Вопросы:

1. Какие задачи включает в себя этап управления требованиями?

**Сбор (выявление) требований**

**Анализ и уточнение требований**

* Выявление конфликтов между требованиями
* Проверка на полноту, реалистичность и непротиворечивость
* Уточнение неясных формулировок ("быстро", "удобно", "гибко")

**. Документирование требований**

* Создание спецификаций (SRS, BRD, User Stories, Use Cases)
* Использование шаблонов и стандартов (например, IEEE 830)
* Применение моделей: диаграмм, прототипов, BPMN и др.

**. Согласование требований с заинтересованными сторонами**

* Проведение ревью, утверждение документа
* Устранение расхождений между ожиданиями и спецификацией
* Подписание "заморозки требований"

**Отслеживание требований (traceability)**

* Связь требований с задачами, тест-кейсами, кодом
* Использование ID-систем, матриц трассировки
* Помогает контролировать реализацию и тестирование

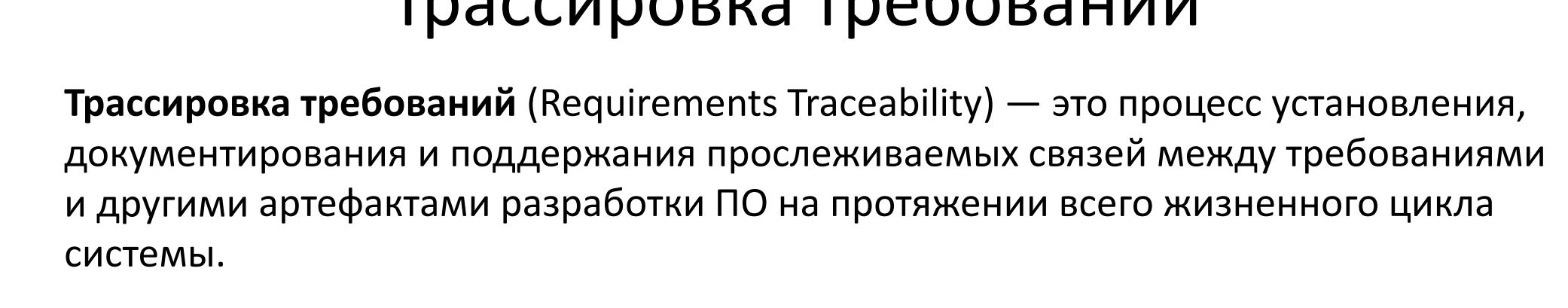
**Управление изменениями требований**

* Оценка влияния каждого изменения
* Процедура согласования изменений
* Обновление документации и оповещение команды

**Контроль выполнения требований**

* Проверка, что требования реализуются в соответствии с ожиданиями
* Участие в приёмке, тестировании
* Верификация и валидация требований

