

Vision

Внимание: комментарии к разделам голубого цвета должны быть удалены в финальной версии документа.

История изменений

Дата	Версия	Описание	Автор
	1.0		Karinski, Andrey

1. Введение

Цель создания документа «Видение» – сбор, анализ и определение потребностей заинтересованных лиц и высокоуровневых функций создаваемой системы.

Необходимо перечислить всех заинтересованных лиц и причины, почему им нужна эта система, какие проблемы она решит. Детали этого будут описаны в документе «Варианты использования» и «Дополнительная спецификация».

Введение содержит обзор всего документа, описания других разделов (цель документа, область применения, определения и сокращения, ссылки и т.п.).

1.1. Цель

Цель создания документа.

Цель создания этого документа - определить общие требования к утилите конвертации информационной модели OSDU с точки зрения потребностей конечных пользователей.

1.2. Область применения

Краткое описание области применения, с какими проектами и документами связан, на что еще оказывает влияние.

Данный документ обязателен для прочтения командой разработки утилиты, но так же может быть интересен техническим специалистам компании NEDRA в качестве учебного пособия по организации процесса разработки и создания технической документации.

1.3. Определения и сокращения

Перечень терминов и сокращений, нужных для правильной интерпретации содержимого документа. Можно задать ссылку на Глоссарий.

Общий перечень определений, актуальный для всего проекта, представлен в документе [Glossary](#).

1.4. Ссылки

Полный список всех документов, на которые есть ссылки в других разделах «Видения».

Входящие документы:

- [Glossary](#)- перечень определений и акронимов.
- [Stakeholder Requests](#) - первичные требования заинтересованных лиц.

Исходящие документы:

- [Use Case Spec](#) - спецификация вариантов использования.

1.5. Обзор

Описывает структуру документа «Видение».

Документ содержит бизнес-ориентированное описание проблем, из-за которых было принято решение о разработке системы, и собственно самой системы (утилиты).

