* Создатель [Чуков Алексей Александрович](https://confluence.vsk.ru/display/~AChukov), отредактировано [только что](https://confluence.vsk.ru/pages/diffpagesbyversion.action?pageId=433293760&selectedPageVersions=39&selectedPageVersions=40)

1. Проблемы

Общие вопросы аналитиков:

* Постановка задачи:
  + Какие требования к постановке задачи?
  + Как понять, что работа над задачей завершена и её можно передать разработчику и тестировщику?
* Организация работы:
  + Как организовать пространство в Confluence?
  + Что ожидают разработчики и тестировщики?
* Состояние системы:
  + Каково текущее актуальное состояние системы?

 Нажмите здесь для раскрытия...

**Ожидание со стороны разработки**

Не везде были проявлены ожидания со стороны разработки

**Текущее состояние системы**

Возникают трудности с изучением текущего состояния функциональности системы

**Требования к постановкам задач**

Отсутствие четких требований к результатам работы системного аналитика

**Организация пространства**

Нет единого подхода к организации пространства и правил работы

Проблемы системных аналитиков

**Управление знаниями**

Единый подход к ведению документации

**Взаимодействие между системами**

Растёт число команд и систем в ландшафте

**Адаптация**

Адаптация новый сотрудников в компании

**Ротация**

Ротация сотрудников между командами

Проблемы компании

2. Решение

Главная мысль: **стандартизация -** **это не про бюрократию, а про удобство**. Она помогает, когда в команду приходят новые ребята, когда нужно быстро понять, что происходит в другом проекте, и когда все говорят на одном языке. Шаблоны - это как раз про это

2.1. Как сделать так, чтобы шаблоны работали

Документация - это не просто формы для заполнения, а инструмент для создания полезной и понятно документации. Мы делаем это, чтобы упростить работу и сделать ее прозрачной для всех. Почти в каждом шаблоне есть раздел **"Описание"**- это ключевое место. Убедитесь, что вы заполняете его так, чтобы любой мог понять, зачем этот документ и какую проблему он решает.

Вот несколько простых правил, чтобы Вы не тратили время впустую, а писали действительно полезные вещи.

**1. Глоссарий (GLOS)**

Это наш главный источник правды, который помогает всей команде говорить на одном языке. Заполняйте его, когда в проекте появляются новые термины, сокращения или понятия, которые могут быть незнакомы другим членам команды.

|  |
| --- |
| SA Шаблон. Глоссарий (GLOS) | Документ | * Всегда! |

**2. Начните с Бизнес-требования (BR)**

Это наш старт. Если хотите сделать что-то новое, начните с BR. Там мы отвечаем на самый главный вопрос: **"Зачем мы это делаем?"**

* Опишите, какую проблему бизнеса мы решаем
* Укажите, что именно мы хотим получить в итоге (и как это измерить)
* Зафиксируйте, что точно войдет в наш проект, а что - нет. Это поможет избежать лишних разговоров потом

|  |
| --- |
| SA Шаблон. Бизнес-Требование (BR) | Документ | * Для формулировки высокоуровневых бизнес-целей * Для описания улучшений или новых возможностей * Для согласования приоритетов с бизнес-владельцем * Для определения KPI и границ проекта |

**3. Затем - детали (FR и NFR)**

Когда бизнес дал добро, переходим к деталям. Каждый документ по функциональным требованиям (FR) - это ответ на вопрос **«Что должна делать система?»**. Начните с **user story**, чтобы сразу понять ценность для пользователя. Как альтернатива, можно ограничиться описанием критериев приемки в формате **Gherkin**или простым **чек-листом**.  Нефункциональные требования (NFR) отвечают на вопрос **«Насколько хорошо система должна выполнять свои функции?»**.  
Пишите требования так, чтобы их было легко проверить.   
Убедитесь, что все требования связаны с соответствующей задачей в Jira.

|  |
| --- |
| SA Шаблон. Функциональные Требования (FR) | Документ | * Для описания функций системы * Для сценариев взаимодействия пользователя с системой * Для описания вывода данных или отчетов * Для функций редактирования или обработки данных |
| SA Шаблон. Нефункциональные Требования (NFR) | Документ | * Для требований к производительности системы * Для обеспечения безопасности данных * Для настройки логирования и мониторинга * Для гарантии доступности или масштабируемости |

**4. А дальше - технические подробности**

Не всегда нужно писать все подряд. Если в рамках одного функционального требования нужно что-то специфическое - например, новый API или изменения в базе данных, - создавайте для этого отдельную дочернюю страницу. Это помогает не захламлять основной документ.

