**От понимания требований к реализации системы**

В проект было решено внедрить объектно-ориентированные методы. Будет описано небольшое число механизмов, выполняющих ключевые требования системы. В результате будет получено множество классов, взаимодействие которых приводит к более содержательному поведению, чем просто сумма поведений отдельных классов.

**6.1.1. Объектно-ориентированная модель системы**

Так как данная система имеет высокий уровень сложности, было решено представить ее в виде абстрактной упрощенной модели, отбросив незначительные детали, чтобы получить более понятную версию. В объектно-ориентированной модели при представлении данных имеется возможность идентифицировать отдельные записи базы. Между записями базы данных и функциями их обработки устанавливаются взаимосвязи с помощью механизмов, подобных соответствующим средствам в объектно-ориентированных языках программирования.

**6.1.1.1. Архитектура систем ПО**

**6.1.1.1.1. Понятие архитектуры ПО**

В рамках данной системы можно ввести следующее понятие архитектуры ПО. Программная архитектура представляет собой описание элементов, из которых конструируется система, взаимодействий между ними, а также образцов, согласно которым они составляются, и ограничений, налагаемых на эти образцы.

**6.1.1.1.1.1. Понять, ЧТО делает система**

* Система облегчает общение курьерской службы с клиентом, с помощью личного кабинета последнего, в котором клиент сможет оформлять заказы, отслеживать посылки и оценивать качество работы курьера.
* Система предоставляет возможность клиенту просматривать оформленные заказы, а также определить статус заявки в реальном времени.
* Благодаря системе клиент имеет возможность оплачивать услуги курьерской службы онлайн.
* Система помогает диспетчеру равномерно распределить нагрузку между курьерами.
* Система обеспечивает GPS-навигацию курьеров для контроля их отклонения от заданного маршрута.
* Система формирует оптимальный маршрут для курьера с учетом последовательности пунктов назначения.
* Система осуществляет уведомление курьера о его доставках.
* Система расставляет заказы по приоритетам, чтобы удовлетворить VIP-клиентов.
* Система облегчает общение диспетчера с курьером, с помощью личного кабинета последнего, в котором курьер сможет видеть оптимальные маршруты и план доставок на день.
* Система формирует рейтинговые списки курьеров.
* Система формирует базу учета ремонтов и проведенных ТО.
* Система создает журнал путевых листов.
* Система обеспечивает конфиденциальность информации для различных уровней пользователей.
* Система формирует персональные скидки для постоянных клиентов.
* Система ведет базу сотрудников, транспорта.
* Система ведет складской учет.
* Система автоматизирует процесс паркинга.
* Система формирует финансовые отчеты и статистику.
* Система обеспечивает удобную навигацию пользователя по парковкам.
* Система обеспечивает возможность оплаты с помощью различных существующих платежных систем.

**6.1.1.1.1.2. Понять, КАК она работает**

* Система считывает оформленный заказ и отправляет его далее на обработку.
* Система строит маршрут до выбранной парковки по запросу клиента.
* Система обеспечивает доступ клиента к парковке.
* Система списывает средства с личного счёта клиента во время пребывания его авто на парковке.
* Система даёт возможность покинуть парковку.
* Система опрашивает пользователя на предмет качества работы парковок и формирует оценку.
* Система анализирует базы данных обо всех парковках, подсчитывает их эффективность, создаёт отчёты и ведёт статистику, предоставляя её администрации по запросу.
* Система даёт возможность владельцам территории установить парковку и тарифы на ней, а также проводить акции.

**6.1.1.1.1.3. Иметь возможность обдумывать и разрабатывать ЧАСТИ системы**

Проанализировав функциональность системы, было решено выделить следующие ее части:

1. Личный кабинет клиента. В нем клиент может найти парковки поблизости, построить к ним маршрут, просмотреть рейтинг и оформить бронь. Также есть доступ к личной информации клиента. Возможность получить обратную связь и просмотреть историю своих парковок. Возможность управления личным счётом.
2. Система выбора доступных парковочных мест. Пользователь взаимодействует с ней, когда хочет найти парковочные места, построить маршрут или оформить бронь.
3. ПО для пунктов контроля обеспечивает автоматизацию процесса парковки и включает в себя систему оповещения технической поддержи о возникших проблемах, которые необходимо решить.
4. Платёжная система обеспечивает быстрое проведение платежей и предоставляет удобные пользователю способы пополнения счёта.
5. Консалтинговый центр обеспечивает поддержку клиентов и передаёт информацию о возникших проблемах команды техническому персоналу.
6. Система учёта парковок обеспечивает мониторинг технического состояния и занятости парковок в режиме реального времени.
7. Работа с клиентами подразумевает гибкую систему настройки тарифов, акций и персональных бонусов.
8. Система управления предприятием обеспечивает формирование автоматической отчётности и статистики по парковкам.