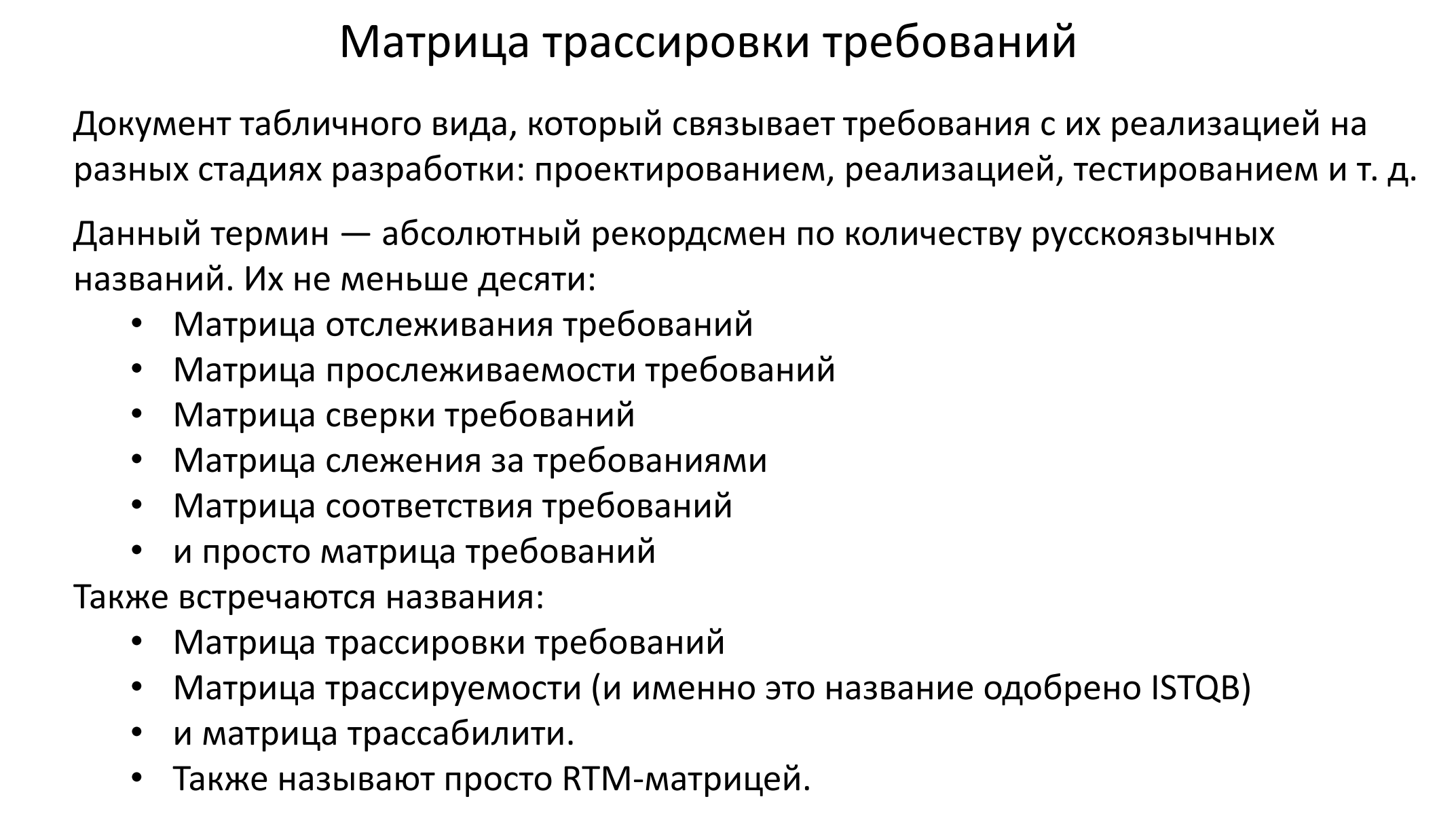
1. Что означает термин Трассировка требований?



**Трассировка требований** — это способ **отслеживать, как каждое требование связано с другими элементами проекта**: задачами, тестами, кодом, документацией и т. д.

Это способ **видеть, откуда пришло требование, где оно используется, и как его реализация проверяется**.

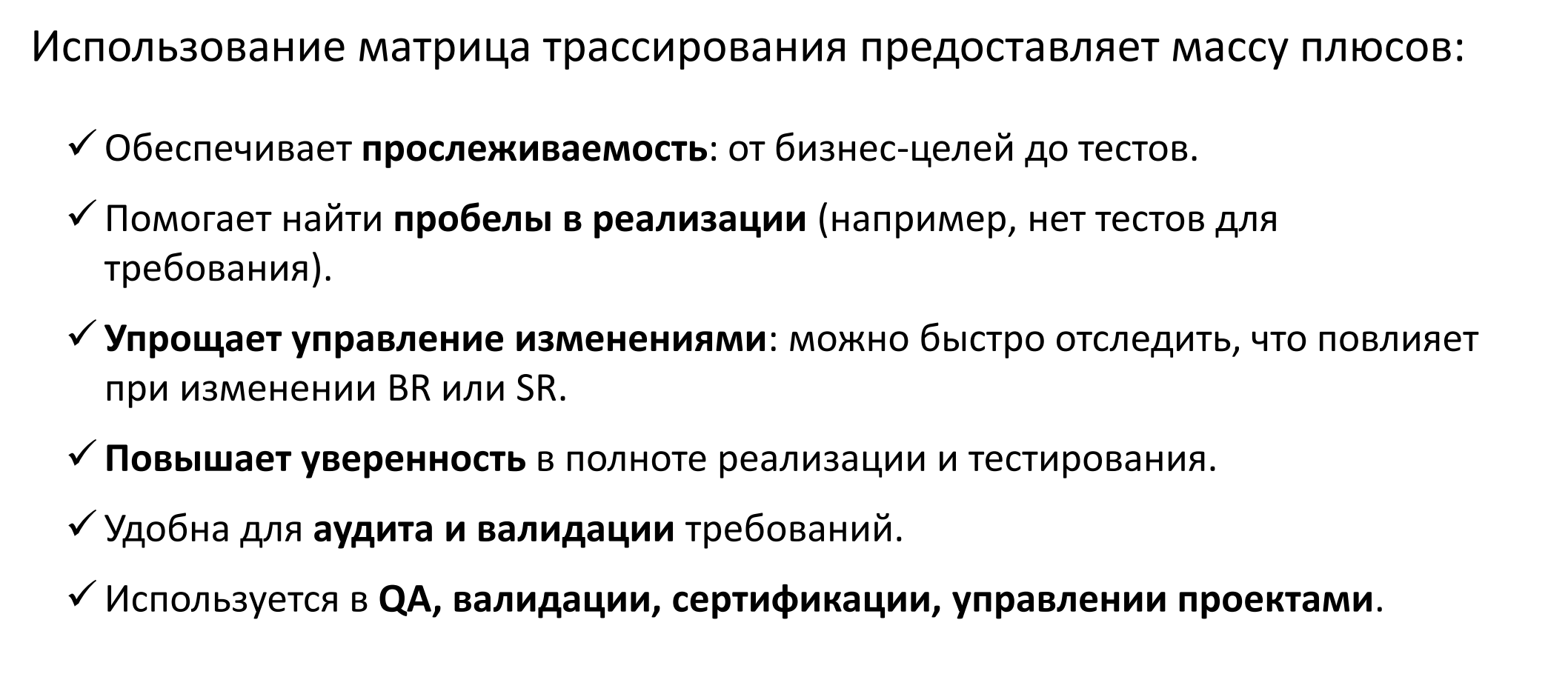
1. Что отображает матрица трассировки требований?