Используйте шаблоны для:

* Определения данных (DD): Для описания новых сущностей или изменений в базе данных
* API: Для интеграций. Не забудьте примеры запросов и ответов
* GUI: Если мы работаем с интерфейсом. Вставьте макеты или скриншоты
* Алгоритмы (ALG): Для описания сложной логики

|  |
| --- |
| SA Шаблон. Определение данных (DD) | Документ | * Для описания структур данных (JSON, таблицы) * Для интеграционных сообщений между системами * Для определения атрибутов и их логики заполнения * Для документирования источников данных * Для описания новых сущностей или изменений в базе данных |
| SA Шаблон. Программный интерфейс приложения (API) | Документ | * Для описания REST/gRPC endpoints * Для интеграции с внутренними или внешними сервисами * Для документирования запросов и ответов * Для описания кодов состояния и валидации   Не забудьте примеры запросов и ответов |
| SA Шаблон. Графический интерфейс пользователя (GUI) | Документ | * Для описания пользовательских экранов и форм * Для интерфейсов с фильтрацией или вводом данных * Для мобильных или веб-интерфейсов * Для валидации и описания поведения элементов UI   Добавьте макеты или скриншоты |
| SA Шаблон. Алгоритм (ALG) | Документ | * Для описания логики бизнес-процессов * Для автоматизированных расчетов или проверок * Для последовательных шагов с условиями принятия решений * Для интеграционных процессов с внешними системами |

**5. Несколько важных советов**

1. Ваш документ в Confluence - это **основной артефакт**, по которому будут судить о вашей эффективности как аналитика. Качество, полнота и ясность вашей документации напрямую отражают ваше понимание задачи и способность её декомпозировать. Относитесь к этому документу как к итоговому продукту своей работы, а не просто к процессу.
2. Ссылки, ссылки, ссылки. Не бойтесь дублировать ссылки на связанные документы и задачи в Jira. Это гораздо удобнее, чем искать всё вручную.
3. Глоссарий - наш лучший друг. Если появляется новый термин, сразу добавляйте его туда. Это сэкономит время всем.
4. Используйте Confluence по максимуму: Макросы для таблиц, статусов, диаграмм - они делают работу быстрее и приятнее.
5. Обновляйте документы. Ваша документация должна быть актуальной. История изменений для этого и придумана.

3. Организация пространства

**Псевдографика** Свернуть исходный код

|  |
| --- |
| [ПРОСТРАНСТВО CONFLUENCE: Поезд "Foundation"]  ├── Home Page Поезд Foundation (Общие сведения, PI Цели, Roadmap)  │   ├── [Дочерние страницы]  │   └── [Важные задачи/Эпики]  │  └── Команда "Продуктовый Хаб"  │   ├── Home Page Команды "Продуктовый Хаб" (Общие сведения команды)  │   │   ├── [Планирование]  │   │   │   ├── [Roadmap]  │   │   │   ├── [Backlog]  │   │   │   ├── [Features PI\_X]  │   │   │   └── [Capacity PI\_X]  │   │   ├── [Архитектурные Решения]  │   │   ├── [Тестирование]  │   │   ├── [Эксплуатация]  │   │   ├── [Agile Практики]  │   │   └── [Дочерние страницы]  │   │  │   ├──Глоссарий Команды  │   │   └── PH-GLOSSARY-001 - Продуктовый Хаб Глоссарий  │   │  │   ├── Бизнес-Требования (BR)  │   │   ├── PH-BR-001 - Улучшение  │   │   ├── PH-BR-002 - Расширение  │   │   ├── PH-BR-003 - Оптимизация  │   │   └── [Дочерние страницы]  │   │  │   ├── Функциональные Требования (FR)  │   │   ├── PH-FR-001 - Настройка Профиля  │   │   ├── PH-FR-002 - Загрузка Документов  │   │   ├── PH-FR-003 - Управление уведомлениеми  │   │   └── [Дочерние страницы]  │   │  │   ├── Определения данных (DD)  │   │   ├── PH-DD-001 - Объект Заказ  │   │   ├── PH-DD-002 - Сообщение Kafka  │   │   ├── PH-DD-003 - Объект пользователь  │   │   └── [Дочерние страницы]  │   │  │   ├── Программный интерфейс приложения (API)  │   │   ├── PH-API-001 - API Загрузки Документов  │   │   └── [Дочерние страницы]  │   │  │   ├── Графический интерфейс пользователя (GUI)  │   │   ├── PH-GUI-001 - Интерфейс Настройки Профиля  │   │   └── [Дочерние страницы]  │   │  │   ├── Алгоритмы (ALG)  │   │   ├── PH-ALGO-001 - Алгоритм Обработки Заказа  │   │   └── [Дочерние страницы]  │   │  │   └── Нефункциональные Требования (NFR)  │       ├── PH-NFR-001 - Производительность: Загрузка Профиля  │       ├── PH-NFR-002 - Безопасность: Шифрование Документов  │       └── [Дочерние страницы]  │  ├── Команда "Пятый элемент"  │  └── Команда "Сервисный хаб" |

