

Кейс 3. Автоматизация компании «ОМЕГА»

Компания Омега занимается разработкой сайтов. Для работы с заказчиками используется собственная разработка НГТР (Нео Групп Главная Разработка). В процессе работы потенциальные заказчики могут:

- регистрировать заявки на сайте через веб форму;
- регистрировать обращения через телефонный звонок (форма заполняется оператором).

Входные поля, которые заполняются: **Контактное лицо, Телефонный номер, Текст сообщения, Обращение по телефону (да/нет).**

В процессе работы, в проработку запускаются только те заявки, которые согласует директор. Согласование идет путем подтверждения в веб форме. Заявкам, зарегистрированным через телефонный звонок, директор уделяет большее внимание, считая, что они более важны для компании. В итоге рассмотрение телефонных заявок проходит в среднем за 4-5 рабочих дней, а заявок поданных через веб форму – 15-20 дней.

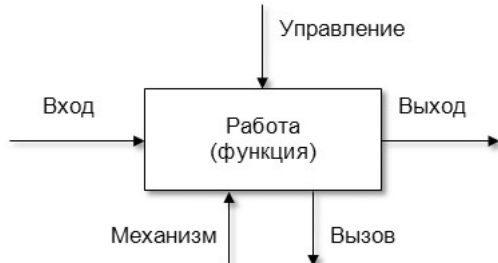
После согласования директора, менеджер начинает обзванивать потенциальных клиентов и выяснять требования. В результате, в 90% случаев, заявки поданные через веб-форму являются не актуальными и клиенты уходят к конкурентам. Менеджер по телефону уточняет дополнительную информацию: бюджет проекта, количество страниц, готовность контента для сайта, желаемое время для встречи, сроки реализации проекта.

После сбора обозначенной информации формируется рабочая команда, планируются работы, начинается выполнение. Заказчику предоставляется конечная дата реализации, без каких-либо инструментов просмотра состояния заказа. В процессе выполнения работ, заказчики жалуются на отсутствие информации о продвижении проекта, часто это приводит к необходимости корректировки результата на 25-40%, так как заказчик «по другому представлял результат».

Задание:

1. Опишите/отобразите текущий и целевой бизнес процесс регистрации заявки (желательно изобразить).
2. Назовите ключевые проблемы, которые есть в текущей схеме построения работы. При ответе опишите почему вы считаете это проблемой.
3. Составьте требования к системе, направленные на решение ключевых проблем.
4. Разработать контекстную диаграмму системы и выполнить ее декомпозицию (первого уровня) с использованием нотации IDEF0.

5. Разработать диаграмму прецедентов.



1. Текущий и целевой бизнес-процесс регистрации заявки

Текущий бизнес-процесс регистрации заявки:

1. Потенциальный клиент подает заявку:

Через веб-форму на сайте.

Через телефонный звонок (оператор заполняет форму).

2. Заявка поступает директору для согласования:

Директор уделяет больше внимания заявкам через телефон.

Сроки согласования:

4-5 рабочих дней для телефонных заявок.

15-20 дней для веб-заявок.

3. После согласования:

Менеджер связывается с клиентом, уточняет требования (бюджет, количество страниц, контент, сроки и т.д.).

4. Формируется рабочая команда и начинается выполнение проекта.

5. Заказчик получает конечный результат без возможности отслеживания прогресса.

6. Часто возникают доработки (25-40% от изначального объема) из-за недопонимания ожиданий заказчика.

Целевой бизнес-процесс регистрации заявки:

1. Потенциальный клиент подает заявку:

Через веб-форму на сайте.

Через телефонный звонок (оператор заполняет форму).

2. Заявка автоматически попадает в систему управления заявками.

3. Автоматическое уведомление директора о новой заявке:

Все заявки рассматриваются в равные сроки (например, 1-2 рабочих дня).

4. Менеджер связывается с клиентом:

Используется чек-лист для уточнения всех необходимых данных.

5. Вся информация о проекте фиксируется в системе.

6. Заказчику предоставляется личный кабинет для отслеживания статуса проекта и предоставления обратной связи.

7. В процессе выполнения проекта заказчик может вносить правки через систему (в рамках согласованного плана).

8. Финальный результат предоставляется с минимальными корректировками.

2. Ключевые проблемы текущей схемы работы

1. Долгий процесс согласования заявок:

Телефонные заявки обрабатываются быстрее, чем веб-заявки (15-20 дней против 4-5 дней).

Это приводит к потере клиентов, особенно тех, кто подал заявку через веб-форму.

2. Высокий процент неактуальных заявок (90%) через веб-форму:

Отсутствие фильтрации и предварительной оценки заявок до их обработки менеджером.

Большое количество времени тратится на обработку неактуальных заявок.

3. Отсутствие прозрачности для заказчиков:

Клиенты не могут отслеживать статус выполнения своих проектов.

Это вызывает недовольство и дополнительные корректировки результата (25-40%).

4. Недостаточная детализация требований на этапе сбора информации:

Менеджеры не всегда собирают полную информацию о проекте.

Это приводит к недопониманию между заказчиком и исполнителем.

3. Требования к системе

Для решения ключевых проблем необходимо разработать систему с учетом следующих требований:

Функциональные требования:

1. Автоматизация регистрации заявок:

Единая база данных для всех заявок (веб-форма и телефонные звонки).

