

# Фазы жизненного цикла информационных систем

Управление жизненным циклом информационных систем

## 3.3. Проектирование



# Проектирование

- ▶ Техническое проектирование
- ▶ Рабочее проектирование и прототипирование

# Методы проектирования

Существует несколько основных методов проектирования ИС. Так, каноническое проектирование основано на пошаговом выполнении определенного порядка действий, при этом большинство процессов выполняются вручную или с минимальной автоматизацией. При этом проектирование зачастую осуществляется непосредственными исполнителями проекта.

Проектирование с использованием CASE-средств предполагает широкое применение программных инструментов, способных облегчить работу. Фокус внимания смещается на анализ и непосредственное проектирование, а инструменты позволяют автоматизировать создание документов, отчетности и генерацию кода.

Типовое проектирование применяется, когда речь идет об имеющихся типовых решениях, которые требуется адаптировать под потребности конкретной компании. В этом случае уникальный продукт не создается; напротив, происходит адаптация типового решения под реальные нужды бизнеса. Несмотря на кажущуюся простоту, типовое проектирование в сложных случаях немногим легче предыдущих методов.

В целом же проектирование ИТ-системы предполагает детальную подготовку к этапу настройки и развертывания модулей системы, для чего необходимо решить несколько задач:

- сформировать список модулей и функций системы, необходимых для поддержки определенных на этапе анализа автоматизируемых бизнес-процессов;
- сформировать список справочников систем (будущей и, если применимо, текущей) для дальнейшего переноса и обновления данных;
- определить примерный сценарий работы системы по категориям пользователей для формирования необходимого набора диалогов и процедур проектируемой системы (включая реакции на все возможные и даже очень маловероятные действия со стороны пользователя);
- определить элементы интерфейса пользователей (в случае гибкого или разрабатываемого «с нуля» решения) для достижения удобства работы с системой;

- сформировать список отчетов и панелей мониторинга (включая их формы и обязательные для реализации элементы); эти отчеты в дальнейшем будут использоваться как для учетных целей, так и для целей мониторинга системы ее администратором (в части формирования и сбора статистики по нагрузке на ее отдельные модули, свободным ресурсам, активности пользователей и т.п.);
- определить перечень настроек функциональных компонентов системы в соответствии с выделенными на предыдущем этапе требованиями;
- определить необходимость, возможности и пути интеграции с существующими и планируемыми к реализации системами на предприятии заказчика, чтобы своевременно предусмотреть технические средства для интеграции.

На этапе детального проектирования важно крайне четко определить все функциональные возможности системы и определить ее место в общей программной архитектуре предприятия.

### Пример

---

Если предполагается, что система взаимодействия с клиентами CRM будет получать данные из личных кабинетов клиентов на портале компании, необходимо предусмотреть подобную интеграцию. Если в компании уже внедрена и работает ERP-система, CRM должна получать и предоставлять ей данные.

---

С организационной точки зрения к этому моменту важна подготовка иерархической структуры работ, базового календарного плана и дорожной карты, которые позволили бы управлять проектом на протяжении всей его реализации.



## 3.3.1. Техническое проектирование

Цель данного этапа — подготовка к этапу настройки модулей системы в целом. Данный этап предусматривает разработку организационной и функциональной структур модуля системы, определение уровня автоматизации информационного процесса предприятия, определение структуры информационного и технического обеспечения и требований к их элементам, постановку и алгоритмизацию задач обработки данных, разработку системы ведения нормативно-справочной базы.

Результаты работ оформляются в виде технического проекта (проектного решения на автоматизацию модуля системы) и представляют собой задание на программирование. Техническое проектирование, создание, настройка, доработка и внедрение модуля системы осуществляются только на основании соответствующих утвержденных технических заданий (проектных решений).

**Технический проект** (проектное решение по автоматизации модуля) — документ, содержащий описание функциональных компонентов системы, необходимых для реализации бизнес-процессов, описанных на этапе подготовки проектного решения, а также необходимых доработок системы и (при необходимости) процедур интеграции с модулями (или внешними системами), внедренными ранее.

Перечень документов, создаваемых на стадии «Технический проект», определяется документом ГОСТ 34.201—89. Основной задачей этой стадии является разработка архитектуры системы и технических решений по ее реализации. Перечень документов, служащих базой технического проекта, определяется условиями договора и технического задания, но чаще всего в него включаются следующие документы.

1. Пояснительная записка:
  - общие положения, например, стадии и сроки, цели и задачи, используемые при разработке системы объекты ИС;
  - описание процесса деятельности;
  - основные технические решения, например, структура системы, основные функции, режимы функционирования, требования к персоналу;
  - мероприятия по подготовке объекта автоматизации к вводу системы в действие, например, подготовка информации к импорту, создание рабочих мест.

