

ОСОБЕННОСТИ СОЗДАНИЯ ПРОГРАММНОГО ПРОДУКТА

ВЫПОЛНИЛ ДОНСКОЙ ДМИТРИЙ

ГРУППА И7-15

1. ДОСТИЖЕНИЕ СОГЛАШЕНИЯ ОБ ОПРЕДЕЛЕНИИ ПРОБЛЕМЫ

Первый шаг состоит в достижении соглашения об определении проблемы, которую необходимо решить. Один из простейших способов заключается в том, чтобы просто записать проблему и выяснить, все ли согласны с такой постановкой.

В рамках этого процесса зачастую полезно рассмотреть преимущества предлагаемого решения, причем их следует описывать на языке клиентов/пользователей. Это обеспечивает дополнительную содержательную основу для понимания реальной проблемы. Рассматривая эти преимущества с точки зрения клиента, программисты также достигают лучшего понимания их взгляда на проблему в целом.

Часто бывает полезно записать проблему в стандартной форме (табл. 2.1). Создание подобной таблицы является простым, но действенным средством, чтобы удостовериться в том, что все участники проекта работают вместе над осуществлением общей цели.

Таблица 2.1. Структурирование проблемы

Элемент	Описание
Проблема	Описание проблемы
Воздействует на что (кого) и результатом чего является	Указание лиц, на которых оказывает влияние данная проблема. Описание воздействия данной проблемы на заинтересованных лиц и бизнес-деятельность
Выигрыш от решения может состоять в следующем	Указание предлагаемого решения. Список основных предоставляемых решением преимуществ

2. ВЫЯВЛЕНИЕ ОСНОВНЫХ ПРИЧИН — ВОПРОСОВ, СТОЯЩИХ ЗА ПРОБЛЕМОЙ.

На данном этапе важно понять корневые причины, лежащие в основе проблемы, и ее проявления. Например, электронный магазин решил бороться с проблемой недостаточной прибыльности. Для этого был проведен анализ причин плохих продаж. Получено, что следующие причины ведут к слишком большим остаткам продукции на складе:

- 1) устаревшие готовые изделия;
- 2) неправильные заказы на покупку;
- 3) повреждения при доставке;
- 4) производственные дефекты;
- 5) возвраты клиентами;
- 6) прочее.

Однако нужно ли устранять все эти причины? Зачастую нет. Некоторые корневые причины просто не стоят того, чтобы их устранять. Нужно определить влияние каждой корневой причины и устранять только те, которые наиболее серьезно влияют на саму проблему. В примере, допустим, наибольшее влияние оказывает корневая причина «Неправильные заказы на покупку».

3. ВЫЯВЛЕНИЕ ЗАИНТЕРЕСОВАННЫХ ЛИЦ И ПОЛЬЗОВАТЕЛЕЙ.

В этом процессе могут помочь ответы на следующие вопросы:

- Кто является пользователем системы?
- Кто является заказчиком (экономическим покупателем) системы?
- На кого еще окажут влияние результаты работы системы?
- Кто будет оценивать и принимать систему, когда она будет представлена и развернута?
- Существуют ли другие внешние или внутренние пользователи системы, чьи потребности следует учесть?
- Кто будет заниматься сопровождением новой системы?
- Не забыли ли мы кого-нибудь?

4. ОПРЕДЕЛЕНИЕ ГРАНИЦ СИСТЕМЫ.

Мир делится на две части (рис. 2.2):

- создаваемая система;
- то, что взаимодействует с системой, — фактор.



Рис. 2.2. Границы системы

Очень важно правильно определить факторы. Для этого следует ответить на приводимые ниже вопросы.

- Кто будет управлять системой?
- Кто будет осуществлять сопровождение системы?
- Откуда система получает информацию?
- Какие внешние системы будут взаимодействовать с системой?

5. ВЫЯВЛЕНИЕ ОГРАНИЧЕНИЙ, НАЛАГАЕМЫХ НА РЕШЕНИЕ.

Ограничения уменьшают степень свободы, которой располагают разработчики при реализации решения. Каждое ограничение может существенно сузить возможность создания предполагаемого решения. Следовательно, в процессе планирования необходимо тщательно изучить все ограничения (табл. 2.2).

Таблица 2.2. Возможные источники ограничений системы

Источник	Образцы вопросов
Экономический	Какие финансовые или бюджетные ограничения следует учесть? Существуют ли соображения, касающиеся себестоимости и ценообразования? Существуют ли вопросы лицензирования?
Политический	Существуют ли внешние или внутренние политические вопросы, влияющие на потенциальное решение? Существуют ли проблемы в отношениях между подразделениями?
Технический	Существуют ли ограничения в выборе технологий? Должны ли мы работать в рамках существующих платформ или технологий? Запрещено ли использование любых новых технологий? Должны ли мы использовать какие-либо закупаемые пакеты программного обеспечения?
Системный	Будет ли решение создаваться для существующих систем? Должны ли разработчики обеспечивать совместимость с существующими решениями? Какие операционные системы и среды должны поддерживаться?
Эксплуатационный	Существуют ли ограничения информационной среды или правовые ограничения? Юридические ограничения? Требования безопасности? Какими другими стандартами ограничены разработчики?
Графики и ресурсы	Ограничена ли команда программистов существующими ресурсами? Могут ли разработчики привлекать сотрудников со стороны? Можно ли увеличить штат? Временно?