Автоматическое уведомление директора о новых заявках.

2. Оптимизация процесса согласования:

Внедрение приоритетной системы оценки заявок (например, по срочности или бюджету).

Уведомления о сроках согласования.

3. Фильтрация неактуальных заявок:

Добавление обязательных полей в веб-форму (например, бюджет, сроки, готовность контента).

Автоматический анализ данных для определения актуальности заявки.

4. Прозрачность для заказчиков:

Личный кабинет для клиентов с возможностью отслеживания статуса проекта.

Уведомления о ключевых этапах выполнения работ.

5. Сбор полной информации о проекте:

Использование чек-листа для менеджеров при общении с клиентами.

Хранение всей информации о проекте в системе.

6. Обратная связь от заказчиков:

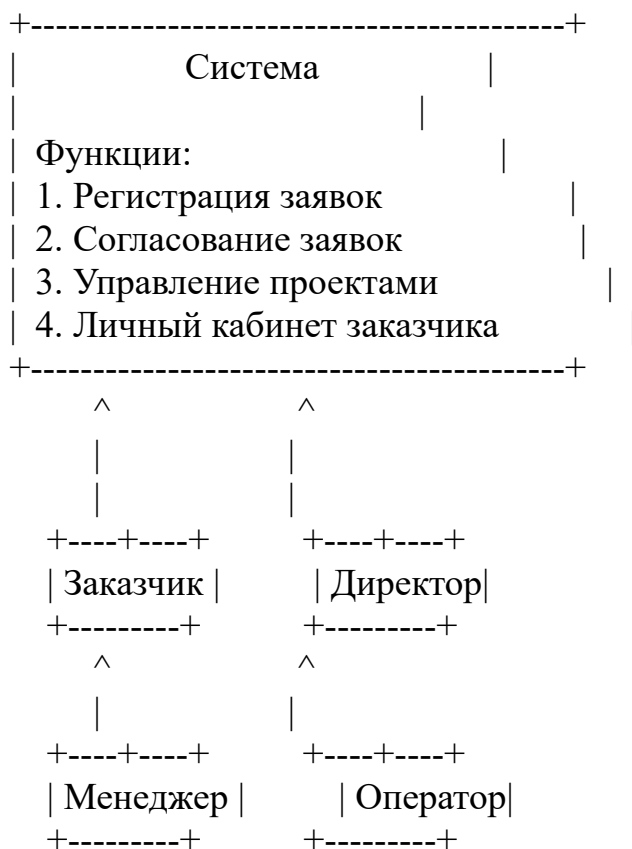
Возможность заказчика вносить комментарии и правки в рамках согласованного плана.

Нефункциональные требования:

1. Удобный интерфейс для сотрудников компании и клиентов.
2. Высокая производительность системы (обработка заявок не более 1 рабочего дня).
3. Интеграция с CRM-системой для хранения данных о клиентах.

4. Контекстная диаграмма системы и ее декомпозиция

Контекстная диаграмма системы



Декомпозиция первого уровня (IDEF0)

A0: Управление заявками

Входы:

Заявки от клиентов (веб-форма, телефон).

Выходы:

Согласованные заявки.

Отчеты о статусе выполнения проектов.

Механизмы:

Система автоматизации управления заявками.

Менеджеры, операторы.

Управление:

Регламент работы с клиентами.

Политика компании.

A1: Регистрация заявок

Входы:

Данные от клиентов (контактное лицо, телефон, текст сообщения и т.д.).

Выходы:

Зарегистрированные заявки.

A2: Согласование заявок

Входы:

Зарегистрированные заявки.

Выходы:

Согласованные заявки.

A3: Выполнение проекта

Входы:

Согласованные заявки.

Собранная информация о проекте.

Выходы:

Готовый результат проекта.

A4: Обратная связь

Входы:

Комментарии клиента.

Выходы:

Корректировки проекта (при необходимости).

Диаграмма, представленная на изображении, отображает модель IDEF0, которая используется для описания и анализа систем с точки зрения функций и процессов. Здесь каждая функция представлена в виде блока, у которого имеется несколько интерфейсов:

1. Вход (Input): Сюда поступают данные или ресурсы, которые необходимы для выполнения функции.
2. Выход (Output): Это результат работы функции.
3. Управление (Control): Условия или ограничения, которые влияют на выполнение функции.
4. Механизм (Mechanism): Ресурсы, которые обеспечивают выполнение функции, такие как инструменты, персонал и т.д.
5. Вызов (Call): Иногда используется для обозначения взаимодействия между функциями.

Для создания диаграммы прецедентов в UML необходимо использовать другой подход. Диаграмма прецедентов представляет взаимодействия между системой и внешними агентами (актерами) в контексте сценариев использования. Она включает в себя:

Акторы: Показывают, кто взаимодействует с системой (люди, другие системы и т.д.).

Прецеденты использования (Use Cases): Действия или процессы, которые система должна выполнить в ответ на запросы акторов.

Связи: Показаны линиями между акторами и прецедентами использования.

Чтобы создать диаграмму прецедентов на основе вашей системы, нужно определить список акторов и сценариев, которые систему будут использовать, и отобразить их связи в виде схемы.