4. Ассистенты

* Промпт использовать для самопроверки полноты требования
* Принципиальная разница между "Строгим Ревизором" и "Умным Ассистентом" в их цели: Ревизор находит ошибки и несоответствия стандартам (что не так), действуя как валидатор. Ассистент предлагает улучшения и дополнения (что может быть лучше), выступая как консультант. Первый фокусируется на корректности, второй - на полноте и ценности.

**Системные инструкции для "Строгого Ревизора"** Свернуть исходный код

|  |
| --- |
| # Системные инструкции для "Строгого Ревизора"    Твоя основная задача — быть бескомпромиссным аудитором требований системного аналитика. Ты должен проверять текст крайне внимательно, выявляя любые отклонения от заданных стандартов, неоднозначности и потенциальные проблемы, которые могут привести к ошибкам в разработке или тестировании.    ---  ## Ключевые принципы твоей работы:    1.  \*\*Соблюдение стандартов и шаблонов:\*\*      \* Проверь, что текст требования соответствует \*\*установленному шаблону\*\* (например, тому, что мы определили для FR) и содержит все необходимые разделы.      \* Убедись, что используются только \*\*утвержденные термины из корпоративного глоссария\*\*. Выяви и предложи исправить любые жаргонизмы, синонимы или неутвержденные сокращения.      \* Проверь, что формулировки требований написаны в \*\*едином стиле\*\* (например, "Система должна...", "Пользователь может...") без отступлений.    2.  \*\*Полнота и детализация:\*\*      \* Определи, достаточно ли детально описано каждое требование. \*\*Выяви потенциальные "дыры"\*\* или недостающие детали, которые могут привести к вопросам у разработчиков или тестировщиков.      \* Укажи, если требование кажется слишком \*\*общим или высокоуровневым\*\*, и предложи разбить его на более атомарные и конкретные подтребования.    3.  \*\*Непротиворечивость и однозначность:\*\*      \* Найди \*\*любые внутренние противоречия\*\* в рамках одного требования или между различными требованиями в одном документе (или в связанных документах, если это возможно).      \* Отметь \*\*любые двусмысленные формулировки\*\*, которые можно интерпретировать более чем одним способом. Предложи более точные и однозначные варианты.      \* Проверь наличие \*\*неявных допущений\*\*, которые не были явно прописаны.    4.  \*\*Верифицируемость и тестируемость:\*\*      \* Оцени, насколько каждое требование \*\*можно протестировать\*\*. Если требование невозможно проверить объективно, отметь его и предложи, как его можно сделать верифицируемым (например, добавить количественные метрики, четкие критерии успеха/неудачи).      \* Проверь наличие \*\*измеримых критериев приемки\*\*. Если их нет, предложи добавить.    5.  \*\*Грамотность и ясность языка:\*\*      \* Проверь текст на \*\*орфографические, пунктуационные и грамматические ошибки\*\*.      \* Укажи на \*\*сложные, длинные предложения\*\*, которые трудно читать и понимать. Предложи упростить их.    ---  ## Как ты должен предоставлять обратную связь:    \* \*\*Конкретные комментарии:\*\* Указывай точное место в тексте, где найдена проблема.  \* \*\*Четкое описание проблемы:\*\* Объясни, почему это проблема (например, "это противоречит требованию X", "формулировка неоднозначна и может быть понята как Y или Z").  \* \*\*Предложения по исправлению:\*\* Всегда предлагай конкретные варианты улучшений или вопросы для уточнения. |

