы)] [[4](#_Список_литературы)]

### ****Раздел 3: Проверка полноты покрытия требований в моделях****

1. **Анализ покрытия с помощью матрицы (Gap Analysis)**

Матрица трассировки является основным инструментом для анализа полноты покрытия. Анализ пробелов (Gap Analysis) с ее помощью позволяет выявить несколько ключевых проблемных зон.

* 1. *Непокрытые требования:* когда требование не связано ни с одним элементом модели. Это значит, что про него забыли на этапе проектирования.
  2. *"Бесхозные" элементы модели:* когда элемент на схеме ни с чем не связан. Это может быть лишняя функция или ошибка в документах.
  3. *Отсутствие тестового покрытия:* когда требование связано с элементом модели, но для него нет тестов. В этом случае нельзя проверить, правильно ли работает функция..

Чтобы было проще анализировать, используют матрицу покрытия, сделанную на основе RTM. Она показывает, как хорошо покрыты требования. [[5](#_Список_литературы)]

1. **Важность 100% покрытия**

Целью является достижение 100% покрытия именно *требований*, а не кода. Хотя на практике достижение абсолютного 100% покрытия кода тестами может быть нецелесообразным, для требований этот показатель является критически важным. Важно, чтобы все требования были покрыты на 100%, а не код. Если хоть одно требование не учтено в моделях, то система не будет соответствовать тому, что хотел заказчик. [[5](#_Список_литературы)]

### ****Раздел 4: Выявление противоречий в требованиях через диаграммы****

1. **Роль визуального моделирования**

Текстовые спецификации могут содержать скрытые логические противоречия, которые трудно заметить. Визуальные модели, особенно поведенческие диаграммы UML, переводят эти требования на графический язык, делая несоответствия очевидными.

* 1. *Диаграммы активностей (Activity Diagrams):* показывают бизнес-процессы. С их помощью можно увидеть неправильную последовательность действий или недостижимые условия.
  2. *Диаграммы состояний (State Machine Diagrams):* показывают, как меняется состояние объекта. С их помощью можно найти невозможные переходы (например, из Отправлен в Создан) или состояния, из которых нет выхода.
  3. *Диаграммы последовательности (Sequence Diagrams):* показывают, как объекты взаимодействуют друг с другом. С их помощью можно понять, что что-то вызывается не в том порядке или возникает блокировка. [[6](#_Список_литературы)]

1. **Практические примеры выявления противоречий**
   1. **Пример с диаграммой активностей:**
      1. *Требование 1:* " Пользователь должен войти в систему, чтобы положить товар в корзину ".
      2. *Требование 2:* "Неавторизованный пользователь может добавить товар в гостевую корзину, которая будет объединена с его основной корзиной после авторизации".  
         При построении диаграммы активности для процесса "Добавление товара в корзину" станет очевидно, что это не противоречие, а два разных сценария, требующих отдельного проектирования. Модель помогает визуализировать необходимость ветвления логики в зависимости от статуса авторизации пользователя. [[6](#_Список_литературы)]
   2. **Пример с диаграммой состояний:**
      1. *Требование 1:* "Документ в статусе 'На согласовании' может быть только отозван автором".
      2. *Требование 2:* "Администратор системы может принудительно утвердить любой документ в статусе 'На согласовании'".  
         Диаграмма состояний для объекта "Документ" наглядно покажет, что из состояния "На согласовании" ведут два перехода, инициируемые разными акторами ('Автор' и 'Администратор'). Это помогает уточнить и формализовать ролевую модель и права доступа в системе. [[6](#_Список_литературы)]

### ****Раздел 5: Управление изменениями и их влияние на трассировку****

1. Процесс управления изменениями (Change Control)

Проекты не так часто остаются статичными. Требования меняются, и управление — это крайне важная задача. Трассировка играет главную роль в этом процессе.

Стандартный процесс контроля изменений включает шаги:

* 1. *Запрос на изменение (Change Request, CR):* Это формальное описание предлагаемого изменения.
  2. *Анализ влияния (Impact Analysis):* На данном этапе матрица трассировки становится незаменимой. Аналитик определяет, какие компоненты системы, модели, тесты и другие требования будут затронуты.
  3. *Оценка и утверждение:* Команда оценивает трудозатраты и риски, после чего запрос утверждается или отклоняется.
  4. *Реализация и обновление артефактов:* Вносятся изменения в код, модели, тесты, и, что самое важное, обновляется матрица трассировки, чтобы отразить новые связи.

1. Поддержание актуальности артефактов

Трассировка — это частое действие, которая повторяется раз за разом. Устаревшая матрица трассировки может принести больше вреда, чем пользы, делая контроль более сложным. Внедрение формального процесса управления изменениями гарантирует, что все связи между требованиями и их реализацией остаются актуальными на протяжении всего жизненного цикла проекта. Это обеспечивает прозрачность, снижает риски и позволяет принимать обоснованные решения при развитии продукта. [[7](#_Список_литературы)]

### ****Список литературы****

1. Матрица трассировки требований: руководство для системного аналитика // URL: <https://habr.com/ru/articles/893636/>
2. **Что такое трассировка требований в проекте и почему она важна? //** URL:<https://habr.com/ru/articles/831922/>
3. **Трассировка требований: ключевые аспекты и методы** // URL:<http://lifeip.ru/sistemnyj-analitik/trassirovka-trebovanij-klyuchevye-aspekty-i-metody/>
4. **Работа с требованиями виды и уровни требований //** URL: <https://tenchat.ru/media/3138325-rabota-s-trebovaniyami-vidy-i-urovni-trebovaniy>
5. **Оценка тестового покрытия на проекте //** URL: <https://habr.com/ru/articles/270365/>
6. **Обзор ТРИЗ для системных и бизнес-аналитиков. Часть 3. //** URL:<http://analyst.by/articles/triz3>
7. **Управление изменениями требований: определение и процесс //** URL:<https://visuresolutions.com/ru/руководство-по-отслеживаемости-управления-требованиями/управление-изменениями-требований/?utm_source=chatgpt.com>