**6.1.1.1.1.4. РАСШИРЯТЬ систему**

После внедрения системы и ее успешного функционирования переходим на этап расширения. На данном этапе предполагается улучшения системы и внесения нового функционала. После выпуска системы версии 1.0 в новой системе версии 1.1 в случае отсутствия будет добавлен или улучшен следующий функционал:

* Гибко настраиваемые пользовательские уведомления
* Система персональных скидок (бонусная программа)
* Расширение существующего набора поддерживаемых платёжных систем

**6.1.1.1.1.5. ПОВТОРНО ИСПОЛЬЗОВАТЬ часть (части) системы при создании новых систем**

Для данной системы будет реализована метрология повторного использования. То есть, система частично будет составляться из частей, написанных ранее компонентов. Это позволит сократить трудозатраты при разработке. Кроме того, так как данная система будет многомодульной, то в дальнейшем при необходимости ее компоненты могут быть использованы при создании нового программного продукта.

**6.1.1.1.2. Определение архитектуры ПО**

Архитектурные модели нужны различным группам заинтересованных лиц, которые будут рассматривать предложенную архитектуру с разных точек зрения. Поэтому было решено ввести 4+1 представление архитектуры.

**6.1.1.1.2.1. 4+1 представление архитектуры ПО**

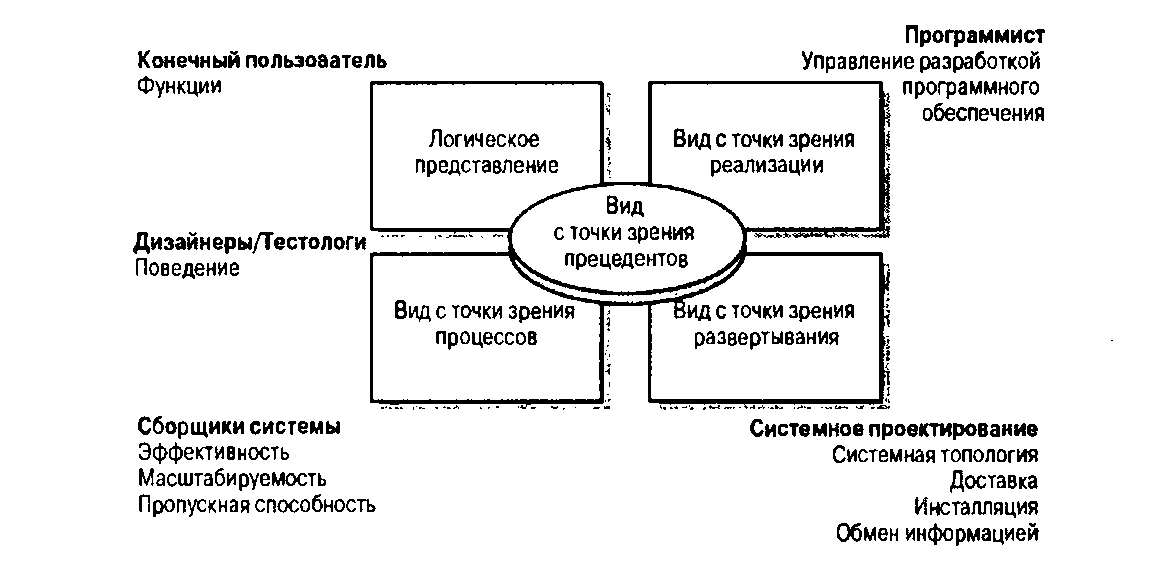
4+1 представление архитектуры лучше всего иллюстрирует обобщенные потребности системы. Эти представления изображены на рисунке. Отметим, что различные заинтересованные лица (программисты, руководство, пользователи) расположены возле тех представлений, которые они обычно используют.

Рис. 4+1 представление архитектуры ПО