2. Входные и выходные данные системы, например, импортируемые и вводимые вручную пользователями данные (названия и источники документов), формируемые системой документы.

### 3. Схема функциональной структуры:

- описание функций подсистем, например, описание функций по отдельным модулям (например, модуль расчета заработной платы, модуль учета труда);
- информационные связи между элементами и с внешней средой.

### 4. Описание автоматизируемых функций:

- исходные данные;
- цели АС и автоматизируемые функции;
- характеристика функциональной структуры, например, список подсистем, описание процесса реализации функций, требования (к надежности, защите информации от несанкционированного доступа, сохранности информации);
- типовые решения.

### 5. Постановка задач и алгоритмы решения:

- характеристики комплекса задач;
- используемая информация;
- результаты решения, например, информация для выдачи выходных сообщений и данные для использования исключительно внутри системы;

## 6. Программное обеспечение:

- структура ПО;
- функции частей ПО;
- методы и средства разработки ПО, например, инструментальные средства проектирования модели предметной области, структур баз данных, разработки, применяемые методологии разработки, проектирования системы или интерфейсов;
  - операционная система, например, совместимые или предпочтительные ОС для клиентской части ПО и серверов (баз данных);
  - средства, расширяющие возможности ОС.

## 7. Информационное обеспечение:

- состав информационного обеспечения:

- **внутримашинная** информационная база — набор электронных таблиц и других объектов баз данных для обработки и хранения данных,

- **внемашинная** информационная база — информация, поступающая в бумажном виде (например, нормативно-справочная информация, годовые бюджеты);

- организация информационного обеспечения:

- принципы организации информационного обеспечения, например, БД должна представлять собой взаимосвязанные реляционные таблицы,

- обоснование выбора носителей данных, например, расчет числа запросов, которые необходимо обрабатывать, на основе статистики предыдущих периодов по числу пользователей, объему информации, частоте обращения к ней,

- принципы и методы контроля в маршрутах обработки данных, например, контроль форматов данных, контроль заполнения обязательных полей,

- описание решений, обеспечивающих информационную совместимость АС с другими системами;

- организация сбора и передачи информации:
  - основные источники информации, например, информация из внешних систем, бумажные носители информации, передаваемые по электронной почте файлы,
  - общие требования к организации сбора, передачи, контроля и корректировки информации;
- построение системы классификации и кодирования:
  - классификации объектов, принятые для применения в АС, например, Общероссийский классификатор валют (ОК 014–94), Коды представления названий стран (ИСО 3166–93), внутриорганизационный справочник сотрудников,
  - классификации объектов, принятые для применения в АС;
- организация внутримашинной информационной базы:
  - принципы построения,
  - структура, например, физическая структура БД системы (с полями, типами данных);
- организация внемашинной информационной базы (информации в бумажном виде от внешних организаций).

8. Комплекс технических средств системы:

- структура комплекса технических средств, например, принципы построения кластеров;
- вычислительный комплекс, например, серверы БД, приложений, веб-сервер, персональные компьютеры;
- абонентские пункты;
- аппаратура передачи данных.

9. Ведомость документов.



В результате подготовки технического проекта должен быть определен перечень компонентов модуля системы, необходимых доработок, процедур интеграции с существующей системой (при ее наличии и необходимости),

необходимый для реализации бизнес-модели предприятия To-Be. Так же однозначно утверждается перечень работ по настройке и доработке модуля системы, а также перечень и формы отчетов и первичных документов, получаемых из модуля системы. Помимо отчетов должны быть определены процедуры для справочников системы (при ее наличии), которые необходимо использовать для импорта, и тех справочников, для которых необходимо разработать механизмы двустороннего обновления информации.