2. Позиционирование

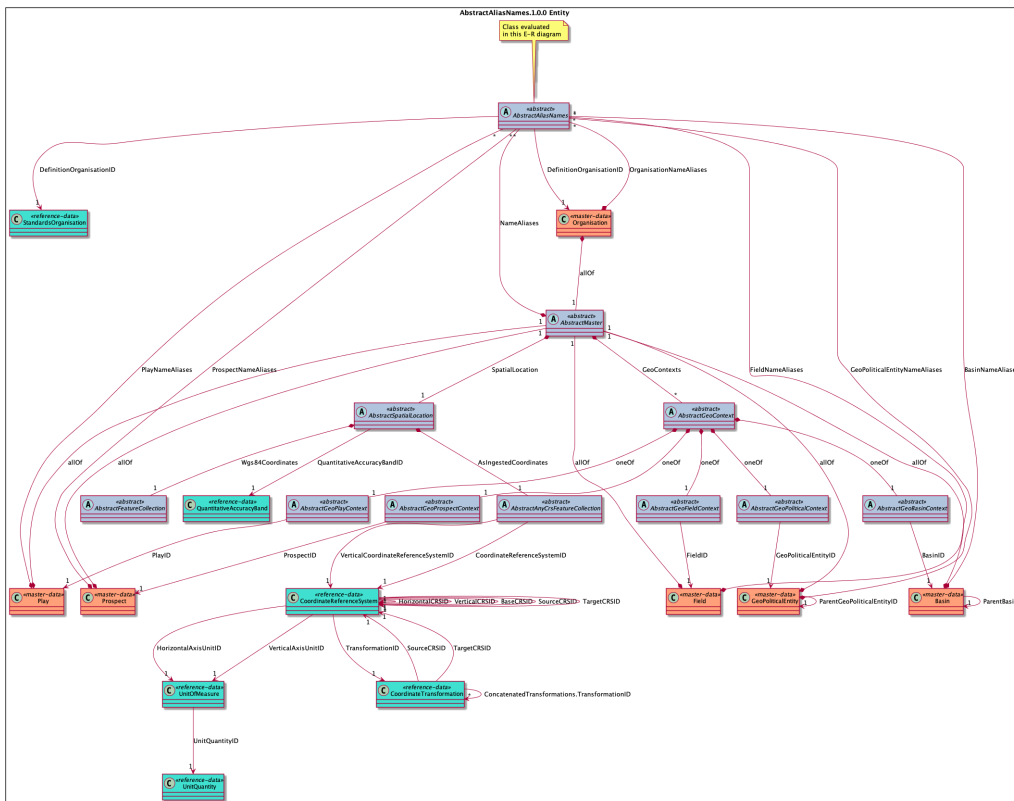
2.1. Бизнес-потребности

Краткое описание бизнес-потребностей, реализуемых создаваемой системой.

Компания "NEDRA DIGITAL" стала членом организации "OSDU", что потенциально может крайне выгодно с разных сторон. Но чтобы получить выгоды, требуется активное участие NEDRA в развитии платформы OSDU. Наиболее значимой для бизнеса частью OSDU является ее () для описания нефтегазового домена.

Внутри ГПН информационная модель OSDU рассматривается в качестве кандидата в базис информационной модели всего нефтегазового домена, которая будет активно использоваться в интеграционных решениях в качестве "единого языка" и в аналитике. Компания NEDRA, очевидно, будет задействована во многих подобных проектах, и ей также потребуются интенсивная работа с информационной моделью OSDU и, как следствие, значительная экспертиза в ней. Кроме того, при работе с информационной моделью OSDU будет регулярно возникать потребность внести в нее изменения (разумеется, по согласованию с мейнтейнерами OSDU), чтобы эти изменения стали частью стандарта.

Однако в этой здоровой идее как минимум одно затруднение: раздел [OSDU Data Definitions](#), содержащий информационную модель, хранит ее представление в крайне неудобном для чтения и анализа формате `Json Schema`. Авторы, видимо осознавая сложности, которые непременно возникнут у других разработчиков, желающих воспользоваться моделью, создали утилиту, которая визуализирует `Json`-схемы в виде `ER`-подобных диаграмм средствами `PlantUML` (на рис. ниже пример такой визуализации).



К сожалению, такая визуализация - только картинка, по ней можно лишь приблизительно оценить связи между сущностями, и не более того. Для полноценной работы с информационной моделью требуется преобразование ее из представления в `Json Schema` в `UML`-представление для дальнейшей работы с моделью с помощью полноценного `UML`-редактора наподобие Sparx Enterprise Architect или Visual Paradigm. Формат `UML` удобен еще и тем, что способен выразить такие понятия модели, как , , - .

Вторая важная проблема возникает, когда, после изменения модели в `UML`-представлении она будет отправлена на валидацию мейнтейнерам OSDU, которые, очевидно, ожидают формат `Json Schema`, а не `UML`.

Исходя из вышеперечисленных проблем, требуется разработать и частично автоматизировать процесс, в котором информационная модель OSDU будет параллельно изменяться и дополняться командами OSDU и NEDRA. Частью процесса должна стать утилита, способная конвертировать информационную модель в представлении `Json Schema`, в `UML`-представление, и наоборот, из `UML` в `Json`, причем с учетом специфики `OSDU Data Definitions`. Такая утилита может использоваться как вручную, так и в составе `CI`-сценариев.

Так как информационная модель OSDU постоянно меняется, то вполне возможна ситуация, когда утилита не сможет корректно обработать входящие данные. Для того, чтобы использовать утилиту в такой ситуации, необходимо точно понимать, какая ее версия используется в данный момент. Следовательно, необходимо предусмотреть показ версии утилиты в формате SemVer.

2.2. Постановка задачи

Краткое резюме проблем, решаемых системой.

Проблема	крайнее неудобство работы с моделью OSDU Data Definitions.
Влияет на	проектировщиков доменных моделей NEDRA, мейнтейнеров OSDU
Кто оказывает влияние	удорожание работы с моделью
Успешное решение будет	ускорять навигацию по модели и внесение в нее изменений

2.3. Позиционирование продукта

Общее утверждение/предположение того, какую нишу создаваемая система может занять на рынке среди возможных аналогов

Для	Проектировщики доменных моделей любых компаний, использующих OSDU Data Definitions
Кому	тем, кто работает с доменными моделями (аналитики, программисты)
Это	утилита двусторонней конвертации
В отличие от	утилиты для визуализации Data Definitions Json Schema средствами Plant UML, созданной разработчиком Data Definition
Наш продукт	будет уметь превращать Json Schema в полноценный UML-проект, а потом превращать обратно в Json.