**Матрица трассировки требований** — это таблица, которая **отображает связи между требованиями и другими элементами проекта**, такими как:

* задачи разработки,
* тест-кейсы,
* модули системы,
* бизнес-цели и др.
* **Показывает, какие требования реализованы** (и где)
* **Какие тесты проверяют каждое требование**
* **Помогает управлять изменениями** — можно увидеть, что затронется
* **Обеспечивает контроль полноты** — не упущено ли что-то?

1. Какие преимущества дает использование матрицы трассировки?



**Контроль полноты реализации требований**

* Можно убедиться, что **каждое требование реализовано** в задачах и коде.
* Нет «потерянных» или забытых требований.

**✅ 2. Связь между требованиями и тестированием**

* Обеспечивает **полноту тестового покрытия** — видно, какие тест-кейсы соответствуют каким требованиям.
* Облегчает **проверку на соответствие** (верификацию).

**✅ 3. Упрощение управления изменениями**

* Если изменилось одно требование, ты **сразу видишь**, какие части системы, тесты и документы надо обновить.
* Это **снижает риски ошибок** при внесении изменений.

**✅ 4. Повышение прозрачности проекта**

* Легко показать заказчику или команде, **как требования реализуются и проверяются**.
* Поддерживает **трассируемость от бизнес-целей до реализации**.

**✅ 5. Удобство при аудите и сертификации**

* При проверках (например, ISO, ГОСТ, FDA) **документы и связи уже зафиксированы**.
* Быстро доказывается соответствие требованиям.

**✅ 6. Обнаружение несогласованностей**

* Можно выявить:
  + лишние элементы (тест или код, не привязанные к требованиям),
  + или наоборот — требования, которые нигде не реализованы.

**✅ 7. Поддержка качества разработки**

* Повышается уверенность, что **система делает то, что должна**, и **только это**.

1. Что включает в себя процедура управления изменениями?

**Инициирование изменения**

* Кто-то (заказчик, аналитик, разработчик и т.д.) предлагает изменение
* Создаётся **запрос на изменение (Change Request)**
* Запрос включает: описание, обоснование, приоритет, срочность

**2. Регистрация изменения**

* Изменение **вносится в реестр** изменений (в системе управления требованиями, например: Jira, IBM DOORS, Requiro, и т.п.)
* Присваивается **уникальный идентификатор** и статус

**3. Анализ влияния**

* Оценивается, **что затронет изменение**:
  + требования
  + задачи
  + архитектура
  + интерфейсы
  + тест-кейсы
  + сроки и ресурсы
* Проводится **оценка трудозатрат и рисков**

**4. Принятие решения**

* Группа управления изменениями (или ответственные лица) решает:  
  ✅ принять изменение  
  ❌ отклонить  
  ⏸️ отложить