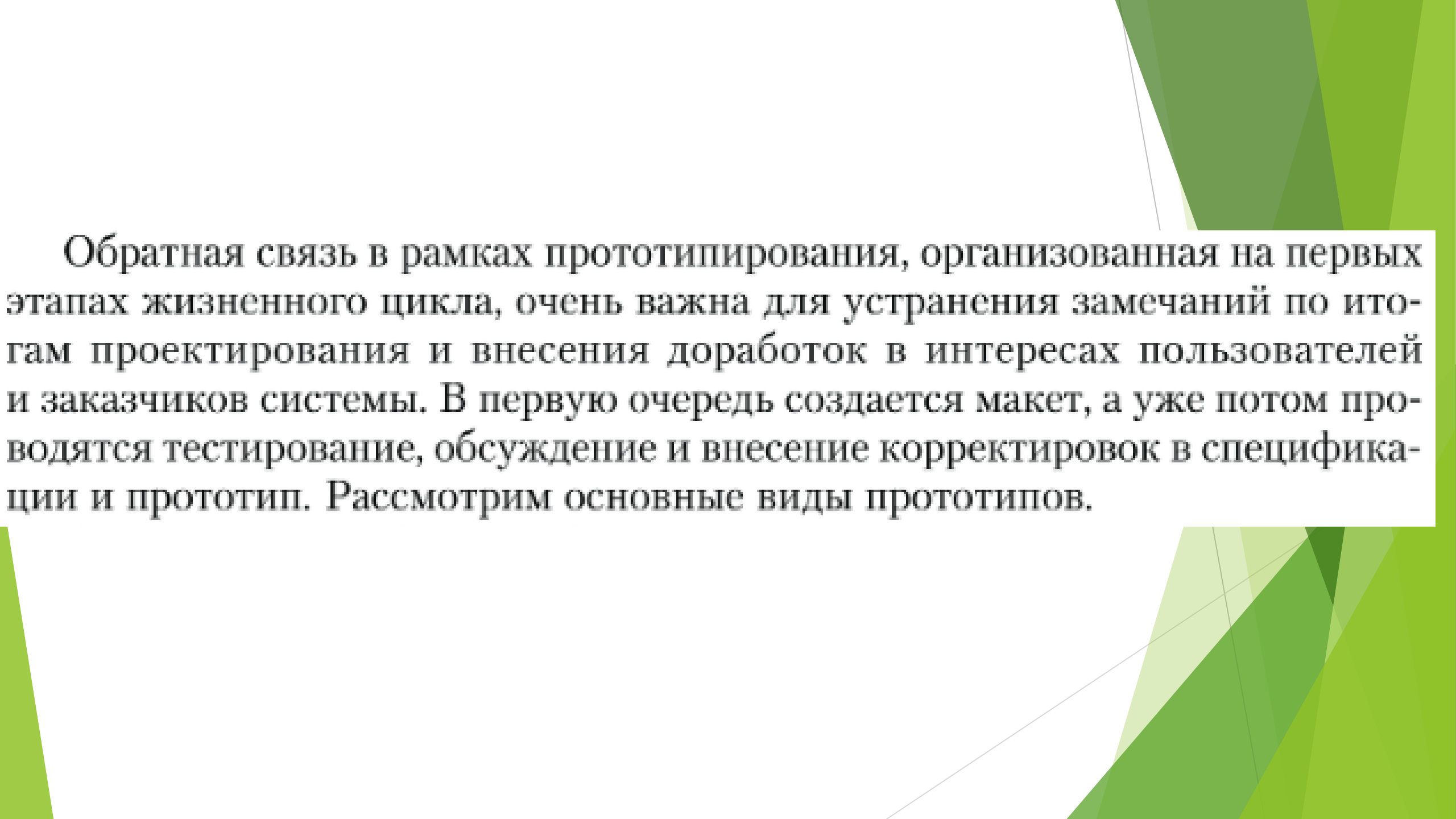
Важно отметить, что после детального определения и согласования всех вышеописанных параметров необходимо к моменту начала следующего этапа выбрать и подготовить также среду разработки и тестирования для проведения развертывания системы.

## 3.3.2. Рабочее проектирование и прототипирование

В целях снижения риска создания не соответствующей требованиям и техническому проекту системы перед разработкой возможно создать ее прототип (макет) для проверки предлагаемых архитектурных/функциональных/интерфейсных решений на практике с будущими пользователями.

---

**Прототип** — «черновая» реализация интерфейса и базовой функциональности системы для анализа принципов ее работы и тестирования совместно с будущими пользователями в целях дальнейшей доработки и совершенствования.