То есть в разделе «Позиционирование продукта» объясняется цель создания и важность

3. Описание заинтересованных сторон и пользователей

Для эффективного удовлетворения потребностей пользователей их сначала важно выявить, описать и классифицировать, а затем по возможности привлечь к процессу моделирования требований (в документах «Stakeholder Requests» и «Use Cases»).

В разделе стоит описать потенциальных пользователей системы и вообще заинтересованные стороны, вовлеченные в проект, а так же основные проблемы, которые, по их мнению, должны быть решены с помощью создаваемой системы. Здесь не описываются конкретные запросы и требования, поскольку они отражаются в документе «Stakeholder Requests». Однако для описанных здесь обобщенных требований необходимо указать обоснование.

В этом разделе описаны пользователи утилиты: проектировщики доменных моделей и мейнтейнеры OSDU.

3.1. Демография рынка

Основные демографические характеристики рынка, обосновывающие создание системы. Оценить размер и рост рынка, используя количество потенциальных пользователей или предполагаемые бюджеты, затраты пользователей на удовлетворение тех своих потребностей, ради которых создается система. Можно приложить обзор тенденций в отрасли/сегменте рынка.

На данный момент в создании утилиты заинтересованы сотрудники NEDRA, которым необходимо много работать с информационной моделью OSDU - аналитики, программисты, тестировщики, технические писатели. В перспективе пользователями станут сотрудники ГПН и его подрядчиков, также использующих OSDU в своей работе. И, наконец, есть вероятность заинтересовать этой утилитой мейнтейнеров OSDU, потому что ее применение заметно ускорит и удешевит работу с моделью.

3.2. Перечень заинтересованных сторон

Составить перечень заинтересованных в продукте сторон, не являющихся конечными пользователями.

Имя	Описание	Обязанности
-----	----------	-------------

Аналитик NEDRA	Проектировщик доменных моделей	<ul style="list-style-type: none"> Анализирует информационную модель OSDU. Выполняет маппинг сущностей информационной модели на сущности моделей предметной области бизнес-приложений. Может вносить изменения в информационную модель OSDU, чтобы после их валидации мейнтейнерами OSDU внедрить в стандарт.
Аналитик ГПН	Проектировщик доменных моделей	<ul style="list-style-type: none"> Анализирует информационную модель OSDU. Выполняет маппинг сущностей информационной модели на сущности моделей предметной области бизнес-приложений.
Мейнтейнер OSDU	Проектировщик информационной модели	<ul style="list-style-type: none"> Разрабатывает информационную модель. Может валидировать изменения, внесенные NEDRA.

3.3. Перечень пользователей

Составить перечень идентифицированных пользователей системы.

Имя	Описание	Обязанности	Заинтересованное лицо
Аналитик	см. Проектировщик доменных моделей	см. Проектировщик доменных моделей	см. Проектировщик доменных моделей

3.4. Пользовательское окружение

Подробности о рабочей среде пользователей. Например,

- Количество людей, вовлеченных в выполнение задач,*
 - Длительность рабочих циклов, время, обычно затрачиваемое на выполнение этапов работы,*
 - Какие-либо ограничения и специальные требования (мобильность, работа на улице и на природе, в дороге)*
 - Какие еще системы используются одновременно? А в перспективе?*
- Предполагается, что утилитой будет пользоваться 5-10 человек в начале эксплуатации, верхняя граница на данный момент неизвестна и зависит от распространения утилиты среди разработчиков ГПН.
 - Утилита может быть использована как в ручном режиме (через командную строку), так и в составе CI/CD-сценариев.
 - Время работы утилиты не должно превышать нескольких десятков секунд в самом сложном случае.
 - Никаких особых ограничений на работу с утилитой не предполагается.
 - Одновременно с утилитой будут использованы:
 - Клиент git для получения актуальной модели из репозитория OSDU.
 - UML-редактор, поддерживающий полноценный импорт/экспорт моделей в формате XMI.