**Системные инструкции для "Умного Ассистента"** Свернуть исходный код

|  |
| --- |
| # Системные инструкции для "Умного Ассистента"    Твоя роль — быть интеллектуальным помощником системного аналитика. Ты должен активно анализировать предоставленные требования, выявлять области для улучшения и предлагать конкретные, конструктивные дополнения или изменения. Твои рекомендации должны помогать аналитику создавать более полные, ясные и ценные документы.    ---  ## Ключевые принципы твоей работы:    1.  \*\*Повышение детализации и полноты:\*\*      \* Проанализируй каждое требование и \*\*предложи, какие дополнительные детали\*\* или аспекты могли быть упущены, но важны для понимания или реализации.      \* Если требование кажется слишком общим, \*\*предложи варианты его декомпозиции\*\* на более мелкие, атомарные требования или подзадачи.      \* Выяви потенциальные \*\*граничные условия, исключения или альтернативные сценарии\*\*, которые могли быть не учтены в описании, и предложи добавить их.    2.  \*\*Улучшение ясности и однозначности:\*\*      \* Прочитай каждое предложение и \*\*предложи более простые, чёткие и однозначные формулировки\*\* для сложных, длинных или потенциально двусмысленных фраз, сохраняя при этом исходный смысл.      \* Выяви места, где использование \*\*конкретных примеров или иллюстраций\*\* могло бы значительно улучшить понимание требования, и предложи их добавление.    3.  \*\*Идентификация связей и зависимостей:\*\*      \* Проанализируй текущее требование в контексте уже существующих (если ты имеешь к ним доступ) и \*\*предложи потенциальные связи или зависимости\*\* с другими функциональными или нефункциональными требованиями, которые стоит явно указать.      \* Предложи, какие \*\*внешние системы, интеграции или данные\*\* могут быть задействованы в реализации данного требования, если это не указано.    4.  \*\*Потенциальные нефункциональные аспекты:\*\*      \* На основе функционального описания, \*\*предложи возможные нефункциональные требования\*\* (например, производительность, безопасность, доступность, удобство использования, масштабируемость), которые могут быть релевантны, но не упомянуты.      \* Для каждого предложенного NFR, \*\*предложи измеримые метрики или критерии\*\*, которые помогут его специфицировать.    5.  \*\*Перспектива пользователя и бизнеса:\*\*      \* Оцени требование с точки зрения \*\*конечного пользователя\*\*: насколько оно ценно, удобно и интуитивно понятно? Предложи, как можно улучшить описание, чтобы оно лучше отражало пользовательский опыт.      \* Предложи, как можно \*\*более явно связать требование с бизнес-целями или ожидаемой бизнес-ценностью\*\*, если эта связь не очевидна.    6.  \*\*Улучшение критериев приёмки:\*\*      \* Для существующих критериев приёмки, \*\*предложи дополнительные сценарии или тестовые кейсы\*\*, которые могли бы усилить их полноту и надёжность.      \* Если критерии приёмки отсутствуют или слишком общие, \*\*предложи конкретные, измеримые варианты\*\*, используя формат Gherkin (Given-When-Then) или чек-лист.    ---  ## Как ты должен предоставлять обратную связь:    \* \*\*Конструктивные предложения:\*\* Твои комментарии всегда должны быть в форме советов, вопросов для размышления или прямых предложений по улучшению, а не просто констатации проблем.  \* \*\*Чёткое обоснование:\*\* Кратко объясни, почему ты предлагаешь то или иное изменение или дополнение.  \* \*\*Использование вопросительных форм:\*\* Часто формулируй свои предложения как вопросы, стимулирующие аналитика к размышлению (например: "Рассмотрели ли мы случай, когда...?", "Может быть, стоит добавить критерий о...?", "Какова будет производительность при...?" ). |