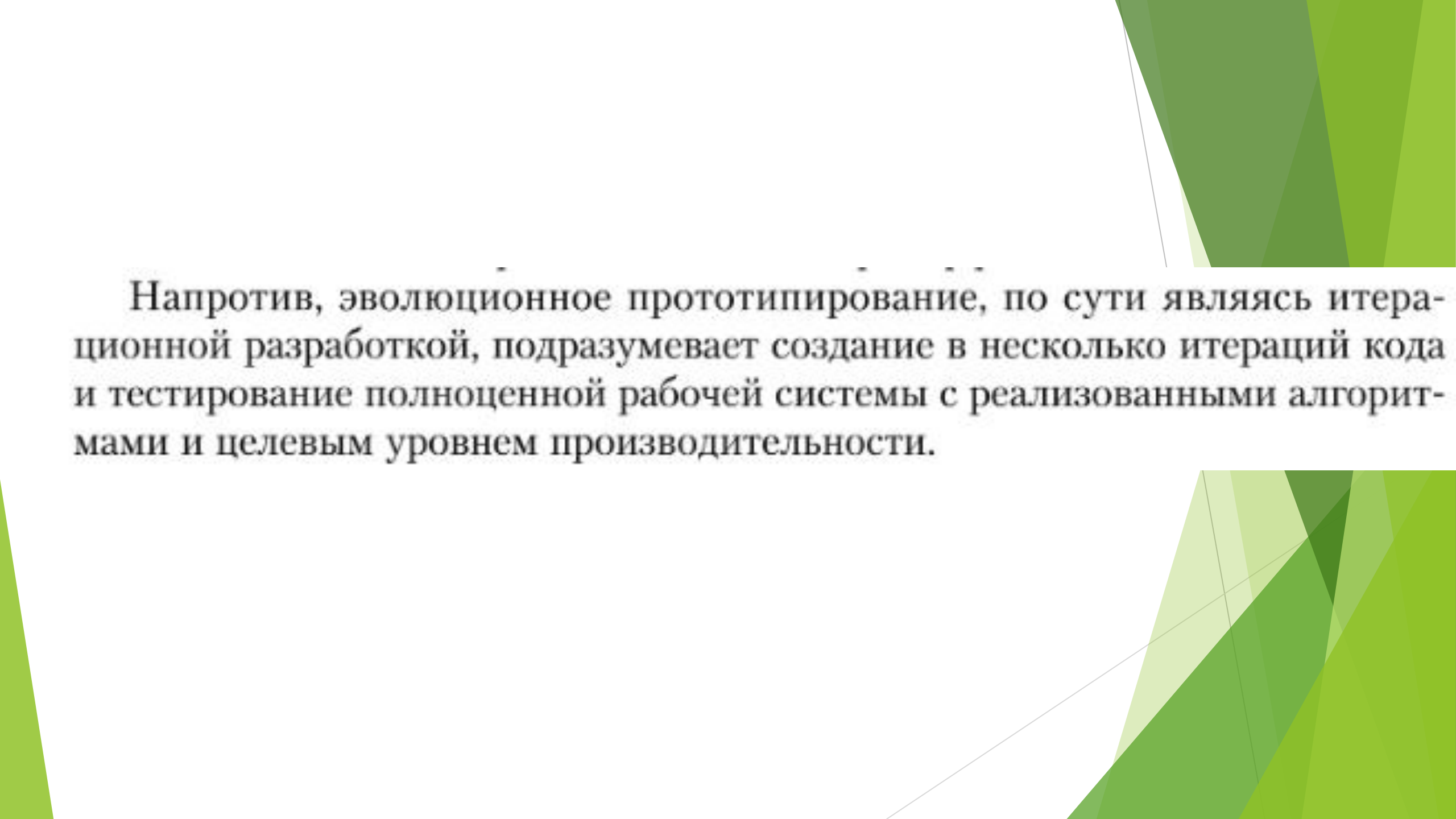


Обратная связь в рамках прототипирования, организованная на первых этапах жизненного цикла, очень важна для устранения замечаний по итогам проектирования и внесения доработок в интересах пользователей и заказчиков системы. В первую очередь создается макет, а уже потом проводятся тестирование, обсуждение и внесение корректировок в спецификации и прототип. Рассмотрим основные виды прототипов.

В качестве первой классификации можно выделить *«горизонтальное»* и *«вертикальное»* прототипирование. Первый случай (*«горизонтальное»* прототипирование) предполагает моделирование исключительно пользовательского интерфейса и форм без фокусирования на логике обработки информации. Имитируются результаты действий или запросов при нажатии на элементы управления и осуществляются переходы между формами системы для формирования представления о реакции системы на те или иные действия пользователя (и какие не предусмотренные в текущей реализации действия потенциальный пользователь может совершить). Это позволяет выявить и устранить противоречия между сформированными на этапе анализа требованиями и их реализацией в техническом проекте. А в рамках *«вертикального»* прототипа, в отличие от предыдущего варианта, фокус переходит на саму структуру системы для проверки корректности и работоспособности сформированного архитектурного решения.

Второй (и более известной) классификацией является «*быстрое*» и «*эволюционное*» прототипирование. В первом случае (как и следует

из названия) по технологии RAD, более подробно рассматриваемой в гл. 6, посвященной методологиям разработки, создается макет определенных компонентов системы для более предметного и эффективного диалога между разработчиками кода (интерфейсов) и пользователями. Важно, что подобный прототип не призван в дальнейшем дорабатываться и становиться частью системы, соответственно, достигается значительная экономия времени и ресурсов, так как нет необходимости фокусироваться на технических деталях быстрогодействия или энергоэффективности системы.



Напротив, эволюционное прототипирование, по сути являясь итерационной разработкой, подразумевает создание в несколько итераций кода и тестирование полноценной рабочей системы с реализованными алгоритмами и целевым уровнем производительности.