**5. Реализация изменения**

* Изменения вносятся в:
  + документацию требований
  + технические задания
  + план проекта
  + код
  + тестовые сценарии

**6. Трассировка изменений**

* Обновляются **связи между артефактами**
* Обеспечивается **двунаправленная трассировка** изменений
* Обеспечивается соответствие новой версии всем требованиям

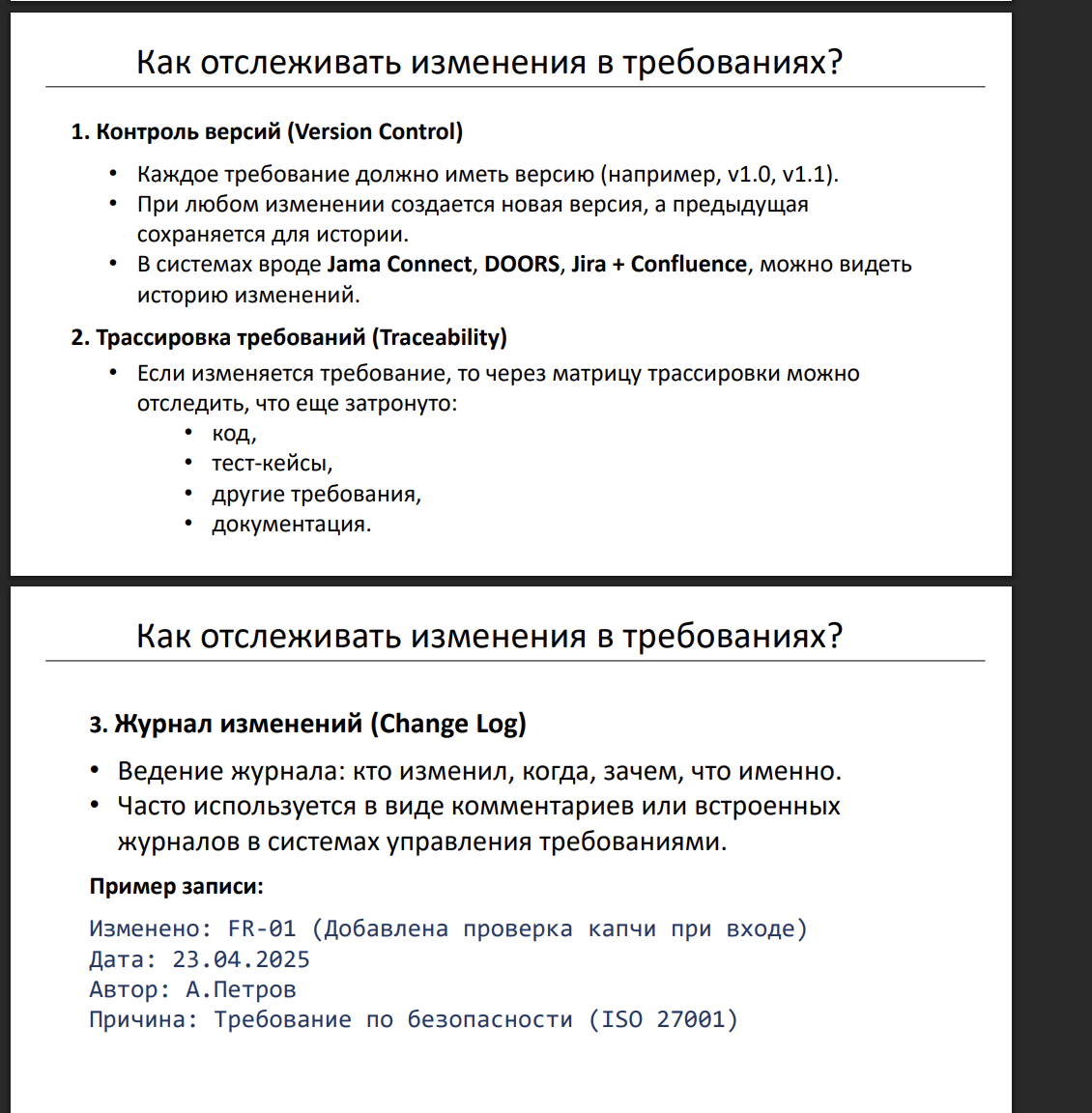
**7. Информирование заинтересованных сторон**