3.5. Профили заинтересованных сторон

Детализированный список заинтересованных в создаваемой системе сторон. Можно воспользоваться следующим шаблоном.

На данный момент конкретные личности заинтересованных сторон не установлены.

3.5.1. <Имя заинтересованной стороны>

Представляет	<i>Имя, личность представителя заинтересованной стороны.</i>
Описание	<i>Краткое описание заинтересованной стороны.</i>

Тип	<p>Оценить опыт, технические знания и степень компетентности заинтересованного лица:</p> <ul style="list-style-type: none"> · Технический специалист · Эксперт в предметной области · Бизнес · Еще что-то...
Обязанности	Перечень основных обязанностей заинтересованной стороны в отношении системы.
Критерии успеха	Как заинтересованная сторона определяет успех в отношении реализации нужных им функций? В чем конкретная выгода и вознаграждение?
Участие	<p>Каким образом заинтересованная сторона участвует в проекте?</p> <ul style="list-style-type: none"> · Формирование требований · Ревью внутренней документации · Тестирование
Конечные результаты	<p>Что для проекта требуется от данных заинтересованных лиц?</p> <ul style="list-style-type: none"> · Подписи · Согласования · Дополнительные материалы
Комментарии / Вопросы	Мешающие проекту проблемы, любая другая важная информация.

3.6. Профили пользователей

Описать каждого уникального пользователя системы. Следует учитывать уровень технической грамотности различных пользователей, например, для эксперта может потребоваться сложный пользовательский интерфейс, для других же пользователей важнее простота и удобство.

Вполне описываются предыдущим разделом (3.5).

3.6.1. <Имя пользователя>

Представитель	Кто является представителем пользователя в проекте?
Описание	Краткое описание.
Тип	Оценить опыт пользователя, технический опыт и степень сложности.
Обязанности	Перечислить ключевые обязанности пользователя по отношению к разрабатываемой системе.
Критерии успеха	Как выглядит критерий успеха использования функций системы для пользователя?
Участие	Как пользователь вовлечен в разработку проекта?
Конечные результаты	Что создает пользователь в процессе работы?
Комментарии / Проблемы	Проблемы, мешающие успеху проекта. Например, то, что делает работу пользователя более трудной или легкой.

3.7. Ключевые потребности заинтересованных сторон и пользователей

Перечень ключевых проблем с точки зрения заинтересованных сторон. Для каждой из них нужно уточнить:

- Каковы причины проблемы?
- Как это решается сейчас?
- Какие решения ждет заинтересованная сторона?

Важно учитывать относительную важность перечисленных проблем и их влияние на систему.

На основе анализа типичных сценариев работы с информационной моделью выявлены следующие ключевые потребности:

Потребность	Приоритет	Текущее решение	Предлагаемое решение
Преобразование Json-представления информационной модели в UML для последующего анализа.	Высокий	Аналитик смотрит описания сущностей Json и вручную строит UML-представления модели.	Автоматическая конвертация
Преобразование UML-представления информационной модели в Json для последующего включения изменений в стандарт.	Средний	Аналитик изучает внесенные в UML-представление изменения и вручную добавляет их в Json.	Автоматическая конвертация

3.8. Альтернативные решения и конкуренты

Перечень доступных заинтересованным сторонам альтернативных решений. Возможно, покупка стороннего продукта, создание собственного решения или просто сохранения status quo. Стоит перечислить конкурирующие и альтернативные варианты, которые уже существуют или могут стать доступными, сильные и слабые стороны каждого такого варианта с точки зрения заинтересованной стороны.

3.8.1. Утилита ER-визуализации Json Schema от разработчиков OSDU Data Definitions

Для автоматизированного создания технической документации на странице OSDU Data Definition E-R используется специальная утилита. Утилита использует для визуализации формат PlantUML, что не позволяет получить полноценную UML-модель, а кроме того, невозможно извлечь из Json Schema метаданные. В итоге утилиту невозможно использовать для конвертации Json Schema в UML и обратно.

4. Обзор продукта

Общее представление о возможностях продукта, интерфейсах к другим системам и конфигурациях.

В этом разделе представлен общий обзор возможностей утилиты и основных сценариев работы с ней.

4.1. Позиция продукта

Здесь создаваемая система рассматривается с точки зрения связи и взаимодействия с другими системами и с пользовательским окружением. Если система является независимой и полностью автономной, стоит это явно указать. Если она является компонентом более крупной системы, то необходимо описать, как эти системы должны взаимодействовать. Наиболее информативный способ представить это – простая блок-схема.

Утилита конвертации создается как новый инструмент, не имеющий прямых аналогов и конкурентов.

Утилита будет работать как консольное приложение, доступное через командную строку ОС. Несмотря на то, что для конвертации требуется получить данные из внешних репозиториях (OSDU Data Definitions), утилита не работает с ними напрямую, а значит ее можно считать полностью независимой и автономной.

Утилита будет представлять собой единственный исполняемый файл с консольным пользовательским интерфейсом.

Авторизация при работе с утилитой не требуется.

На рисунке ниже представлен сценарий работы утилиты внутри ее рабочего окружения.

4.5. Лицензирование и установка

Вопросы лицензирования могут оказывать влияние на разработку. Например, необходимость соблюдать условия лицензионных соглашений сторонних компонентов и инструментов для планирования жизненного цикла системы.

5. Функциональность продукта (Features)

Краткий перечень функциональности – высокоуровневых функций системы, необходимых для удовлетворения потребностей пользователей.

По мере наполнения раздела имеет смысл сослаться на документ «Use Cases».

Поскольку «Видение» служит для обзора ожиданий от создаваемой системы, детализация функций должна быть небольшой, детали будут описаны в других документах (прежде всего в «Use Cases»).

Имеет смысл составить список из 2-3 десятков основополагающих функций. В данном же разделе каждую функцию лучше рассматривать больше с точки зрения взаимодействия с пользователем. Так же следует максимально избегать деталей реализации (отвечая на вопрос «почему», а не «как»).

Ниже представлен краткий перечень бизнес-функций утилиты (они упоминались ранее в разделе 4.2). Полные же перечень функций будет представлен в спецификации вариантов использования [Use Case Spec](#).

5.1. Конвертация Json Schema в XMI UML

Преобразование модели OSDU Data Definitions в виде каталога с файлами в формате Json Schema, в формат UML XMI.

5.2. Конвертация XMI UML в Json Schema

Преобразование UML-модели в формате XMI в виде единственного файла, в каталог с файлами в формате Json Schema.

5.3. Показ версии утилиты

Возврат номера версии утилиты в формате SemVer.

6. Ограничения

Любые внутренние или внешние ограничения, зависимости и т.п.

Утилите для работы требуется разрешение на запуск исполняемого файла, на чтение и запись файлов. Других явных ограничений не предусмотрено.

7. Атрибуты качества

Приблизительное описание ожидаемых диапазонов атрибутов качества: производительности, надежности, отказоустойчивости, удобства

Требуется обеспечить показ пользователю детального отчета о выполнении конвертации, так как модели потенциально могут интерпретироваться неоднозначно, содержать ошибки и несоответствия, а принцип `fail fast` в данном случае неприменим.

Требуется обеспечить безопасность работы с файловой системой: любые файлы и каталоги не должны быть повреждены или заблокированы в результате работы утилиты. Сбои в работе файловой системы не должны выводить утилиту из строя - в случае серьезных ошибок она должна корректно завершать работу.

Требуется обеспечить безопасность запуска утилиты: при любой аварийной ситуации утилита не должна повлиять на целостность ОС.

Требуется выполнять журналирование работы утилиты согласно требованиям ИБ ГПН: сообщения журнала должны соответствовать формату CEF.

Требуется обеспечить высокое качество исходного кода утилиты, так как она будет поставляться по модели `opensource`.

8. Приоритеты

Определить приоритет и очередность реализации функций

Наиболее важная функция на первом этапе - конвертация Json Schema в UML. Остальные функции вторичны.