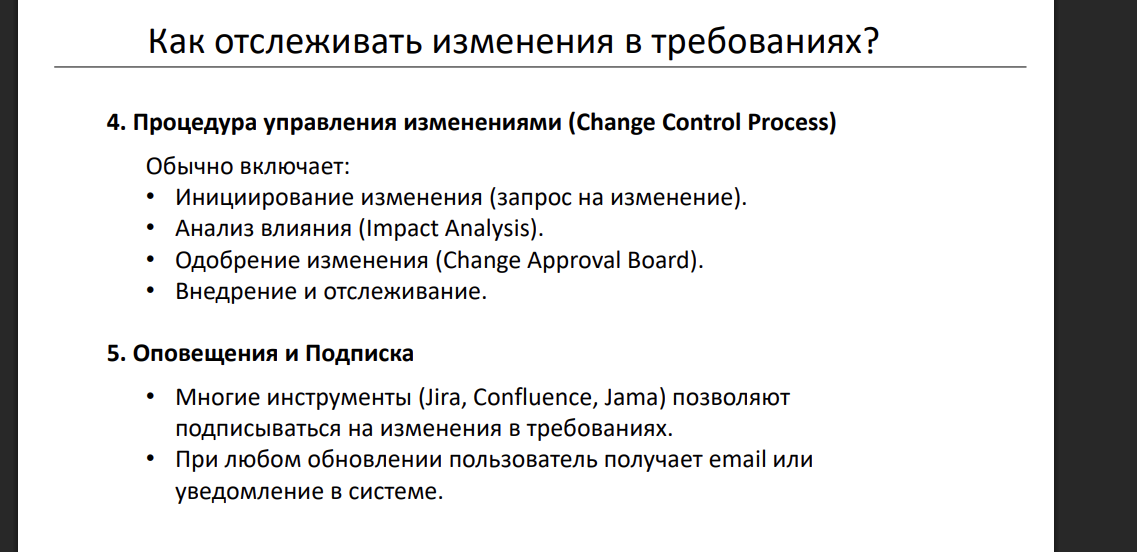
* Все, кого касается изменение, получают уведомление
* Обновляются статусы, комментарии, отчёты

**8. Аудит и контроль**

* Ведётся журнал всех изменений
* Возможен откат (если нужно)
* Проводится **верификация** (всё ли внесено) и **валидация** (правильно ли работает)

1. Как документально оформить изменение требования?





## **Создание запроса на изменение (Change Request)**

Это может быть отдельный документ или карточка в системе (например, в Jira, Confluence, IBM DOORS, ReqIF и др.).

### Основные поля:

| **Поле** | **Описание** |
| --- | --- |
| **ID изменения** | Уникальный номер (например, CHG-00123) |
| **Дата** | Дата подачи запроса |
| **Автор изменения** | ФИО или роль (например, заказчик, аналитик) |
| **Связанные требования** | ID или ссылки на изменяемые требования |
| **Описание изменения** | Что конкретно нужно изменить |
| **Обоснование** | Почему это изменение важно (причина, выгода) |
| **Тип изменения** | Добавление / Удаление / Модификация |
| **Влияние на проект** | Как это повлияет на сроки, бюджет, архитектуру, тесты и т.п. |
| **Решение** | Принято / Отклонено / Отложено |
| **Ответственный** | Кто будет реализовывать изменение |
| **Подписи / утверждение** | Кто согласовал (в случае формальной процедуры) |

## ✅ **2. Обновление документации требований**

Если изменение принято:

* Обновляется **спецификация требований** (например, SRS, BRD, backlog, user stories)
* Рядом с обновлённым требованием указывается:
  + новая версия или дата изменения
  + ссылка на соответствующий Change Request
  + описание, что именно поменялось (до / после)

## ✅ **3. Версионирование**

* Требованиям и документам присваивается новая **версия** (например, SRS v1.2 → v1.3)
* Ведётся **журнал изменений** (changelog), где кратко описаны все корректировки

1. Что означает анализ влияния изменения требования?

**Анализ влияния изменения требования** — это процесс оценки того, **как изменение одного требования повлияет на другие элементы проекта**, включая:

* другие требования,
* архитектуру,
* проектные документы,
* код,
* тесты,
* сроки и ресурсы.

1. Для каких целей используется RACI-матрица?

**RACI-матрица** используется для **распределения ролей и ответственности** между участниками проекта, особенно в задачах, где участвует много сторон и важно понимать, **кто за что отвечает**.