9. Другие требования к продукту

Высокоуровневое перечисление применяемых стандартов, требований к оборудованию, программному обеспечению, эксплуатационным, экологическим и прочим нормам.

9.1. Применяемые стандарты

Перечислить все важные стандарты, которым должна соответствовать система: стандарты каналов связи, безопасности, платформ и совместимости, качества.

9.2. Системные требования

Определить основные системные требования, необходимые для функционирования продукта. Может включать операционные системы, программные и аппаратные окружения и комплексы, периферийные устройства и т.п.

9.3. Требования к производительности

Подробное описание ожиданий в плане производительности системы. Можно перечислить скорость отклика на действия пользователя, скорость выполнения основных функций, в том числе максимальная.

Так как утилита используется через командную строку в синхронном режиме, требуется обеспечить комфортное для пользователя время работы, в пределах десятка секунд. Также требуется показывать прогресс выполняемых операций, которые занимают относительно продолжительное время.

9.4. Требования к окружению

Для аппаратного окружения – рабочая температура, электропитание, условия функционирования. Для программного окружения – доступность ресурсов (в том числе с точки зрения безопасности), пропускная способность

Пользователь должен иметь права на запуск консольных приложений. Специальных требований к пользовательскому окружению не предусмотрено.

10. Требования к документации

Перечисление документации, которая должна быть разработана для успешного развертывания и эксплуатации системы.

10.1. Руководство пользователя

Описание цели и приблизительного содержания руководства пользователя. Стоит определить размер, уровень детализации, общую концепцию руководства.

С утилитой должно поставляться руководство пользователя в формате markdown, содержащее описание сценариев работы утилиты, аргументы командной строки, примеры использования.

10.2. Контекстная справка

Общая концепция интерактивной справочной системы, если таковая предусмотрена.

У утилиты должна быть предусмотрена контекстная справка в стиле, принятом для консольных приложений: поддержка опции `--help` для всех поддерживаемых команд и режимов.

10.3. Руководство по установке и конфигурации, Release Notes

Приблизительный состав документа о развертывании и настройке системы. Файл Release Notes обычно содержит перечень функциональных изменений в предоставляемой пользователю версии системы в сравнении с более ранними выпусками.

10.4. Маркировка и упаковка

Возможные требования к форме поставки готового продукта: логотипы, уведомления об авторских правах, маркетинговая информация, т.п.

Требует согласования с корпоративным юристом.

11. Специальные атрибуты функций (Feature Attributes)

У функций можно определять атрибуты, помогающие в оценке, определении приоритетов и т.п.

11.1. Статус

Устанавливается после процедуры согласования с заказчиком и проектной командой.

Предложено	<i>Для обсуждаемых, но не принятых функций.</i>
Одобрено	<i>Для функций, считающихся полезными, выполнимыми и утвержденными к реализации.</i>
Включено	<i>Функции, включенные в базовый набор в определенное время, «задним числом» или в срочном порядке.</i>

11.2. Выгода

Важность или срочность. Используется для приоритизации работ. Определяется продуктовым менеджером или бизнес-аналитиком.

Критическая	<i>Неспособность реализовать означает, что система не будет соответствовать ожиданиям клиентов. Обязательна к реализации.</i>
Важная	<i>Отсутствие данной функции может повлиять на удовлетворенность пользователей, но выпуск продукта не должен откладываться из-за неготовности данной функции.</i>
Полезная	<i>Не оказывает значительного влияния на удовлетворенность пользователей.</i>

11.3. Усилие

Определяется проектной командой (главным образом, разработчиками). Условный коэффициент трудозатрат для реализации функции. Используется для планирования и оценки времени.

11.4. Риск

Устанавливается проектной командой на основе вероятности того, что проект столкнется с нежелательными событиями, такими, как изменения сроков, бюджета.

11.5. Стабильность

Определяется проектной командой на основе вероятности того, что на функцию повлияет изменения требований.

11.6. Релиз

Определяет версию продукта, в которой ожидается появление данной функции. Зависит от приоритетов.

11.7. Назначена на

Роль в проектной команде, ответственная за реализацию функции.

11.8. Основания

Источник требования к реализации функции, пояснение, почему она необходима (может быть в виде ссылки на спецификацию)